

COMMUNICATIONS

CACM.ACM.ORG

OF THE

ACM

10/2014 VOL.57 NO.10

Reading News with Maps by Exploiting Spatial Synonyms

Abstractions for
Software-Defined
Networks

Certificate
Transparency

Disrupting and
Transforming
the University

Gradual Evolution

Unconventional
Computing



CHI 2015

CROSSINGS

SEOUL • KOREA

CALL FOR SUBMISSIONS

DEADLINES

Paper & Notes

22 September 2014

Interactivity Demos, Case Studies, Workshops, Courses, Doctoral Consortium

6 October 2014

Other "late-breaking" content - see website for full details:

<http://chi2015.acm.org/authors/>

5 January 2015

Submit to CHI2015, ACM's premiere conference on human factors in computing systems! Join us for the first CHI conference in Asia in Seoul, Korea, a world-class center of emerging trends in culture, technology, and design.

The CHI2015 theme is Crossings: crossing boundaries, disciplines, and nations. We encourage submissions that reflect international perspectives on people and technology; scientists and practitioners; and academic and business interests. Showcase your latest research and design on the world's most innovative technologies in cross-disciplinary innovation and research in computer science, cognitive psychology, design, social science, human factors, AI, graphics, visualization, multimedia design and more.

See us in Seoul, Korea

18-23 April 2015



ACM Books



MORGAN & CLAYPOOL
PUBLISHERS

Publish your next book in the ACM Digital Library

ACM Books is a new series of advanced level books for the computer science community, published by ACM in collaboration with Morgan & Claypool Publishers.

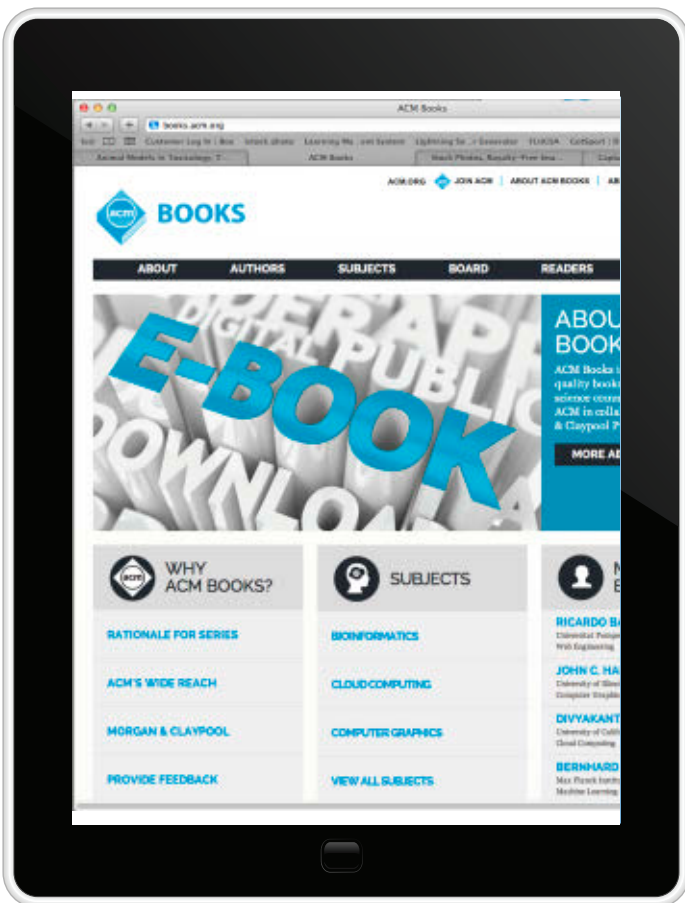
I'm pleased that ACM Books is directed by a volunteer organization headed by a dynamic, informed, energetic, visionary Editor-in-Chief (Tamer Özsu), working closely with a forward-looking publisher (Morgan and Claypool).

—Richard Snodgrass, University of Arizona

books.acm.org

ACM Books

- ◆ will include books from across the entire spectrum of computer science subject matter and will appeal to computing practitioners, researchers, educators, and students.
- ◆ will publish graduate level texts; research monographs/overviews of established and emerging fields; practitioner-level professional books; and books devoted to the history and social impact of computing.
- ◆ will be quickly and attractively published as ebooks and print volumes at affordable prices, and widely distributed in both print and digital formats through booksellers and to libraries and individual ACM members via the ACM Digital Library platform.
- ◆ is led by EIC M. Tamer Özsu, University of Waterloo, and a distinguished editorial board representing most areas of CS.



Proposals and inquiries welcome!

Contact: **M. Tamer Özsu**, Editor in Chief
booksubmissions@acm.org



Association for
Computing Machinery

Advancing Computing as a Science & Profession

Departments

5 **From ACM's Chief Executive Officer**
ACM's Challenges and Opportunities
By John White

7 **Cerf's Up**
Unconventional Computing
By Vinton G. Cerf

8 **Letters to the Editor**
Responsible Programming
Not a Technical Issue

10 **Blog@CACM**
Finding a Research Job, and
Teaching CS in High School
John Langford considers how to stand out when seeking a research position, while Mark Guzdial suggests what teachers need to know to teach computer science at the high school level.

35 **Calendar**

106 **Careers**

Last Byte

112 **Future Tense**
Garden of Life
When machines are in the natural world, what in the world is still unnatural?
By Daniel H. Wilson

News



13 **Still Seeking the Optical Transistor**
Optical information handling is a critical staple for communications and the Internet, but using light for computer-scale computation remains a distant dream.
By Don Monroe

16 **Gradual Evolution**
Dynamically typed languages adopt features of static typing to cope with growth.
By Neil Savage

19 **Museums Go High-Tech with Digital Forensics**
Scientists are using cutting-edge scanning and visualization techniques to wow visitors and find new stories in ancient artifacts.
By Nidhi Subbaraman

Viewpoints

22 **Technology Strategy and Management**
The Bitcoin Ecosystem
Speculating on how the Bitcoin economy might evolve.
By Michael A. Cusumano

25 **Inside Risks**
Risks and Myths of Cloud Computing and Cloud Storage
Considering existing and new types of risks inherent in cloud services.
By Peter G. Neumann

28 **Kode Vicious**
Outsourcing Responsibility
What do you do when your debugger fails you?
By George V. Neville-Neil

30 **The Business of Software**
Vendor: Vidi, Vici
Some hidden costs of outsourcing.
By Phillip G. Armour

32 **Viewpoint**
Disrupting and Transforming the University
Higher education institutions must modify their business models in response to technology-driven influences.
By Henry Lucas

36 **Viewpoint**
A Turing Tale
Assessing the accuracy of popular descriptions of Alan Turing's influences and legacy.
By Edgar G. Daylight

Practice



40

40 **Certificate Transparency**
Public, verifiable, append-only logs.
By Ben Laurie

47 **Security Collapse in the HTTPS Market**
Assessing legal and technical solutions to secure HTTPS.
By Axel Arnbak, Hadi Asghari, Michel van Eeten, and Nico van Eijk

56 **Why Is It Taking So Long to Secure Internet Routing?**
Routing security incidents can still slip past deployed security defenses.
By Sharon Goldberg

Contributed Articles



78

64 **Reading News with Maps by Exploiting Spatial Synonyms**
Use this map query interface to search the world, even when not sure what information you seek.
By Hanan Samet, Jagan Sankaranarayanan, Michael D. Lieberman, Marco D. Adelfio, Brendan C. Fruin, Jack M. Lotkowski, Daniele Panozzo, Jon Sperling, and Benjamin E. Teitler

78 **Wikidata: A Free Collaborative Knowledgebase**
This collaboratively edited knowledgebase provides a common source of data for Wikipedia, and everyone else.
By Denny Vrandečić and Markus Krötzsch

Review Articles



86

86 **Abstractions for Software-Defined Networks**
New abstractions are critical for achieving SDN goals.
By Martin Casado, Nate Foster, and Arjun Guha

Research Highlights

97 **Technical Perspective**
Attacking a Problem from the Middle
By Bart Preneel

98 **Dissection: A New Paradigm for Solving Bicomposite Search Problems**
By Itai Dinur, Orr Dunkelman, Nathan Keller, and Adi Shamir

IMAGES BY ALICIA KUBISTA/ANDREJ BORYS ASSOCIATES; ISAAC BASIRE (ILLUSTRATION); AAV PHOTO LAB



About the Cover:
This month's cover story (p. 64) introduces a system that uses a map query interface to present news found online. The system, NewsStand, demonstrates how extracting geographic details from news stories adds a new dimension of information for readers to better appreciate and understand the content. Cover illustration by Coherent Images.



ACM, the world's largest educational and scientific computing society, delivers resources that advance computing as a science and profession. ACM provides the computing field's premier Digital Library and serves its members and the computing profession with leading-edge publications, conferences, and career resources.

Executive Director and CEO

John White
Deputy Executive Director and COO
 Patricia Ryan
Director, Office of Information Systems
 Wayne Graves
Director, Office of Financial Services
 Darren Ramdin
Director, Office of SIG Services
 Donna Cappel
Director, Office of Publications
 Bernard Rous
Director, Office of Group Publishing
 Scott E. Delman

ACM COUNCIL

President
 Alexander L. Wolf
Vice-President
 Vicki L. Hanson
Secretary/Treasurer
 Erik Altman
Past President
 Vinton G. Cerf
Chair, SGB Board
 Patrick Manning
Co-Chairs, Publications Board
 Jack Davidson and Joseph Konstan
Members-at-Large
 Eric Allman; Ricardo Baeza-Yates;
 Cheri Pancake; Radia Perlman;
 Mary Lou Soffa; Eugene Spafford;
 Per Stenström
SGB Council Representatives
 Paul Beame; Barbara Boucher Owens;
 Andrew Sears

BOARD CHAIRS

Education Board
 Andrew McGettrick
Practitioners Board
 George Neville-Neil

REGIONAL COUNCIL CHAIRS

ACM Europe Council
 Fabrizio Gagliardi
ACM India Council
 Srinivas Padmanabhuni
ACM China Council
 Jiaguang Sun

PUBLICATIONS BOARD

Co-Chairs
 Jack Davidson; Joseph Konstan
Board Members
 Ronald F. Boisvert; Marie-Paule Cani;
 Nikil Dutt; Roch Guerrin; Carol Hutchins;
 Patrick Madden; Catherine McGeoch;
 M. Tamer Ozsu; Mary Lou Soffa

ACM U.S. Public Policy Office

Renee Dopplick, Acting Director
 1828 L Street, N.W., Suite 800
 Washington, DC 20036 USA
 T (202) 659-9711; F (202) 667-1066

Computer Science Teachers Association
 Lissa Clayborn, Acting Executive Director

COMMUNICATIONS OF THE ACM

Trusted insights for computing's leading professionals.

Communications of the ACM is the leading monthly print and online magazine for the computing and information technology fields. *Communications* is recognized as the most trusted and knowledgeable source of industry information for today's computing professional. *Communications* brings its readership in-depth coverage of emerging areas of computer science, new trends in information technology, and practical applications. Industry leaders use *Communications* as a platform to present and debate various technology implications, public policies, engineering challenges, and market trends. The prestige and unmatched reputation that *Communications of the ACM* enjoys today is built upon a 50-year commitment to high-quality editorial content and a steadfast dedication to advancing the arts, sciences, and applications of information technology.

STAFF

DIRECTOR OF GROUP PUBLISHING

Scott E. Delman
 publisher@cacm.acm.org

Executive Editor

Diane Crawford
Managing Editor
 Thomas E. Lambert

Senior Editor

Andrew Rosenbloom

Senior Editor/News

Larry Fisher

Web Editor

David Roman
Editorial Assistant
 Zarina Strakhan
Rights and Permissions
 Deborah Cotton

Art Director

Andrij Borys
Associate Art Director
 Margaret Gray

Assistant Art Director

Mia Angelica Balaquiot

Designer

Iwona Usakiewicz

Production Manager

Lynn D'Addesio

Director of Media Sales

Jennifer Ruzicka
Public Relations Coordinator
 Virginia Gold
Publications Assistant
 Emily Williams

Columnists

David Anderson; Phillip G. Armour;
 Michael Cusumano; Peter J. Denning;
 Mark Guzdial; Thomas Haigh;
 Leah Hoffmann; Mari Sako;
 Pamela Samuelson; Marshall Van Alstyne

CONTACT POINTS

Copyright permission
 permissions@cacm.acm.org

Calendar items
 calendar@cacm.acm.org

Change of address
 acmhelp@acm.org

Letters to the Editor
 letters@cacm.acm.org

WEBSITE

http://cacm.acm.org

AUTHOR GUIDELINES

http://cacm.acm.org/guidelines

ACM ADVERTISING DEPARTMENT

2 Penn Plaza, Suite 701, New York, NY
 10121-0701
 T (212) 626-0686
 F (212) 869-0481

Director of Media Sales

Jennifer Ruzicka
 jen.ruzicka@hq.acm.org

Media Kit acmm mediasales@acm.org

Association for Computing Machinery (ACM)

2 Penn Plaza, Suite 701
 New York, NY 10121-0701 USA
 T (212) 869-7440; F (212) 869-0481

EDITORIAL BOARD

EDITOR-IN-CHIEF

Moshe Y. Vardi
 eic@cacm.acm.org

NEWS

Co-Chairs

William Pulleyblank and Marc Snir

Board Members

Mei Kobayashi; Kurt Mehlforn;
 Michael Mitzenmacher; Rajeev Rastogi

VIEWPOINTS

Co-Chairs

Tim Finin; Susanne E. Hambrusch;
 John Leslie King

Board Members

William Aspray; Stefan Bechtold;
 Michael L. Best; Judith Bishop;
 Stuart I. Feldman; Peter Freeman;
 Mark Guzdial; Rachelle Hollander;
 Richard Ladner; Susan Landau;
 Carl Landwehr; Carlos Jose Pereira de Lucena;
 Beng Chin Ooi; Loren Terveen;
 Marshall Van Alstyne; Jeannette Wing

PRACTICE

Co-Chairs

Stephen Bourne

Board Members

Eric Allman; Charles Beeler; Bryan Cantrill;
 Terry Coatta; Stuart Feldman; Benjamin Fried;
 Pat Hanrahan; Tom Limoncelli;
 Kate Matsudaira; Marshall Kirk McKusick;
 Erik Meijer; George Neville-Neil;
 Theo Schlossnagle; Jim Waldo

The Practice section of the CACM

Editorial Board also serves as the Editorial Board of [queue](http://queue.acm.org).

CONTRIBUTED ARTICLES

Co-Chairs

Al Aho and Andrew Chien

Board Members

William Aiello; Robert Austin; Elisa Bertino;
 Gilles Brassard; Kim Bruce; Alan Bundy;
 Peter Buneman; Peter Druschel;
 Carlo Ghezzi; Carl Gutwin; Gal A. Kaminka;
 James Larus; Igor Markov; Gail C. Murphy;
 Shree Nayar; Bernhard Nebel;
 Lionel M. Ni; Kenton O'Hara;
 Sriram Rajamani; Marie-Christine Rousset;
 Avi Rubin; Krishan Sabnani;
 Ron Shamir; Yoav Shoham; Larry Snyder;
 Michael Vitale; Wolfgang Wahlster;
 Hannes Werthner; Reinhard Wilhelm

RESEARCH HIGHLIGHTS

Co-Chairs

Azer Bestavros and Gregory Morrisett

Board Members

Martin Abadi; Amr El Abbadi; Sanjeev Arora;
 Dan Boneh; Andrei Broder; Stuart K. Card;
 Jeff Chase; Jon Crowcroft; Matt Dwyer;
 Alon Halevy; Maurice Herlihy; Norm Jouppi;
 Andrew B. Kahng; Xavier Leroy; Kobbi Nissim;
 Mendel Rosenblum; David Salesin;
 Steve Seitz; Guy Steele, Jr.; David Wagner;
 Margaret H. Wright

WEB

Chair

James Landay

Board Members

Marti Hearst; Jason I. Hong;
 Jeff Johnson; Wendy E. MacKay

ACM Copyright Notice

Copyright © 2014 by Association for Computing Machinery, Inc. (ACM). Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and full citation on the first page. Copyright for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or fee. Request permission to publish from permissions@acm.org or fax (212) 869-0481.

For other copying of articles that carry a code at the bottom of the first or last page or screen display, copying is permitted provided that the per-copy fee indicated in the code is paid through the Copyright Clearance Center; www.copyright.com.

Subscriptions

An annual subscription cost is included in ACM member dues of \$99 (\$40 of which is allocated to a subscription to *Communications*); for students, cost is included in \$42 dues (\$20 of which is allocated to a *Communications* subscription). A nonmember annual subscription is \$100.

ACM Media Advertising Policy

Communications of the ACM and other ACM Media publications accept advertising in both print and electronic formats. All advertising in ACM Media publications is at the discretion of ACM and is intended to provide financial support for the various activities and services for ACM members. Current Advertising Rates can be found by visiting <http://www.acm-media.org> or by contacting ACM Media Sales at (212) 626-0686.

Single Copies

Single copies of *Communications of the ACM* are available for purchase. Please contact acmhelp@acm.org.

COMMUNICATIONS OF THE ACM

(ISSN 0001-0782) is published monthly by ACM Media, 2 Penn Plaza, Suite 701, New York, NY 10121-0701. Periodicals postage paid at New York, NY 10001, and other mailing offices.

POSTMASTER

Please send address changes to *Communications of the ACM*
 2 Penn Plaza, Suite 701
 New York, NY 10121-0701 USA

Printed in the U.S.A.



Association for Computing Machinery



ACM's Challenges and Opportunities

ACM held a strategic planning retreat last November. The motivation for the retreat was the realization that while ACM is a large, financially sound, and increasingly relevant

educational and scientific computing society, the ecosystem that supports scholarly and professional societies is undergoing rapid change, and this change is creating significant challenges.

The retreat involved an in-depth look at membership, publications, and conferences/SIGs. Over two days, ideas regarding future ACM directions emerged within each of these areas, ideas emerged that crosscut multiple areas, and ideas emerged at varying levels of detail.

Membership. What surfaced in the membership discussion as a primary new direction to consider is understanding and focusing on ACM's reach and doing more with the large segment of the computing community "touched" by ACM in one form or another. Back-of-the-envelope estimates suggest the size of this community is on the order of 3.4 million individuals ... individuals who use ACM and/or contribute to ACM, but are not members. The community includes: SIG-only members, individual subscribers to publications, admins/librarians, institutional users of ACM's Digital Library, article purchasers, Web account holders, Listserv subscribers, webinar registrants, conference attendees, chapter-only members, website visitors, authors, conference committees, reviewers and PC members, and social media followers.

This broad ACM community has a varying level of engagement with ACM. One portion is primarily users of ACM content (for example, website visitors, webinar registrants, social media followers), while others actually contribute to ACM's mission (for example, authors, reviewers, PC members, conference committees). There was a desire to: 1) understand and capture/record who these individuals are; 2) recognize them more formally; and 3) give

something back—particularly to those who contribute to ACM's mission.

Publications. Over the two days of the retreat, we reviewed multiple dimensions of the ACM Publications program. In the discussion, there was general agreement that for the foreseeable future it was in ACM's best interests to see Publications generate a surplus that enables a wide range of ACM activities. That said, there was significant discussion of the challenges facing the Publications business.

The bulk of these challenges stem from the open access (OA) movement and its potential impact on subscription-based publishing. To date, ACM has moved to be more in line with OA while continuing to grow DL revenue. While we agreed ACM can expect substantial DL revenue in the short term, the rate of DL revenue growth will likely decline, and the long-term revenue picture for the DL is unclear.

Given these points and the related retreat discussion, there are short-term and long-term issues to explore and address. In the short run, we need to improve the current publications operations, move further with OA publishing within ACM, and critically review the current portfolio of publications.

Regarding the longer run, our discussion centered on issues surrounding the future of scientific publishing in general. The issues at the core of this discussion were: 1) journal vs. conference publishing; 2) a vision of what published content will look like in the future; and 3) a future vision for the DL in terms of features, functions, and business models.

Conferences and Technical Communities (SIGs). From the beginning, ACM SIGs have had significant technical and financial autonomy within ACM. With this autonomy has come

significant responsibility for fostering strong communities. Over the past 50 years, this model has proved incredibly successful. SIGs dominate the technical landscape of ACM.


In moving a relatively healthy SIG structure forward, the retreat identified a handful of issues to explore and address. First and foremost was to consider further repositioning conferences within the publishing culture of computing research. There is tension with the community's success in establishing conferences as a unique and major publishing venue for computing. Multiple issues need further in-depth consideration: the proliferation of conferences and the trend toward a publishing culture of incremental results; the demise of workshops as a venue for informal presentation and discussion of research; the role of journal vs. conference publishing. The second area of discussion was focused on fostering conferences outside the SIG structure.

In addition to the topics noted here, there was discussion regarding three major cross-cutting issues:

- ▶ **Community:** build and support a sense of "community" in everything we do,

- ▶ **Quality:** ensure a high level of quality and relevance across everything we do, and

- ▶ **Practitioners:** explicitly consider reaching and serving practitioners in everything we do.

A lot came out of the ACM retreat. There are now committees and task forces established to address the main outcomes. The goal of the new administration is to see these committees complete their work and, as a result, see a significant change in how ACM addresses membership, publications, conferences, and technical communities while maintaining quality, building a broad sense of community, and serving practitioners. 

John White (white@hq.acm.org) is the CEO of ACM.

© 2014 ACM 0001-0782/14/10 \$15.00

**Previous
A.M. Turing Award
Recipients**

1966 A.J. Perlis
1967 Maurice Wilkes
1968 R.W. Hamming
1969 Marvin Minsky
1970 J.H. Wilkinson
1971 John McCarthy
1972 E.W. Dijkstra
1973 Charles Bachman
1974 Donald Knuth
1975 Allen Newell
1975 Herbert Simon
1976 Michael Rabin
1976 Dana Scott
1977 John Backus
1978 Robert Floyd
1979 Kenneth Iverson
1980 C.A.R Hoare
1981 Edgar Codd
1982 Stephen Cook
1983 Ken Thompson
1983 Dennis Ritchie
1984 Niklaus Wirth
1985 Richard Karp
1986 John Hopcroft
1986 Robert Tarjan
1987 John Cocke
1988 Ivan Sutherland
1989 William Kahan
1990 Fernando Corbató
1991 Robin Milner
1992 Butler Lampson
1993 Juris Hartmanis
1993 Richard Stearns
1994 Edward Feigenbaum
1994 Raj Reddy
1995 Manuel Blum
1996 Amir Pnueli
1997 Douglas Engelbart
1998 James Gray
1999 Frederick Brooks
2000 Andrew Yao
2001 Ole-Johan Dahl
2001 Kristen Nygaard
2002 Leonard Adleman
2002 Ronald Rivest
2002 Adi Shamir
2003 Alan Kay
2004 Vinton Cerf
2004 Robert Kahn
2005 Peter Naur
2006 Frances E. Allen
2007 Edmund Clarke
2007 E. Allen Emerson
2007 Joseph Sifakis
2008 Barbara Liskov
2009 Charles P. Thacker
2010 Leslie G. Valiant
2011 Judea Pearl
2012 Shafi Goldwasser
2012 Silvio Micali
2013 Leslie Lamport

ACM A.M. TURING AWARD NOMINATIONS SOLICITED

Nominations are invited for the 2014 ACM A.M. Turing Award. This, ACM's oldest and most prestigious award, is presented for contributions of a technical nature to the computing community. Although the long-term influences of the nominee's work are taken into consideration, there should be a particular outstanding and trendsetting technical achievement that constitutes the principal claim to the award. The recipient presents an address at an ACM event that will be published in an ACM journal.

Nominations should include:

- 1) A curriculum vitae, listing publications, patents, honors, and other awards.
- 2) A letter from the principal nominator, which describes the work of the nominee, and draws particular attention to the contribution which is seen as meriting the award.
- 3) Supporting letters from at least three endorsers. The letters should not all be from colleagues or co-workers who are closely associated with the nominee, and preferably should come from individuals at more than one organization. Successful Turing Award nominations should include substantive letters of support from prominent individuals broadly representative of the candidate's field or related field impacted by the candidate's contribution.

**For additional information on ACM's award program
please visit: www.acm.org/awards/**

**Additional information on the past recipients
of the A.M. Turing Award is available on:
<http://amturing.acm.org/byyear.cfm>.**

**Nominations should be sent electronically
by November 30, 2014 to:
[Barbara Liskov, MIT CSAIL c/o mcguinness@acm.org](mailto:Barbara.Liskov@MIT.CSAIL)**



Association for
Computing Machinery



Vinton G. Cerf

DOI:10.1145/2666093

Unconventional Computing

The August 2014 issue of *IEEE Spectrum* had two articles of interest related to computing: “Silicon’s Second Act” and “Spin Memory Shows Its Might.”

On top of that, in the last couple of years, IBM has demonstrated two remarkable achievements: The Watson Artificial Intelligence system and the August 8, 2014 cover story of *Science* entitled “Brain Inspired Chip.” The TrueNorth chipset and the programming language it uses have demonstrated remarkable power efficiency compared to more conventional processing elements.

What all of these topics have in common for me is the prospect of increasingly unconventional computing methods that may naturally force us to rethink how we analyze problems for purposes of getting computers to solve them for us. I consider this to be a refreshing development, challenging the academic, research, and practitioner communities to abandon or adapt past practices and to consider new ones that can take advantage of new technologies and techniques.

It has always been my experience that half the battle in problem solving is to express the problem in such a way the solution may suggest itself. In mathematics, it is often the case that a change of variables can dramatically restructure the way in which the problem or formula is presented; leading one to find related problems whose solutions may be more readily applied. Changing from Cartesian to Polar coordinates often dramatically simplifies its expression. For example, a Cartesian equation for a circle centered at (0,0) is $X^2 + Y^2 = Z^2$ but the polar version is simply $r(\varphi) = a$ for some value of a .


It may prove to be the case that the computational methods for solving problems with quantum computers, neural chips, and Watson-like systems will admit very different strategies and tactics than those applied in more conventional architectures. The use of graphics processing units (GPUs) to solve problems, rather than generating textured triangles at high speed, has already forced programmers to think differently about the way in which they express and compute their results. The parallelism of the GPUs and their ability to process many small “programs” at once has made them attractive for evolutionary or genetic programming, for example.

One question is: Where will these new technologies take us? We have had experiences in the past with unusual designs. The Connection Machine designed by Danny Hillis was one of the first really large-scale computing machines (65K one-bit processors) hyperconnected together. LISP was one of the programming languages used for the Connection Machines along with URDU, among others. This brings to mind the earlier LISP machines made by Symbolics and LISP Machines, Inc., among others. The rapid advance in speed of more conventional processors largely overtook the advantage of special purpose, potentially language-oriented computers. This was particularly evident with the rise of the so-called RISC (Reduced Instruction Set Computing) machines developed by John Hennessy (the MIPS system) and David Patterson

(Berkeley RISC and Sun Microsystems SPARC), among many others.

David E. Shaw, at Columbia University, pioneered one of the explorations into a series of designs of a single instruction stream, multiple data stream (SIMD) supercomputer he called Non-Von (for “non-Von-Neumann”). Using single-bit arithmetic logic units, this design has some relative similarity to the Connection Machine although their interconnection designs were quite different. It has not escaped my attention that David Shaw is now the chief scientist of D.E. Shaw Research and is focused on computational biochemistry and bioinformatics. This topic also occupies his time at Columbia University, where he holds a senior research fellowship and adjunct professorship.

Returning to new computing and memory technologies, one has the impression the limitations of conventional use of silicon technology may be overcome with new materials and with new architectural designs as is beginning to be apparent with the new IBM Neural chip.

I have only taken time to offer a very incomplete and sketchy set of observations about unconventional computing in this column, but I think it is arguable that in this second decade of the 21st century, we are starting to see serious opportunities for rethinking how we may compute. 

Vinton G. Cerf is vice president and Chief Internet Evangelist at Google. He served as ACM president from 2012–2014.

Copyright held by author.

Responsible Programming Not a Technical Issue

VINTON G. CERF'S Cerf's Up column "Responsible Programming" (July 2014) raised the interesting question of whether it is responsible to produce software without using state-of-the-art defect-detection-and-validation tools. Research on such tools is ongoing for decades, and despite progress made possible through improved SAT solvers and theorem provers, software-defect tools are known primarily within their research community and rarely used in development projects, open or closed source. Perhaps a more important question involves how computer science can accelerate development of effective tools and motivate their widespread adoption.

The answer is legal, not technical. The standard software license, disclaiming all liability and suitability for any purpose, should be prohibited, with software subject to the same liability and suitability terms as any other manufactured product. If software is insecure or simply does not work, then the manufacturer must show it was constructed with sufficient care to avoid liability. This financial and legal pressure would eventually end egregious practices, including failure to use the best available tools and practices, shipping bug-ridden code, and using customers as beta testers.

The transition from the current state of software to where it should be will take time, financial investment, new ideas, and great effort. It cannot happen overnight but might never happen if software producers avoid assuming responsibility for the correct, secure operation of their products.

James Larus, Lausanne, Switzerland

Make Security the Predominant Architecture

Just before reading Seda Gurses's Viewpoint "Can You Engineer Privacy?" (Aug. 2014), I had been reading

the latest on hacking car control units by manipulating the software controlling the car, especially the engine, the steering wheel, and other car components,¹ pondering the need for a new approach to security and privacy.

Why are intruders so successful? For one thing, computer science and engineering often simplifies attacks, with appliances and application systems using standardized and generalized algorithms, protocols, and component systems. These concepts are also the basis of the software industry's ability to quickly develop new systems that are open for further development. Intruders are likewise able to create tools for unwelcome manipulation.

What new paradigm of computer science would allow software developers to improve system security and personal privacy? How about one that is application-specific, employs nonstandard protocols and address

The standard software license, disclaiming all liability and suitability for any purpose, should be prohibited, with software subject to the same liability and suitability terms as any other manufactured product.

schemes (such as on LANs in cars), and eliminates concepts like algorithms and data structures "reserved for future use," or more general algorithms in applications than are needed ("upward compatibility"), as in a companywide hardware and software platform in car computers? (This is not to say I advocate the idea of handcrafting all future secure systems.)

Rather than make it easy for would-be intruders to develop generalized tools, application engineers should look to develop standardization variation generators, or SVPs, to create strategic complexity specific to families of applications or even to individual appliances. In the case of cars, SVPs must be able to generate a specific protocol for communication between sensors, steering activators, and control processors, even though they are derived from a general class of protocols. Dynamic solutions like protocol variations that depend on car-key identification are especially promising, not by substituting encryption and information hiding but by providing another self-contained obstacle to foil intruders.

Privacy and security can be engineered, even in highly sensitive systems, but such engineering works only if application system architects and software developers view computer security as the predominant architecture, not just as added functionality, on which to develop applications.

Georg E. Schaefer, Ulm, Germany

Reference

1. Pauli, D. Students hack Tesla Model S, make all its doors pop open in motion. *The Register* (July 21, 2014).

Hold the Politics

For Moshe Y. Vardi to state as fact, as he did in his Editor's Letter "Openism, IPism, Fundamentalism, and Pragmatism" (Aug. 2014), "Only the drastic measures taken by the U.S. government..." averted catastrophe as a result of the Lehman Brothers bankruptcy and "...this event shredded the dogma

of capitalism..." is counter to the view of many, including me, that the Lehman collapse is an offshoot of national, state, county, and municipal government structure and policy that sucks up 45% of the economy in the form of taxes. Vardi characterizing it as "fundamentalism" then claiming "...history shows that fundamentalist ideas rarely work..." misdirects the argument by attempting to associate criticism of the current economic mess with religious fundamentalism, William Jennings Bryan, and the Scopes trial, rather than the actual substance of the view.

The core of the column seemed to be whether the "reader-pays" or "author-pays" publication model is "more sustainable" for the ACM Digital Library. Fine. That can be done without an anti-capitalist diatribe. It should be remembered that technology is the core interest of the membership, including both liberals and conservatives, and all have access to political discussion elsewhere.

Bryan Batten, Carlsbad, CA

Math and the Computing Paradigm

Regarding Peter J. Denning's and Peter A. Freeman's Viewpoint "Computing's Paradigm" (Dec. 2009), Simone Santini's letter to the editor "Computing Paradigm Not a Branch of Science" (Apr. 2010) said computing can be categorized as both a branch of science and a branch of mathematics, claiming, "The abstract problem of symbol manipulation is mathematical..." and "The instantiation of the symbol-manipulation model in useful systems is a problem for the engineering of computing." In response, Denning and Freeman said, "Computing does not separate neatly into math and engineering, as Santini claims." But what is indeed wrong with Santini's distinction, which has been endorsed by many others over the years? Denning and Freeman even predicted, "Santini's desire to parse computing into separate elements will fail, just as all such previous attempts have failed."

Virtually all software development is done through trial and error, (unit) testing, and never-ending patching following delivery, with stupendous (productivity) costs; recall the classic

Microsoft software alert: "You may have to reboot your system." There are good reasons for this practice. One is there are no tools (or supporting theory) for systematic top-down iterative formal development, from requirements to running system. Most software products do not need meaning-preserving transformations or formal verifications.

This state of the art does not mean we can dismiss a math approach to development, validation, and annotations of library elements for machine-assisted reuse. It is actually a failure of computer science, better called "informatics," to have not developed a math approach to the software development life cycle. Consider recent unwelcome consequences of the lack of formal verification techniques: the Heartbleed flaw in OpenSSL, Goto fail in Apple OS, and the CVE-2014-1776 patch for Internet Explorer.

Though sorting belongs to one of the oldest algorithms in computer science, the Cygwin library (part of a Unix-like command-line interface for Microsoft Windows) had (still has?) an "improved" defective version of qsort with guaranteed quadratic behavior on certain inputs. In any case, I have never encountered even an informal proof that the output of a sorting algorithm is a permutation of the input.

This is not to say I think computer science should be viewed as a branch of mathematics but rather as a way to urge more research in formal techniques, hopefully yielding tools for all phases of the development life cycle. Redeveloping the Linux operating system this way would be a genuine advance, making it possible to maintain it at a high level instead of exclusively tinkering with its code.

Denning's and Freeman's response should not have demeaned Santini's distinction, endorsing again and again the pathological optimism approach (such as Scrum and Agile) to software development. In the meantime, see my Technical Opinion "Software Engineering Considered Harmful" (Nov. 2002).

Dennis de Champeaux, San Jose, CA

Communications welcomes your opinion. To submit a Letter to the Editor, please limit yourself to 500 words or less, and send to letters@cacm.acm.org.

© 2014 ACM 0001-0782/14/10 \$15.00

Coming Next Month in COMMUNICATIONS

Scene Understanding by Labeling Pixels

The Data on Diversity

An Embarrassment of Riches: A Critical Review of Open Innovation Systems

Designing User Incentives for Cybersecurity

On Facebook, Most Ties Are Weak

Building Volumetric Appearance Models of Fabric Using Micro CT Imaging

Plus the latest news about parallel programming, keeping online reviews honest, and computing what fits.

The *Communications* Web site, <http://cacm.acm.org>, features more than a dozen bloggers in the BLOG@CACM community. In each issue of *Communications*, we'll publish selected posts or excerpts.

twitter

Follow us on Twitter at <http://twitter.com/blogCACM>

DOI:10.1145/2659758

<http://cacm.acm.org/blogs/blog-cacm>

Finding a Research Job, and Teaching CS in High School

John Langford considers how to stand out when seeking a research position, while Mark Guzdial suggests what teachers need to know to teach computer science at the high school level.



John Langford
The Perfect Candidate

<http://bit.ly/1p2ep9F>
July 15, 2014

The last several years have seen phenomenal growth in machine learning, such that this earlier post from 2007 (<http://bit.ly/1o7LY1f>) is understated. Machine learning jobs are growing everywhere. The core dynamic is a digitizing world, which makes people who know how to use data effectively a very hot commodity. Anyone reasonably familiar with machine learning tools and a master's level of education can get a good job at many companies, while Ph.D. students coming out sometimes have bidding wars and many professors have created startups.

Despite this, hiring in good research positions can be challenging. A good research position is one where you can:

- ▶ Spend the majority of your time working on research questions that interest you.
- ▶ Work with other like-minded people.
- ▶ For several years.

I see these as critical. Research is hard enough that you cannot expect to succeed without devoting the majority of your time. You cannot hope to succeed without personal interest. Other like-minded people are typically necessary in finding the solutions of the hardest problems. Typically you must work for several years before seeing significant success. There are exceptions to everything, but these criteria are the working norm of successful research I see.

The set of good research positions is expanding, but at a much slower pace than applied scientist types of positions. This makes sense, as the pool of people able to do interesting research grows slowly, and anyone funding this should think hard before making the necessary expensive commitment for success.

What makes a good candidate for a research position? People have many diverse preferences, so I can only speak for myself with any authority. There are several things I do and do not look for:

1. **Something new.** Any good candidate should have something worth teach-

ing. For a Ph.D. candidate, the subject of your research is deeply dependent on your advisor. It is not necessary that you do something different from your advisor's research direction, but it is necessary that you own (and can speak authoritatively about) a significant advance.

2. **Something other than papers.** It is quite possible to persist indefinitely in academia while only writing papers, but it does not show a real interest in what you are doing beyond survival. Why are you doing it? What is the purpose? Some people code. Some people solve particular applications. There are other things as well, but these make the difference.

3. **A difficult long-term goal.** A goal suggests interest, but more importantly it makes research accumulate. Some people do research without a goal, solving whatever problems happen to pass by that they can solve. Very smart people can do well in research careers with a random walk amongst research problems, but people with a goal can have their research accumulate in a much stronger fashion. I am not an extremist here—solving off-goal problems is fine and desirable, but having a long-term goal makes a long-term difference.

4. **A portfolio of coauthors.** This shows you are able to and interested in working with other people, as is often necessary for success. This can be particularly difficult for Ph.D. candidates whose advisors expect them to work exclusively with (or for) them. Summer internships are both a strong tradition and a great opportunity here.

5. **I rarely trust recommendations because I find them difficult to interpret.**

When the candidate selects the writers, the most interesting bit is who the writers are. Letters default positive, but the degree of default varies from writer to writer. Occasionally, a recommendation says something surprising, but do you trust the recommender's judgment? In some cases yes, but in many cases you do not know the writer.

Meeting the above criteria within the context of a Ph.D. is extraordinarily difficult. The good news is that you can "fail" with a job that is better in just about every way.

Any time criteria are discussed, it is worth asking: should you optimize for them? In another context, lines of code (<http://bit.ly/1sny5sc>) is a terrible metric to optimize when judging programmer productivity. Here, I believe optimizing for (1), (2), (3), and (4) are all beneficial and worthwhile for Ph.D. students.



Mark Guzdial
What It Takes
to be a Successful
High School Computer
Science Teacher

<http://bit.ly/UNV9Dx>
 May 14, 2014

Around the world, education systems are moving computing into primary and secondary schools. Whole nations (such as England and Denmark) and individual U.S. states are trying to figure out how to teach computing to students in K-12 (<http://bit.ly/1pU93wh>). A recent *Economist* editorial (<http://econ.st/1snyye4>) highlights the biggest challenge to getting computing into schools: How do we get enough well-prepared computer science teachers?

The first part of answering that question is: "Who are we starting with?" In Israel, the Technion is preparing high school computer science teachers from graduates with degrees in STEM (<http://bit.ly/1qNf67z>). The Computing at School effort in England is working to prepare existing Information and Communications Technologies (ICT) teachers to teach computer science (<http://bit.ly/1o7quwM>). Here in the U.S., we are mostly preparing business teachers to teach computer science (<http://bit.ly/1oC5puh>), because computer science is classified as Career and Technical Education (CATE) in most states (<http://bit.ly/1pznE2F>), and CATE teachers have business teaching certifications.

A related question is: "What do we need to prepare the teachers to do?" Sure, teachers who have an undergraduate degree have the content knowledge. ICT teachers may know *something* about computing, but may not know CS. Existing teachers know a lot about running classes, but may not have the CS background. The best possible world is to have teachers who know CS content, teaching practices, and how to teach CS, but the reality is that we will have to prioritize. What does it take to be a successful high school computer science teacher?

When we were working on teacher professional development in the early days of our "Georgia Computes!" project (<http://bit.ly/1mg860a>), our external evaluator interviewed high school teachers of Advanced Placement Computer Science classes. He grouped the teachers in terms of more-successful teachers (able to recruit more students into the AP CS course, had a high pass rate of students on the AP CS exam, were confident and satisfied with being an AP CS teacher) and less-successful teachers. The interviews give us some insight into what we want high school CS teachers to be able to do.

The quotes below were in response to a question about how the teacher prepared students for the AP CS exam.

Everything in that class is more or less an assessment. They are supposed to read certain sections in the book, and then they have quizzes over the reading. After they do the reading assignments, they have Gridworld case study quizzes and also Gridworld case study segments of code that they will go in and manipulate to change to get the things in the Gridworld case study to react different ways. Those are pretty much graded as labs or programs or quizzes.

The teacher in the above quote is describing is what you would expect any high school teacher to do when teaching pretty much anything. There is a heavy emphasis on assessment and reading. This is one of our less-successful teachers.

If I read these [student quizzes], I can see any misconceptions or gaps in what I have done. I get a picture in my mind of where the current class is. Making them do the explaining is new this year. I am seeing them do a lot better there. I will do little code (assignments) that they will write once a week. They have to write it by hand away from the computer, and I will read that and write

comments on what they are doing and help them grade it with a rubric, and also pass them back after I have read them for them to grade, too, and have them look at what was catching it or where it did not quite get to it.

This is a response from a more-successful teacher. We see a lot of CS-specific teaching techniques: students explaining their code, writing code by hand away from the computer (as well as at the computer), and self-grading of code by rubric.

Particularly interesting to me is what the teacher is *doing* in each of these quotes. The first teacher makes assignments. The second teacher talks about creating assignments and rubrics, but she also reads student quizzes, and reads and comments on student code. *None of our successful teachers ever talked about writing programs.*

That is a big difference from what we teach CS majors. The latest ACM/IEEE computer science curriculum standards (<http://bit.ly/1omiR0K>) highlight that we expect graduates to be professional software developers who use good engineering practices. A teacher with software development knowledge and skills certainly would know how to read and comment on student code, but that is a small part of what we teach people to do as software developers.

Studies like these give me hope we can provide professional learning opportunities to existing teachers *without* trying to turn them into software developers. We mostly need to teach CS teachers to read code. Raymond Lister (<http://bit.ly/1pznSa4>) has done a lot of research exploring the developmental path from being able to read and trace code into being able to write code, and his results suggest reading and tracing *precedes* code-writing skills. It should be easier to teach reading than to teach software development.

Can we teach teachers to read and comment on code without teaching them to be software developers? Can we teach reading code more efficiently to a broad range of teachers than we could teach software development skills to those same teachers? These are important questions to answer to be able to prepare enough high school computer science teachers worldwide. □

John Langford is a senior researcher at Microsoft Research. Mark Guzdial is a professor at the Georgia Institute of Technology.

© 2014 ACM 0001-0782/14/10 \$15.00

SHAPE THE FUTURE OF COMPUTING. JOIN ACM TODAY.

ACM is the world's largest computing society, offering benefits that can advance your career and enrich your knowledge with life-long learning resources. We dare to be the best we can be, believing what we do is a force for good, and in joining together to shape the future of computing.

SELECT ONE MEMBERSHIP OPTION

ACM PROFESSIONAL MEMBERSHIP:

- Professional Membership: \$99 USD
- Professional Membership plus
ACM Digital Library: \$198 USD (\$99 dues + \$99 DL)
- ACM Digital Library: \$99 USD
(must be an ACM member)

ACM STUDENT MEMBERSHIP:

- Student Membership: \$19 USD
- Student Membership plus ACM Digital Library: \$42 USD
- Student Membership PLUS Print *CACM* Magazine: \$42 USD
- ACM Student Membership w/Digital Library
PLUS Print *CACM* Magazine: \$62 USD

- Join ACM-W:** ACM-W supports, celebrates, and advocates internationally for the full engagement of women in all aspects of the computing field. Available at no additional cost.

Priority Code: CAPP

Payment Information

Name

ACM Member #

Mailing Address

City/State/Province

ZIP/Postal Code/Country

Email

Payment must accompany application. If paying by check or money order, make payable to ACM, Inc, in U.S. dollars or equivalent in foreign currency.

- AMEX
- VISA/MasterCard
- Check/money order

Total Amount Due

Credit Card #

Exp. Date

Signature

Return completed application to:
ACM General Post Office
P.O. Box 30777
New York, NY 10087-0777

Prices include surface delivery charge. Expedited Air Service, which is a partial air freight delivery service, is available outside North America. Contact ACM for more information.

Satisfaction Guaranteed!

BE CREATIVE. STAY CONNECTED. KEEP INVENTING.



Association for
Computing Machinery

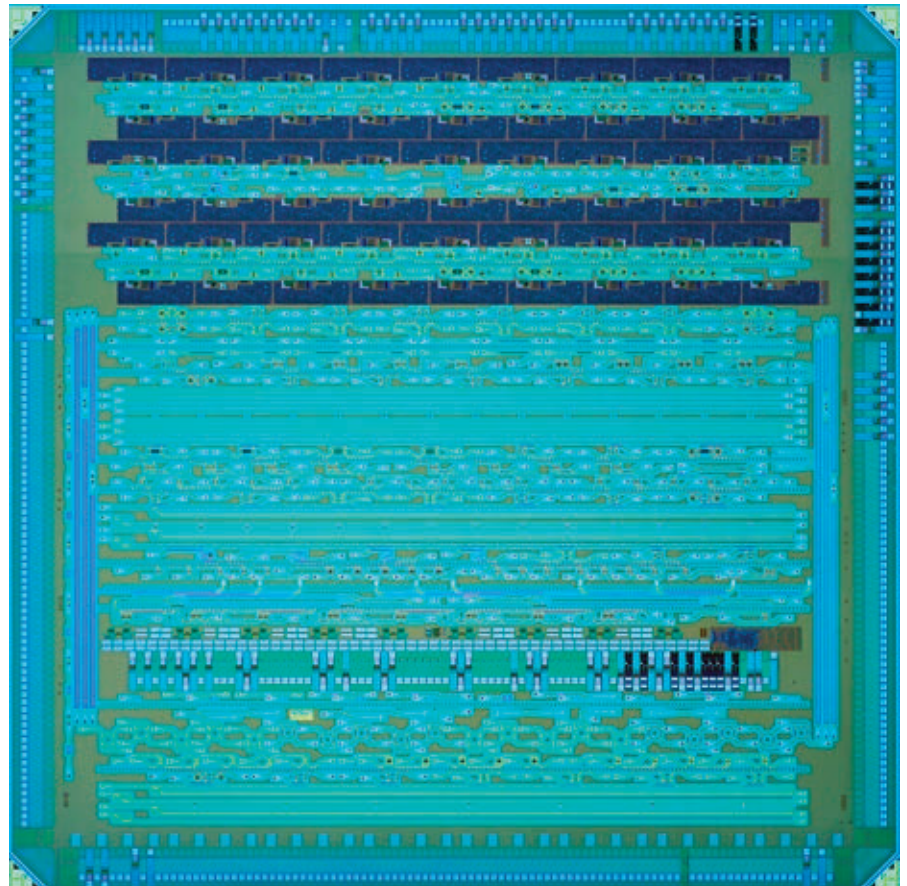
Still Seeking the Optical Transistor

Optical information handling is a critical staple for communications and the Internet, but using light for computer-scale computation remains a distant dream.

IT'S AN INTRIGUING IDEA: a transistor that uses photons of light to perform computations, instead of the electrons used today. "By the end of the decade, supercomputers could be using more light, or 'photonic,' components than electronic, and may run at least 100 times faster than today's generation," said Alan Huang of Bell Labs in Holmdel, NJ.

However, that quote was found in an Associated Press story from 1990, and the intervening quarter-century has not been kind to that bold projection (for one thing, Bell Labs closed its Holmdel site in 2006). The major reason, of course, is that electronics have continued to improve exponentially according to Moore's Law, leaving optical transistors and other once-promising alternatives eating their dust. By the year 2000, computer clock rates had indeed increased by nearly a factor of 100, but without any help from optical transistors.

To be sure, optical information handling is a critical staple for communications and the Internet, but using light for computer-scale computation remains a distant dream.



A test chip developed at the Massachusetts Institute of Technology in 2012, which monolithically integrates electrical and optical components.

Nonetheless, there is a perennial crop of journal articles describing new or improved “optical transistors” using a variety of approaches.

The most mature devices target opportunities in processing optical signals for communications, where optics already provide clear advantages. Yet for computation, the challenge is to find applications where light-based devices could beat conventional electronics at the modest scale that is practical for a nascent technology. Even then, the devices would need critical properties that permit them to be reliably assembled into complex systems. Without a clear view of the entire system that could exploit them, it is difficult to estimate the potential.

Optics versus Electronics

Encoding information in light differs profoundly from doing so using charge. For one thing, electrons exert strong forces on each other, making it easy to manipulate them and to arrange the interactions that enable logical operations. By contrast, photons of light travel long distances without interacting, which makes massive parallelism easier, but means that logically combining two light signals generally requires an intermediary material. Although computing at the speed of light seems impressive, the electromagnetic impulses in ordinary computers propagate essentially as fast.

What actually slows electrical signals is the time needed to charge the wiring capacitance. This charging also stores energy that is usually discarded when the next signal comes along. Because the energy of photons does not depend on how far they travel, they have a power advantage for long-distance connections where the savings overcome the energy costs of generating them in the first place.

In the decades following the invention of the laser in 1960, researchers had high hopes for optical devices that exploited the special advantages of light, such as free-space propagation. Early demonstrations included correlators that scanned an entire image simultaneously for a particular pattern (like a piece of artillery), but these optical analog systems, like their electronic cousins, succumbed to the digital onslaught, with its more

Encoding information in light differs profoundly from doing so using charge.

robust and flexible design.

In telecommunications, light transmitted through optical fibers has completely replaced copper wiring for carrying long-distance signals. Telecom providers also regularly amplify and clean up optical signals directly in transit, without converting them to electrical signals. Advanced systems even separate different wavelengths of light and route them to different destinations, for example by sending them through two-dimensional arrays of waveguides built on semiconductor wafers, without the energetically costly optical-electrical-optical conversions.

Communications systems require far fewer optical devices than would be needed for a general-purpose computer. In a 2010 commentary in *Nature Photonics*, David Miller of Stanford University cautioned that research devices almost never meet all the requirements for robust design and operation of large systems. These include cascability and fan-out, which ensure the output of a device can provide the input for multiple others, restoration of the signal in both level and quality, and isolation of the input from the output. In addition, most proposed devices are profligate energy consumers.

Still, in recent years researchers have devised many new ways to manipulate light, such as photonic crystals, metamaterials, plasmonics, and even individual molecules. “We would be pessimists indeed not to believe these opportunities will somehow transform information processing,” Miller wrote, “but we will need to be both realistic and creative to get there.”

Light on Light

Even in sophisticated communica-

tions systems, electrical control signals determine how the light is manipulated. In the long-sought optical transistor, the control comes instead from a light signal, which the device leverages to modulate a stronger light source that can be likened to an electrical power supply.

One widely used mechanism for modulating light intensity takes advantage of the coherent, well-defined waves emitted by lasers. A beam is first split in two parts that are sent on two different routes; for example, through separate threadlike waveguides on a semiconductor wafer. When they are brought back together, the waves interfere, either enhancing or diminishing their power. Making a small shift in the phase of one of the beams can then turn the output on and off. Optical modulator devices used in telecommunications typically create this shift using electric fields or heat.

Even with this sensitive mechanism, however, the direct effect of a third light beam on the usual waveguide materials is too small to make a practical transistor. Seng-Tiong Ho of Northwestern University in Evanston, IL, and his collaborators devised an optically switched device based on a related two-waveguide device called a directional coupler. They calculate that incorporating a semiconductor material into one waveguide allows another beam to change its gain or loss, and thereby switch a more powerful beam between the waveguides. Ho hopes such devices could allow faster interconnections between, for example, microprocessors. For this application, “it doesn’t have to be very complicated” to be useful, he notes.

Minghao Qi and his colleagues at Purdue University in Lafayette, IN, use a different interference effect that occurs in waveguides that are wrapped into a circle to form a “ring resonator.” If the ring circumference is a multiple of the wavelength, light traveling in a nearby waveguide will be captured in the ring. Heating the waveguide with weak light in a second waveguide can disrupt this delicate match so the original light instead passes by freely. The device, which can be made with standard silicon-processing techniques, shows optical gain and isolation of

the input from the output, both critical for larger circuits. Such optically gated devices might function in spite of electromagnetic interference, Qi suggests, or they could be used in artificial neural networks. “It’s important for us to understand what technology potentially has any niche applications that may allow it to survive and then it can grow.”

One possible niche is extending communications, where optics already beat electronics. Research by semiconductor giants like IBM and Intel, as well as projects funded by the U.S. Defense Advanced Research Projects Agency (DARPA), has demonstrated compact, high-speed optical devices directly integrated with silicon electronics. So far, electrical interconnections still dominate, not just within chips, but even between chips on a board. Nonetheless, optical communications are appearing in some advanced systems for interconnecting boards, and are likely to displace electronics at ever-shorter distance scales.

To provide new options for merging optics and electronics, Swastik Kar of Northeastern University in Boston, MA, teamed up with nanotube expert Young Joon Jung to build optically switched devices on a semiconductor wafer. Although the output of these devices was electrical current, not light, the team demonstrated logical operations combining light and electrical inputs, including a two-bit adder and a four-bit analog-to-digital converter. “We believe that this kind of device,” Kar says, “will allow a better merging of the processing and transfer of information.” An important feature of the nanotube device is that the current in the off state is reduced by a factor of 10^5 , much more than for typical photodiodes. Such small off-currents are critical when large numbers of devices are used on a chip.

Seeking a New Niche

Some researchers see an opportunity for optics in a technology that does not yet exist: quantum computing using quantum bits (qubits) that exist in multiple states simultaneously. “If you had 100 qubits, you could be competitive or better than current classical computers for certain types of calculations,” says Vladan Vuletić of the Massachu-

setts Institute of Technology in Cambridge, MA. He stressed, “You don’t need enormous numbers.”

Vuletić and his colleagues have made the ultimate optical transistor, in which a light beam is turned on and off by the absorption of a single photon. To do this, they placed a cloud of about 20,000 cesium atoms in an optical cavity and cooled them to a few millionths of a degree above absolute zero. Because the light bounces back and forth hundreds of thousands of times, the absorption of a single photon of a second beam by one of the atoms changes the resonance enough to turn the beam on and off.

Of course, this apparatus is not very practical, even for a dedicated supercomputer. “To really make multiple quantum gates to do even some simple calculations with photons, probably these systems that we use now are too bulky,” Vuletić acknowledges. He and others have recently demonstrated much smaller resonators could be assembled together over the next five years or so.

Still, it will be years, if ever, before any type of quantum computing becomes established, and even then other non-optical schemes may be used for implementing the qubits. So for now, computing with light is still waiting for its time to shine. ■

Further Reading

Krishnamurthy, V., Chen, U., Seng-Tiong Ho
“Photonic transistor design principles for switching gain ≥ 2 ,” *Journal of Lightwave Technology* **31**:2086 (2013).

Varghese, L., Fan, L., Wang, J., Gan, F., Niu, B., Xuan, Y., Weiner, A., Qi, M.
“A Silicon Optical Transistor,” <http://arxiv.org/abs/1204.5515>

Kim, Y., Jung, H., Park, S., Li, B., Liu, F., Hao, J., Kwon, Y., Jung, Y., Kar, S.

“Voltage-switchable photocurrents in single-walled carbon nanotube–silicon junctions for analog and digital optoelectronics,” *Nature Photonics* **8**, 239 (2014)

Chen, W., Beck, K., Bückler, R., Gullans, M., Lukin, M., Tanji-Suzuki, H., Vuletić, V.

“All-Optical Switch and Transistor Gated by One Stored Photon,” *Science* **341**, 768 (2013).

Don Monroe is a science and technology writer based in Brookline, MA.

© 2014 ACM 0001-0782/14/10 \$15.00

Milestones

Computer Science Awards

GELENBE RECEIVES DENNIS GABOR AWARD

Erol Gelenbe, an ACM Fellow and recipient of the ACM SIGMETRICS Achievement Award in 2008, has been awarded the “In Memoriam Dennis Gabor Award” from the NOVOPER Foundation for Technical and Intellectual Creation of the Hungarian Academy of Sciences, for outstanding research with important impact in innovation.

Gelenbe has contributed fundamental results to stability and control of random access communications. His practical inventions include the design of the first random access fiber-optic local area network, a patented admission control technique for ATM networks, a neural network-based anomaly detector for brain magnetic resonance scans, and the “cognitive packet network” routing protocol to offer quality of service to users.

KAMINKA AWARDED LANDAU PRIZE

Gal Kaminka of Bar-Ilan University’s Department of Computer Science and the Gonda Multidisciplinary Brain Research Center has been awarded the Landau Prize for Arts and Sciences in the robotics category.

Kaminka is a leading contributor to intelligent robotics, the science of using artificial intelligence to make robots smarter.

He said his goal “is to understand social intelligence; to understand the transition from a single mind to many; to build robots that are socially-intelligent; that are able to reason about, manipulate, collaborate with, and coordinate with other robots and humans; and to build computational models that explain social intelligence, that allow replication of it, that facilitate predictions of its occurrence, and that enable measurement and quantification.”

Landau awards celebrate Israeli scientists who achieve breakthroughs in their fields and contribute to the advancement of science.

Gradual Evolution

Dynamically typed languages adopt features of static typing to cope with growth.

WHEN BRENDAN EICH created JavaScript over the course of several weeks in 1995, his aim was to make it easy to write small applications for Netscape Navigator 2.0, one of the early browsers for the newly emerging World Wide Web. He probably did not envision that, two decades later, the language would be one of the most widely used on the Web and that the programs written in it would have tens of thousands or even a million lines of code.

“If you want to write an app and have it run in the widest number of places and on the widest type of devices, you pretty much have to write it in JavaScript,” says Anders Hejlsberg, a Technical Fellow at Microsoft Research.

However, such widespread use brings problems. “It was never intended for large programs,” Hejlsberg says. “Apps are getting bigger and as they get bigger, they get harder to maintain.”

The issue is that JavaScript is a dynamically typed language, as are several other popular programming languages such as Perl, Python, and PHP. Such languages do not require developers to define every variable type as they go, and the system checks the program for errors at runtime. That makes it easier to write and rewrite an application quickly, but it also means bugs can remain hidden for months or even years, until a particular execution of the program trips over the error and crashes. As Hejlsberg puts it, in a dynamically typed language, you do not discover the flaws while you are building the space shuttle; you find them once it is flying.

The “type” in “dynamically typed” is a set of values upon which certain operations may be performed. For in-

The intent of TypeScript, an optionally typed language, is to allow developers to choose how much static typing they want to incorporate into their programs.

stance, the type may be integers, and the operations that can be performed on that type would include addition, subtraction, multiplication, and so on. Programs work on a broad variety of types, and problems arise when an operation is applied to a type for which it was not intended. If the type is, say, a string of names, trying to multiply them will not work.

Traditionally, languages have been either statically or dynamically typed. In statically typed languages, such as Java or C++, a type checker runs during compilation of the program and can catch many errors in the program. It does this by looking at annotations in the program. An annotation is a form of metadata that specifies which function is called for at for a given variable; that makes it easier for the type checker to ensure a line of code can actually do what it says it wants to do. Typically, statically typed languages increase development time but are safer—that is, less apt

to crash—in the long run.

In an attempt to combine the best of both systems, computer language experts have been busy developing static type systems that can be used along with popular dynamically typed languages. Hejlsberg’s attempt to keep his metaphorical space shuttle from blowing up after launch is called TypeScript, a superset of JavaScript. TypeScript overlays aspects of a static type system onto JavaScript to cope with the complexities of ever-larger programs. Earlier this year, Facebook launched Hack, a version of PHP that includes static typing. Typed Racket adds static typing to Racket, a dialect of Scheme, and mypy does the same for Python.

“The whole purpose of a static type system is to build a model of what is going to happen when it runs,” Hejlsberg says. “It’s almost as if you have the entire reference manual for what you’re doing at your fingertips, and the compiler just looks it up for you as you go along.”

No matter how they start out, programs tend to get bigger, and as they do there is more room for errors to creep in. “It’s the inevitable progression of a piece of software,” says Benjamin

Pierce, a professor of computer and information science at the University of Pennsylvania. Many projects

begin as a single person trying to write an app to do one particular task, Pierce says. “Pretty soon, you have 10 people adding things to it and making it better and better and it’s doing things the original person never thought of and it’s getting out of control.”

Static typing makes altering the program easier, because when one item is changed, the type checker can point to all the other instances where it would need to change; in a dynamically typed program, the developer would have to hunt through all the code for those



instances. It also makes possible the use of tools, such as autocomplete, that make life easier for developers.

Though a programmer in an Integrated Development Environment (IDE) sees the annotations in TypeScript, they are stripped away when the program is compiled. “To the browser, it just looks like plain old JavaScript,” says Hejlsberg. TypeScript also incorporates JavaScript libraries, such as JQuery, to help make development smoother. As an open source project, it also takes advantage of developments such as DefinitelyTyped, a compilation of type definitions created by various people.

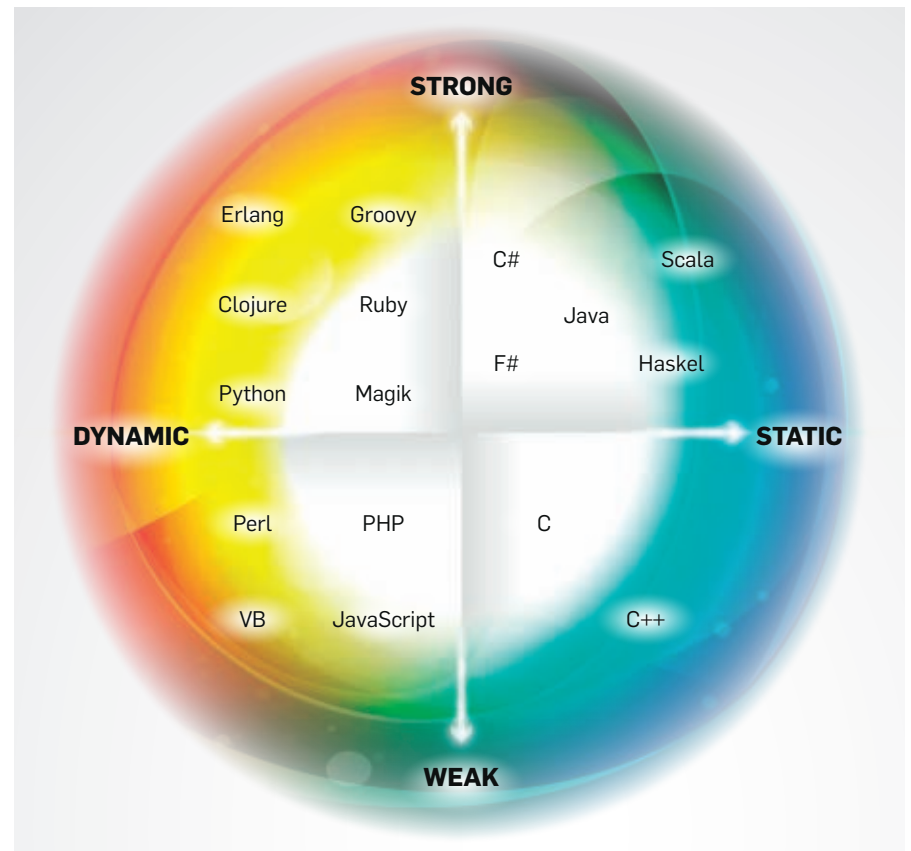
TypeScript 0.8.0 was released in October 2012, and version 1.0 came out in April of this year. Hejlsberg says he and his team are already looking toward ECMAScript 6, the upcoming version of the standards behind JavaScript, and planning to incorporate its new features.

A Matter of Choice

Hejlsberg says the intent of TypeScript is to allow developers to choose just how much static typing they want to incorporate into their programs; it is what he calls an optionally typed language. “Typing was sort of a switch. It was either on or off,” he says. “Now with Typescript, we’ve turned that switch into a dial.”

Jeremy Siek, an associate professor of computer science at Indiana University, developed a similar type system called gradual typing along with Walid Taha, a professor of computer science at Rice University and at Halmstad University in Sweden. “If you don’t put any type annotations anywhere, it really behaves like a dynamically typed language,” Siek explains. “As you add more type annotations, more and more things start to get checked.”

Gradual typing allows developers to decide what is important to them when writing a program. If they want something quick, in which they can check how pieces of the program run before the whole thing is completed, they can skew toward dynamic typing. If they need to be sure they are rooting out errors—say they are programming



The continuum of development languages.

something critical, such as a pacemaker—they can rely more on static typing to make sure the code is safe. “There’s this trade-off between safety and flexibility, and it (gradual typing) lets the programmer choose,” Siek says.

Indeed, there have been solid reasons to prefer dynamic typing, says Julien Verlaguet, a software engineer at Facebook who led the development of Hack, which is supplanting PHP as the code underlying the social networking site. When the company started out, Facebook developers had relied on the dynamically typed PHP because it provided rapid feedback; they could modify code and quickly see how that changed the user experience. “The only way to test if something feels right is to interact with it,” he says. With a statically typed language, they would have had to wait minutes for the code to be compiled. “The compilation time basically kills the interaction for the developer.”

Facebook fixed the time lag problem by introducing HVVM, a virtual machine that acts as a fast compiler. In March, Facebook introduced Hack, a gradual typing language. In general, Verlaguet says, static typing is preferable, both because of the safety of the resulting code and the efficiency provided by the development tools it makes possible. Developers would have statically typed much of the code in the first place if they had that option, says Verlaguet. “It’s just that there wasn’t a type checker in place to enforce it.”



If a code base gets large enough, however, there will be scenarios in which dynamic typing works better because it provides different options for defining terms, so allowing the developer to choose which to use makes sense, he says.

To provide options for Python programmers, Jukka Lehtosalo started developing mypy while he was a Ph.D. student in computer science at the University of Cambridge. “You can give it any Python code and



it checks whether it's consistent with static type," Lehtosalo says.

"Basically, mypy is just Python; or, strictly speaking, a subset of Python," he says. "In mypy, you can use any Python IDE, and it works because it's valid Python."

This is not the first marriage of Python and static typing. Cython was created in 2007 as a superset of Python that allowed it to use types from C and C++. Lehtosalo says that where Cython was aimed at improving performance, mypy is more about enhancing developers' productivity and improving the quality of the code.

Evolution Continues

As a one-man project, mypy is not as far along as some other gradual type systems. Lehtosalo, whose job at Dropbox is unrelated to this work, hopes to keep moving the language from experimental to something widely useful.

Because both mypy and TypeScript remove the type annotations before running, they are giving up one method to make certain the code is more efficient. It is possible to use type in-



formation to speed up programs, says Lehtosalo, and that may happen with a future version of mypy. A language that did that, however, would be a new variant, and thus wouldn't be 100% compatible with Python.

Hejlsberg points out that, while annotations technically are not used to improve performance at runtime, the checking they provide at an earlier stage increases the likelihood the compiler will generate efficient code.

Whether gradual typing will become the major programming paradigm remains to be seen. Google has been promoting gradual typing language Dart to replace JavaScript, but adoption has been slow. Meanwhile, in June Apple introduced Swift, which is statically typed, as a successor to Objective C.

Still, languages are adding gradual typing, from both ends of the spectrum. For instance, C# 4.0, originally developed by Hejlsberg, added dynamic type to that previously static language. "I do think we are seeing a gradual breaking down of the traditional taxonomy of languages," says Hejlsberg.

"I've certainly been happy to see a lot of the newer languages come out with gradual typing," Siek says. "My hope is that more and more of them will decide to take the middle ground, or take the let-the-programmers-decide approach, and will be more gradual." □

Further Reading

Inside Typescript
<http://bit.ly/1pW7xNX>

Siek, J.G., Taha, W.
Gradual Typing for Functional Languages, Scheme and Functional Programming 2006, Portland, OR.

Pierce, B.C.
Types and Programming Languages, MIT Press, Cambridge, MA, 2002.

Meijer, E., Drayton, P.
Static Typing Where Possible, Dynamic Typing When Needed: The End of the Cold War Between Programming Languages, OOPSLA'04 Workshop on Revival of Dynamic Languages, 2004.

Wright, A.
Type Theory Comes of Age, CACM 53 (2), February 2010.

Neil Savage is a science and technology writer based in Lowell, MA.

© 2014 ACM 0001-0782/14/10 \$15.00

Milestones

IMU Lauds Khot for Unique Games Theorem

The International Mathematical Union (IMU) recently awarded the prestigious Rolf Nevanlinna Prize to Subhash Khot, a professor in the computer science department at New York University's Courant Institute of Mathematical Sciences, "for his prescient definition of the 'Unique Games' problem, and leading the effort to understand its complexity and its pivotal role in the study of efficient approximation of optimization problems."

The Nevanlinna Prize recognizes outstanding contributions in all mathematical aspects of computer science, as well as in scientific computing and numerical analysis. The award is targeted at young mathematicians, who must be under 40 years of age on Jan. 1 of the year the award is given.

The IMU noted Khot's work has "led to breakthroughs in algorithmic design and

approximation hardness, and to new exciting interactions between computational complexity, analysis and geometry," adding that it "will be driving research in theoretical computer science for many years to come."

Lance Fortnow, chair of the School of Computer Science at the Georgia Institute of Technology, says the IMU's recognition of Khot could help draw more attention to these types of problems. "People within the complexity field know of Khot's work, but I think it generates more attention from mathematicians," Fortnow says.

Khot's award-winning work is based on his 2002 theorem known as the Unique Games Conjecture (UGC), which asserts the problem of determining the approximate value of a certain type of game, known as a unique game, has Non-deterministic Polynomial-time hard (NP-hard)

algorithmic complexity.

Most computer scientists believe so-called "NP-hard" problems cannot be solved exactly by any algorithm that runs in a reasonable amount of time. An example of this type of problem would be arranging a seating plan for a wedding, where a set of constraints (such as feuding family members) would add to the complexity, due to the large number of possible solutions.

Khot's theorem addresses these types of NP-hard problems through the use of the Network Coloring problem, which considers a network of nodes and a set of colors, and asks whether it is possible to color the vertices of the network in such a way that two vertices that share an edge always have different colors.

A key issue that has been raised in the scientific community is that UGC remains unproven. However, its influence on other

research areas has already been felt, according to Fortnow.

"The neat thing about the research is how well it ties in with so many other research directions that are going on," explains Fortnow. "The closest connection is with semi-definite programming, which is a relatively new algorithmic technique to get a better handle on some of the hardest problems in computer science, which deals with approximate solutions. Unique Games Conjecture really seems to capture the hardness of this semi-definite programming technique, and shows us the limits of what we can do with SDTs (software development tools)."

Fortnow notes that UGC is also impacting, although to a significantly lesser degree, research in the areas of coding theory, quantum computing, and voting theory.

—Keith Kirkpatrick

Museums Go High-Tech with Digital Forensics

Scientists are using cutting-edge scanning and visualization techniques to wow visitors and find new stories in ancient artifacts.

IN 1887, A uniquely shaped and mysterious mummy caught the eye of an important museum collector and hitched a ride from the storehouse of an unknown antiquities dealer in Egypt to the vast vaults of the British Museum in London.

No question, it was a woman. Unlike most of the cases carrying the wealthy dead of ancient Egypt, which depicted their inhabitants in their embalmed state, her case was painted like a woman in the prime of her life. Metal eyes gazed out from a wooden facemask. A painted dark cloak covered the body of the case, and from its base, reddish feet with silver-painted toenails emerged. Who was she?

The first revelations came in the 1960s, when British Museum staff blew the dust off their mummy cases and ran several of them whole through X-ray scanners. Experts were surprised to find

Using computer tomography (CT) devices, researchers could now explore their mummies layer by layer, without touching the case or wrapping.

one-third of this case empty, and the rest filled by the preserved remains of a tiny female. The case was fashioned like the shapely curves of a full-grown woman, but carried a girl. Museum researchers concluded she was no older than 12.

In 1975, the coffin was cleaned and inscriptions found on the case included the name of the dead girl: Tjayasetimu, temple singer. This made sense; the children of wealthy families traditionally took these roles. Also, child mummies were a rare find. Tjayasetimu would not have been preserved unless her parents could afford the lavish last rites.

It was not until the mid-2000s that British Museum staff really got a more-complete picture of this girl's identity. By then, museums switched to scanning their mummies with medical computer tomography (CT) machines. X-ray scans revealed bony outlines, but CT searched for density. Without touching the case or wrapping, researchers could now explore their mummies layer by layer.

Under Tjayasetimu's wrappings, researchers were surprised to see the



A British Museum staffer views scans of the mummy of 'Gebelein Man B,' part of the interactive 'Ancient Lives, New Discoveries' exhibition that incorporates state-of-the-art technology enabling visitors to virtually explore inside mummy cases and examine bodies underneath the wrappings.

delicate features of an exquisitely preserved face, and clumps of hair that would have hung to her shoulders. The scans revealed adult teeth in her jaws, waiting to push out, had she lived for another year.

“The embalmer has done a brilliant job,” says John Taylor, a curator at the British Museum specializing in ancient Egypt and funerary archaeology. From the length of her leg bones, the teeth in her jaw, and the styling of her hair—all viewed through CT scans—Taylor and his colleagues concluded Tjayasetimu was probably seven, maybe pushing eight, when she died.

“That was quite a revelation to be able to pinpoint her age with that level of accuracy,” Taylor says, and much of this was possible only because of advanced CT scanning technology, and the 3D visualizations that followed.

Today, Tjayasetimu rests beside seven other mummies in a new exhibit at the British Museum. Visitors still stand behind the glass of the display case, but on touchscreens mounted beside each, they can take a virtual tour inside the mummy cases, zooming in and peeling back layers of wood, plaster, and linen.

The British Museum is not alone in laying thousand-year-old artifacts next to cutting-edge technology in its latest public exhibits.

Museums all over the world are using a variety of scanning techniques to study fragile historical objects and uncover secrets that were previously out of their reach.

The techniques have filled in stories about individuals and whole cultures. Also, digitizing samples allows more researchers access for longer periods. On display, the new technology offers museum visitors a chance to share in an explorer’s moment of discovery.

“This is the closest you’re doing to get to the ‘eureka’ moment of seeing something unexpected,” says J.P. Brown, Regenstein Conservator at Chicago’s Field Museum. “There’s something about being able to manipulate the specimens which is interesting and exciting. I don’t think you get the same sense of discovery from looking at illustrations.”

Leading the scanning team at the Field Museum, Brown has had his share of ‘eureka’ moments.

In 2010, the team was scanning a female Peruvian mummy, arranged, as

these traditionally are, in the fetal position. As the first scans showed up, they saw a rounded object, which looked like a broken vegetable gourd traditionally packed with stored food. When they looked closer, they realized that gourd was the disconnected skull of a dead infant, probably buried with its mother. Though the Field Museum has seen a number of “multiple burial bundles” (many individuals wrapped within the same mummy casing), this was the only one they have found with a baby in it, Brown says.

In the spring of 2012, the Field Museum unveiled a new mummy exhibit titled “Opening the Vault.” As at the British Museum, exhibits were accompanied by large touchscreens. Visitors looking at the real mummy from behind the display glass could virtually explore the insides of the box.

“You just don’t realize when you look at it in drawings, how much of an Egyptian mummy is linen wrappings. It’s a heck of a lot of wrappings,” Brown says.

Sliced like Salami

Archaeologists were using CT scanning as a research tool back in the mid-1950s, but it wasn’t until the graphics-processing and 3D-visualization revolutions of the mid-2000s that museum staff realized the data could be put on display.

Brown explains that a CT scanner is “basically a salami slicer” capturing images of virtual sections of the insides of a carefully wrapped mummy. After the flat slices have been stored by the computer, it is the job of a person—part art-

Museums all over the world are using a variety of scanning techniques to study fragile historical objects and uncover secrets that were previously out of their reach.

ist, part programmer, part researcher—to stitch together the two-dimensional data into a colorful, engaging interactive exhibit, or a life-like rendering that gives researchers a realistic view of the insides of the object.

Among the leaders of this craft is the Interactive Institute Swedish ICT, an experimental information technology and design research company that has been converting technology geared for hospitals and clinics into teaching and research tools.

Staff at the Interactive Institute collaborated with the Field Museum to build the displays for the “Opening the Vault” exhibit. The Interactive Institute has been involved with 15 other interactive exhibits across the world, at places like London’s British Museum and the Smithsonian Institution in Washington, D.C.

The group has scanned everything from mummies to Martian meteorites to geckos, says Thomas Rydell, studio director, who adds that the real challenge comes later: “When you have this scan, you have to decide what you want to tell your audience about it.” A visitor only stops at a display for two or three minutes, so what is presented “has to be super-intuitive.”

At Stockholm’s Museum of Mediterranean and Near Eastern Antiquities museum, visitors can get even closer to history. One of the museum’s eight resident mummies is a 2,300-year-old Egyptian priest named Neswau, an important man preserved in an ornate and intricately decorated mummy case. When staff scanned Neswau, the CT scans revealed several amulets placed inside the wrappings, positioned at strategic locations on his body. The scans picked up the details of a falcon-shaped amulet, which the team then converted into 3D-printed replicas.

Now, as museum visitors view Neswau’s mummy from behind the glass of the display case, they can run their fingers over a printed amulet identical to the one touching the priest’s body.

There is a thrill to that, Rydell says. “I’m holding something in my hand that someone placed under the wrapping several thousand years ago.”

Age of the ‘Laser Cowboys’

Museums all over the world are reach-

ing for that same accessibility, relying on 3D scanning and printing techniques to achieve realistic replicas of ancient objects.

At the Smithsonian Institution's National Museum of Natural History, laser-assisted 3D scanning and replication is under way on a grand scale.

In November 2013, the museum launched the Smithsonian X 3D collection on its website, with the goal of making scores of artifacts from the museum's collections, only 1% of which are on display, accessible anywhere in the world. As a result, for example, a flight history enthusiast from Auckland, New Zealand, can use a home computer to view a visualization of the Wright brothers' 1903 airplane from any angle, or zoom in to view the seams and detailing on Amelia Earhart's flight suit.

Unlike the layer-by-layer exploration of mummies with CT, the Smithsonian's laser scans only scour the surface of an object.

Besides giving visitors a great way to explore the Institution's trove of age-old wonders, the tools are valuable to researchers as well.

In June, the museum revealed the largest printout of its data yet: a scale model of a whale fossil, embedded in the ground exactly as it was when Smithsonian paleontologist Nicholas Pyenson first saw its bones peeping out of a roadcut in the Chilean desert in 2011. "The whale fossil is a great example of how we can take actual scientific data from the field and replicate it for a museum audience," explains Gunter Waibel, director of the Digitization Program Office at the Smithsonian Institution.

The fossil trove, which contained 40 dead baleen whales between six million and nine million years old, was uncovered by a road-building company midway through expanding the Pan-American Highway. The road was growing, and the paleobiological treasure trove was in jeopardy so, the story goes, the Smithsonian called in its scanning specialists, Vince Rossi and Adam Metallo, 3D Program Officers who have come to be known as the "laser cowboys." As *Smithsonian Magazine* would later report, the team took an emergency trip to Chile and spent two weeks scanning the fossils in the very same soil layers in which

In June, the museum revealed the largest printout of its data yet: a scale model of a whale fossil, embedded in the ground exactly as it was found.

they were found, preserving clues that paleontologists would need in reconstructing how the whales got there in the first place.

The brand-new 3D-printed exhibit, a product of those first scans, is a happy bonus.

Ge-whiz displays and lush 3D renderings may seem like they are stealing the spotlight from genuine, original historical objects, but that is not the case, Waibel insists. The museum's goal is to make the collections and specimens more accessible to the public and to provide "an alternate way to interact with history, science, and art," says Waibel.

"Nothing will ever replace the experience of viewing an original artifact in person," he says. ■

Further Reading

Taylor, J., Antoine, D.
Ancient Lives New Discoveries: Eight Mummies, Eight Stories, *British Museum Press*, London, England, 2014.
<http://amzn.to/YAqfAO>

Pyenson, N., et al.
Repeated mass strandings of Miocene marine mammals from Atacama Region of Chile point to sudden death at sea. *Proceedings of the Royal Society, B*, 281, February 2014.

Reconstructing the mummy Neswau with Autodesk ReCap
<http://bit.ly/1pLvqYG>

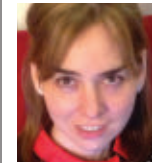
"Cerro Ballena." Cerro Ballena. Smithsonian Institution, n.d. Web. 11 July 2014.
<http://cerroballena.si.edu/>

Nidhi Subbaraman is a freelance science and technology writer based in New York City.

© 2014 ACM 0001-0782/14/10 \$15.00

ACM Member News

KATABI AIMS MACARTHUR GRANT AT WIRELESS NETWORKING



Last September, Dina Katabi was shocked to learn she had been selected to receive the MacArthur

Foundation 2013 Computer Science Foundation Fellowship and a \$625,000 grant to further her research in 802.11 wireless networking.

"I thought it was a prank. I didn't even know I had been nominated," laughs Katabi, a professor in the Massachusetts Institute of Technology (MIT) Department of Electrical Engineering and Computer Science.

MacArthur Foundation fellowships are awarded to researchers who have made "a past impact and are projected to make a future impact" in their fields. That encapsulates the Syrian-born Katabi's efforts over the last 15 years to "improve the robustness, performance and data rates of computer networks and to utilize wireless and radio signals to open up new applications."

Katabi received her B.S. from Damascus University, then moved to the U.S. to study computer science at MIT, where she earned her Master's degree in 1999 and her Ph.D. in 2003; she joined the faculty, as director of the MIT Center for Wireless Networks and Mobile Computing, in 2003. In MIT's Computer Science and Artificial Intelligence Laboratory, Katabi works to solve the fundamental interference issues caused by the proliferation of wireless networks and traffic. She developed a "ZigZag" algorithm that reconstructs the contents of damaged packets; it salvages and combines the usable fragments, significantly lowering retransmission rates.

Katabi also is fine-tuning a prototype wireless device called Wi-Track, capable of monitoring people's movements behind walls and doors. "This has incredible implications for a wide range of applications, like health care, gaming, and virtual reality," Katabi says.

—Laura DiDio



DOI:10.1145/2661047

Michael A. Cusumano

Technology Strategy and Management

The Bitcoin Ecosystem

Speculating on how the Bitcoin economy might evolve.

BITCOINS^a CONTINUE TO attract attention but remain somewhat difficult to understand (see the May 2014 *Communications Economic and Business Dimensions* column by Marshall Van Alstyne, “Why Bitcoin Has Value”). They are by far the most used of several hundred “crypto-currencies.” Theoretically, they are free and available to anyone with a computer and an Internet connection. They could replace cash, credit and debit cards, money transfers, and currency conversions through banks and other financial intermediaries. They represent another potentially disruptive Internet technology. But are they truly free and easily available? Not really. For most of us, bitcoins are a complex platform technology that requires the help of intermediaries—an ecosystem of “complementary” product and service providers that charge fees.

The software solves the dilemma of how two or more people who do not

Bitcoins are currently less like a currency and more like a computer-generated commodity.

know each other can establish trust and mutually agree upon a transaction: the so-called “Byzantine Generals Problem.” The technology first appeared in 2008 after its creator (or creators) known as Satoshi Nakamoto made the software freely available.² Bitcoin generates a public ledger (a “block chain”) when anyone creates, buys, or transfers a bitcoin. No one can change this ledger, and anyone can see it by downloading the open source peer-to-peer software. This scheme offers significant protection against theft or counterfeiting. However, because the ledger does not

contain names or physical addresses, some criminal use of bitcoins to transfer and store money has occurred.⁵ (Anonymous overseas bank accounts might be worse in this regard.)

Bitcoins are currently less like a currency and more like a computer-generated commodity. Those who solve calculations receive blocks of bitcoins in decreasing size as more are produced. The value of a single bitcoin, which has ranged from fractions of a penny to over a thousand dollars, comes primarily from speculators. There were about 12.5 million bitcoins in mid-2014, valued at approximately \$7 billion (just under \$600 each).^b Unlike currencies that governments create, Bitcoin software limits the supply to 21 million units, a number that might be reached around the year 2140. Yet bitcoins can be divided into fractions and more fractions, which are especially useful to conduct micropayments over the Internet, such as for music or newspaper articles. In principle, the sup-

^a The technology is referred to in uppercase singular while lowercase and plural or “BTCs” refer to the crypto-currency units.

^b See <http://bit.ly/1B9acsG> and <http://www.coindesk.com/price/>.



ply of bitcoins could be large enough to function as a currency substitute. Moreover, bitcoin users can avoid high transaction fees. Goldman Sachs estimates intermediaries siphon off as much as 10% of value for money transfers.⁴ Credit cards include 2% or 3% as charges to vendors ultimately paid by consumers, or about \$500 billion a year, excluding other card charges or currency conversion costs.⁷

Yet bitcoins are not yet ready for prime time. The technology is more like Windows, iOS, or Android in their early stages. Yes, Bitcoin brings multiple parties together for a common purpose—to make financial transactions over the Internet. But, like other new platforms, we need an ecosystem of complementary products and services to make the technology truly easy to use (see my January 2010 *Communications* column, “The Evolution of Platform Thinking”).

The Bitcoin ecosystem must solve several common platform-market problems. One is the “chicken-and-egg” dilemma: If more vendors accept bitcoins, more people will use them; and if there are more bitcoin

users, vendors are more likely to accept them. Conversely, vendors will not accept bitcoins unless more people have them, and people will not use bitcoins unless vendors accept them. New operating systems and credit cards all faced this dilemma. In addition, the supply of bitcoins is limited by a numerical ceiling and algorithms that increase in complexity each time someone generates another bitcoin. Creating bitcoins today therefore requires big investments in computers and electric power. Limited supply causes prices to rise and fluctuate due to speculation. Buying bitcoins from people who already own them requires transferring money, but many banks that would provide such services refuse to handle bitcoins.¹⁴ If more governments join China to outlaw bitcoin payments or impose capital gains reporting on sales of bitcoins, as the Internal Revenue Service does in the U.S., then bitcoin use could be severely compromised.^{1,12}

Bitcoin entrepreneurs follow the value chain. The first step is mining, creating more BTC units from scratch, and then we have storing bitcoins in

virtual “wallets.” The wallets resemble smartphone apps with the equivalent of digital bar codes to hold keys and transaction information. To buy or convert bitcoins to traditional currencies, a wallet company needs information from a user’s bank account or credit card, and then must interact with another step in the chain—payment processors or exchanges that convert bitcoins into other currencies or send bitcoins to vendors as payments. These companies generally charge between 0.5% and 1%. Finally, we need other companies to guarantee against losses from fraud, theft, or huge swings in value.

Mining companies have received mixed reviews when the costs are not always transparent. Cloud Hashing, founded by Emmanuel Abiodun in London in 2013, started by mining bitcoins on a PC, later moved to commercial-scale machines, and now rents out computing power and “mining contracts” as a service.^{6,10} It can cost more to generate bitcoins than they are worth on the market but, if value goes

c See <https://cloudhashing.com>.



Association for
Computing Machinery

ACM Conference Proceedings Now Available via Print-on-Demand!

Did you know that you can now order many popular ACM conference proceedings via print-on-demand?

Institutions, libraries and individuals can choose from more than 100 titles on a continually updated list through Amazon, Barnes & Noble, Baker & Taylor, Ingram and NACSCORP: CHI, KDD, Multimedia, SIGIR, SIGCOMM, SIGCSE, SIGMOD/PODS, and many more.

For available titles and ordering info, visit:
libraries.acm.org/pod



up, customers eventually win. Bitfury, founded in 2011 and now located in San Francisco, received \$20 million in venture funding to design application-specific integrated circuits (ASICs) for Bitcoin mining.^{d,15}

Payments processors and wallet firms are critical intermediaries and have also raised a lot of venture capital. Bitpay, founded by Tony Gallippi and others in 2011 in Atlanta, GA, has relationships with more than 30,000 vendors and has received some \$35 million in funding, led by Richard Branson (Virgin Atlantic), Jerry Yang (Yahoo), and Li Ka-Shing (Horizon Ventures).^c San Francisco wallet company Coinbase, founded in 2012 by Brian Armstrong and Fred Ersham, received \$25 million in venture capital from Andreessen-Horowitz and others. It holds about 20% of all bitcoin customer wallets (1.3 million, growing about 30% a month), charges 1% to convert into or out of bitcoins, and has relationships with more than 30,000 merchants.^{f,4,11} New entrant Circle Financial, founded in 2013 in Boston by Jeremy Allaire, formerly of Macromedia and Brightcove, has received \$26 million in venture capital. Circle Financial provides payment acceptance tools for merchants who must pay, as well as easy and free ways for consumers to buy, sell, store, or receive bitcoins.^{g,16} Bitstamp, the most prominent current exchange, was co-founded in 2011 by Nejc Kodrič, originally based in Slovenia and now operating out of the U.K.^h

Despite such successes, the largest exchange and wallet service, Mt. Gox of Tokyo, went bankrupt in 2014. Hackers exploiting poor accounting practices or a loophole in the block chain caused the company to lose somewhere between \$350 and \$500 million in bitcoins from customer accounts. There is a paradox here: The Bitcoin community has tried to get around traditional banks and government regulation but now they probably need banks and regulators if bitcoins are to become more secure and widely accepted.^{3,9}

d See <http://www.crunchbase.com/organization/bitfury>.

e See <https://bitpay.com>.

f See <http://www.coinbase.com>.

g See <https://www.circle.com>.

h See <https://www.bitstamp.net>.

Theft or loss from computer hardware failures or other risks has also created demand for insurance. U.K.-based Elliptic will store bitcoins and cover holdings against loss for a fee of 2%, similar to a private Federal Deposit Insurance Corporation.ⁱ Inscrypto is designing a hedge fund to protect against large swings in the value of bitcoins.^{j,8}

Finally, it remains unclear if the average consumer will treat bitcoins as a currency replacement or as a commodity for speculation. Users might pay less than credit card fees, but not much less, and the chicken-and-egg problem will be difficult to solve. To build some momentum, Circle Financial and Coinbase are giving away \$10 bitcoin accounts.^{k,6} The Massachusetts Institute of Technology (MIT) Bitcoin Club also raised \$500,000 to give each MIT undergraduate (there are 4,500) \$100 in Bitcoins in September 2014.¹³ What these new users do with their free accounts should provide some insight into Bitcoin's future. ■

i See <https://www.elliptic.co>.

j See <http://go.inscrypto.com>.

k See <http://bit.ly/Rf4oEz>.

References

1. Abrams, R. I.R.S. takes a position on Bitcoin: It's property. *The New York Times* (Mar. 26, 2014).
2. Andreessen, M. Why Bitcoin matters. *The New York Times Dealbook* (Jan. 21, 2014).
3. Galston, E. The Bitcoin paradox that undid Mt. Gox. *The Wall Street Journal* (Feb. 27, 2014).
4. Goldman Sachs Investment Research. All about Bitcoin. *Top of Mind*, Issue 21 (Mar. 11, 2014), 8.
5. Hays, T. In Bitcoin-aided crime, some see a digital wild west. *Boston Globe* (Feb. 17, 2014).
6. Keohane, D. Firm wants to help Bitcoin go mainstream. *The Boston Globe* (May 19, 2014).
7. Kessler, A. Angling to be the MasterCard of Bitcoin. *The Wall Street Journal* (May 17–18, 2014).
8. Manjoo, F. For Bitcoin, secure future might need oversight. *The New York Times*, (Mar. 5, 2014).
9. Matthews, C.M. New front in Bitcoin probe. *The Wall Street Journal* (May 20, 2014).
10. Popper, N. Into the Bitcoin mines. *The New York Times* (Dec. 21, 2013).
11. Popper, N. \$25 million in financing for coinbase. *The New York Times* (Dec. 12, 2013).
12. Ramzy, A. Chinese Bitcoin investors fret as value of virtual currency plunges. *The New York Times* (Dec. 19, 2013).
13. Schworm, P. Each MIT undergraduate to get \$100 in Bitcoin. *The Boston Globe* (Apr. 30, 2014).
14. Sidel, R. Banks mostly avoid providing Bitcoin services. *The Wall Street Journal* (Dec. 22, 2013).
15. Vance, A. and Stone, B. Bitcoin rush. *Bloomberg Businessweek* (Jan. 13–19, 2014), 46–51.
16. Vigna, P. Jeremy Allaire's Bitcoin startup, Circle, unveils first product. *The Wall Street Journal* (May 16, 2014).

Michael A. Cusumano (cusumano@mit.edu) is a professor at the MIT Sloan School of Management and School of Engineering and author of *Staying Power: Six Enduring Principles for Managing Strategy and Innovation in an Uncertain World* (Oxford University Press, 2010).

Copyright held by author.

Inside Risks

Risks and Myths of Cloud Computing and Cloud Storage

Considering existing and new types of risks inherent in cloud services.

CLOUD COMPUTING AND STORAGE are often seen as general blessings, if not financial salvations. There are good reasons behind this claim. Cloud services are indeed usually much cheaper than their dedicated counterparts. Administration and management oversight are simpler under a single, central authority. Small businesses and startups have taken advantage, using low-cost cloud services during their first few years. Cloud platforms are critical avenues to getting started for many companies, giving them access to many customers at low cost. Many business leaders see the cloud as an engine for small businesses and job creation.

Cloud storage services are also a boon for individual users, most of whom do not back up their computers and mobile devices regularly or at all. Cheap, automatic backup to cloud storage protects their valuable data from loss.

Despite all these bounties, cloud services also present new kinds of risks, which are considered here. Prospective cloud users should evaluate these risks before making their decisions about how to use clouds. The main issue is that expectations of trustworthiness may be unrealistic. Confidentiality, system integrity, data integrity, reliability, robustness, resilience may be questionable. Protection against surveillance, and



denials of service are essential, as are perpetual access and long-term compatibility of stored data. The integrity, accountability, and trustworthiness of potentially untrustworthy third parties and even unknown *n*th parties must also be considered. Those parties may have business models that are radically incompatible with user needs; furthermore, they might go out of business—with users holding the bag. Moreover,

insider misuse may create additional risks. All these risks are relevant to many different types of applications. As one example, from users' perspectives, having unencrypted email maintained by a cloud provider may be particularly risky.

The basic concept of cloud computing and cloud storage has a lineage spanning two generations, with significant experience in designing and administering these systems. Time-

sharing systems with common computing resources and possibilities for collaborative data access have been around since the 1960s (CTSS, Multics, Tymshare), with varying types of sharing. Since the 1980s, Project Athena at MIT has employed the Sun Network File System, and later the Andrew File System, to provide three services that would be identified with today's cloud services: remote storage of application programs, remote storage of personal files, and remote backup of personal files. However, time-sharing and Athena's three services have been under single operational administration, thus minimizing the number of entities that users must trust (while at the same time providing a single point of failure). We know from experience how to offset some of the risks when cloud services are the responsibility of a single administration. With respect to both local and remote servers, some of the risks can be reduced. For example, private systems and intranetworks under local control or more likely the control your own employers (with respect to hardware, software, certificate authorities, and pooled system and network administration) are likely to have greater trustworthiness.

What is new—and the source of new risks—is the scale and distributivity of some of the clouds. They are large distributed systems with few centralized controls. Clouds that provide access to vast amounts of information (such as Google and Amazon) are extremely valuable resources. However, other clouds that store your own data (along with everyone else's) can present serious problems relating to trusting potentially untrustworthy entities.

Giving the “cloud” name to the old concept of large, shared, distributed systems is misleading. It creates a new buzzword, and hides the problems of risks that designers and admins have otherwise grappled with for years. Some of the cloud providers have ignored many of the old risks and are evidently largely oblivious to newer risks as well. Clearly, *cloud computing* is simply *remote computing*, which was one of the primary reasons for the creation of the ARPANET—to allow people in one coastal time zone to benefit from unused resources in other time zones at

Some of the cloud providers have ignored many of the old risks and are evidently largely oblivious to newer risks as well.

certain hours of the day. This has clearly been an even greater benefit in the Internet, with its worldwide coverage. Similarly, *cloud storage* is simply *remote storage*, which in early days became common as off-site backup for obvious reasons of fault tolerance, emergency preparedness, and other reasons.

One risk in identifying remote storage for offsite backup as “cloud storage” is that this term masks the existence of in-house alternatives, such as the common practice of periodically recording file-system snapshots on small detachable media, and keeping them in a safe place. This can be particularly important after nasty penetration attacks that may have compromised a system with the insertion of malware, sniffers, and so on. Furthermore, remote archiving—especially if widely distributed among different repositories—leaves users unsure of whether their information is still retrievable in its original form (unless they have actually retrieved it).

Ron Rivest has been quoted as saying, “Cloud computing sounds so sweet and wonderful and safe ... we should just be aware of the terminology; if we [are] calling it swamp computing. I think you might have the right mind-set.”²

To paraphrase a quote often attributed to Roger Needham, Butler Lampson, or Jim Morris, if you think cloud computing and cloud storage are the answer to your problems, you do not understand those would-be solutions, and you do not understand your problems.

A few examples of relevant recent risky exploits are worth noting here. (Further background on the first nine items and other examples of cloud

compromises can be found in the ACM Risks Forum: <http://www.risks.org/>.)

- ▶ Dropbox's sharing services were hacked, resulting from a security hole in its link-sharing scheme. The exploits were also disseminated by the perpetrators.

- ▶ No-IP had 22 of its most frequently used domains taken down by Microsoft, under a seemingly overreaching federal court order.

- ▶ Amazon Web services have gone down (briefly) several times. Code Spaces (a valued source-code repository built using Amazon's AWS facilities) was effectively destroyed by an attacker demanding ransom.

- ▶ Cisco Systems had a private crypto key embedded in their VoIP manager that allowed unauthorized control of sensitive messaging gear.

- ▶ Cryptolocker and other ransomware programs have forcibly encrypted stored information, and demanded payment to decrypt it (although in some cases have never done so even after receiving the ransom!). Although most of these attacks have been on individual users, the opportunity for attacks on remote storage repositories is clearly a risk. (Recently, an antidote website has reportedly been created.)

- ▶ TrueCrypt (full disk encryption) was discontinued as source-available software by its pseudonymous authors, “as it may contain unfixed security issues.” (Uncertainty remains as to the severity and impact of those possible issues, and how they found their way into the codebase.)

- ▶ Similar things happened to Lavabit, which provided privacy and security features in email services to over 400,000 customers, but was then withdrawn after prolonged legal harassment that attempted to coerce the installation of surveillance equipment.

- ▶ Megaupload.com was taken down by authorities, blocking both illicit and legitimate users.

- ▶ Nirvanix went belly-up financially, giving its users two weeks to exit.

- ▶ Various talks at Black Hat and DEF CON in August were rather disenchanting. In short, essentially every device seems to be compromisable, often with a fixed master password embedded in the system, but with many more subtle vulnerabilities as well. This is old news to Inside Risks readers, but

could be shocking to everyone else.

Among old risks that are still pervasive, even in-house use of local storage can result in hardware outages and database software failures. Redundant copies might actually all wind up in a single vulnerable cloud repository. Furthermore, older data formats may no longer be supported. Local storage still requires attention to backup that can be successfully retrieved—in some cases many years later. Furthermore, if the original information is encrypted, the ability to manage and recover old keys becomes critical.

Recently, increasingly efficient cryptographic schemes are emerging in research communities for proof of data possession and proof of data retrievability. Unfortunately, simple and inexpensive techniques along these lines have not yet found their way from theory to practice. Perhaps more useful are the efforts cloud providers make to ensure their own data storage is recoverable. It may well be that, on average, cloud providers' systems are better administered than the information technology groups of many organizations and agencies. At least, cloud users certainly hope so! Nevertheless, various risks remain.

Another old problem that has been exacerbated involves the ability to delete information ubiquitously. The existence of pervasive copies and different versions has clearly exploded as a result of copies that have been replicated for resilience. Internet mirrors have proliferated far beyond anyone's ability to keep track of unsearchable versions. With storage in some unaccountable remote repository, pervasive deletion will always seem to be questionable. Besides, approaches that may succeed in pervasive deletion may also be victimized by accidental or malicious deletion. In this case, some sort of time machine would be desirable.

Many socially relevant risks also need to be considered, such as different versions of unauthentic data; the presence of misinformation in not quite identical searchable versions of what purports to be the same information; and situations in which people or organizations desire that certain information disappear completely.

Of course, international laws and regulations also present numerous problems—first by their imprecision


or overextension, and second by the uncertainty surrounding the origins and destinations of data and other resource requests. For example, if a nation insists that all information belonging to its citizens must be stored within systems under its own legal jurisdiction, how can that be assured when it is so easy to subvert, and when ownership is itself murky? In addition, we must be cognizant of the risks of ubiquitous surveillance in unaccountable and in some cases unknown remote resources.

As noted in many past Inside Risks columns (this is the 234th in the series), almost every computer or human entity is potentially untrustworthy, with respect to accidents, intentional misuse, and attacks. As an example that remains problematic, the outsourcing of elections with regard to dependence on proprietary systems and software, computing resources, registration databases, networks (whether open or private), and—above all—dependence on potentially untrustworthy people, aptly illustrates the end-to-end nature of the risks from the very beginning of the election cycle to the disputes that result from sources of error, fraud, and confusion—with concomitant fear, uncertainty, and doubt. Insider misuse is serious in all shared resources, but particularly in elections (numerous cases have been noted in the Risks Forum and elsewhere). In this example, outsourcing to unaccountable entities is problematic.

Despite the risks discussed here, there are some hopes for constructive alternatives. Research communities have various approaches to pieces of this puzzle, but rarely to systems as a whole. As a result, many of the previous columns in this series are relevant to the use or misuse of remote resources—even if they focused on problems that were previously considered as local. For example, cryptography that is managed solely by end users for information stored remotely in encrypted forms is often touted as a solution to the problem of having to trust an untrustworthy remote storage provider. Homomorphic cryptography has the potential to allow computations on encrypted information, without the need for that information to be decrypted. These approaches can improve the confidentiality of the information,

and also provide a means for sharing the information through out-of-band shared cryptographic keys. However, they remain vulnerable to other compromises such as accidental or malicious deletion, lapse of contracts with remote providers, loss of cryptographic keys, unavailability of servers, invasive usage monitoring, and so on. As is true in general, key management becomes a fundamental risk in itself. Furthermore, convenient schemes for recovery of lost keys (for example, backdoors) are always vulnerable to misuse—as are any backdoors that can be misused by insiders or external attacks.

Virgil Gligor¹ has considered some of the risks inherent in virtualization in a context very similar to what is examined in this column. Virtualization has certain aspects that are common to the abstractions provided by remote execution and remote access, in the sense that there are well-defined interfaces for dealing with both cases—whether they are virtually remote or physically remote. There are also questions of the trustworthiness of the underlying mechanisms for enforcing the virtualization abstractions—for example, encapsulating, avoiding, or otherwise masking lower-layer vulnerabilities. Gligor's article implicitly addresses some of the topics noted here, and deserves a careful reading for those readers who would like further background than that included here.

I emphasize that clouds can offer real and significant benefits. They also bring many risks, which can be masked by the simplicity of the cloud abstraction. You should weigh these risks when designing, selecting, and configuring your cloud services. 

References

1. Gligor, V. Security limitations of virtualization and how to overcome them. Security Protocols Workshop, SPW 2010, Cambridge, U.K., 2010.
2. McMillan, R. Cloud computing a security nightmare, says Cisco CEO. *Computerworld*; http://www.computerworld.com/s/article/9131998/Cloud_computing_a_security_nightmare_says_Cisco_CEO.

Peter G. Neumann (neumann@csl.sri.com) is Senior Principal Scientist in the Computer Science Lab at SRI International, and moderator of the ACM Risks Forum.

The author is enormously grateful to the members of the ACM Committee on Computers and Public Policy for their continued wisdom and counsel in acting as an advisory group for risks-related activities. This column gained significantly from their feedback.

Copyright held by author.



Kode Vicious Outsourcing Responsibility

What do you do when your debugger fails you?

Dear KV,

I have been assigned to help with a new project and have been looking over the admittedly skimpy documentation the team has placed on the internal wiki. I spent a day or so staring at what seemed to be a long list of open source projects the project team members intend to integrate into the system they have been building, but I could not find where their original work was described. I asked one of the project team members where I might find that documentation and was told there really is not much they need to document, because all the features they need are available in various projects on github.

I really do not get why people do not understand that outsourcing *work* also means outsourcing *responsibility*, and that in a software project, responsibility and accountability are paramount.

Feeling a Sense of Responsibility

Dear Responsible,

While it might seem that the advent of the “fork me on github” style of system design is a new thing, I, unfortunately, have to assure you it is not. Since the invention of the software library, sometime before I was born, and probably before you were as well, the idea that one could build a system by just grabbing the bits one needed has been the way software has been built. We all depend on bits of code we did not write, and often on code we cannot even

We all depend on bits of code we did not write, and often on code we cannot even read, as it arrives in binary form.

read, as it arrives in binary form. Even if we could read it, would we? The code to OpenSSL was open source and readable by anyone who cared or dared, yet the Heartbleed bug sat around for two years undiscovered. The problem is not just about being able to see the code; it has a lot more to do with the complexity inherent in what you might be dragging in to get the job done.

You are correct to quiz the other team members as to why there is not any documentation on how they intend to stitch together the various bits they download. Even if many parts are made up of preexisting software, there must be an architecture to how they are integrated. In the absence of architecture, all is chaos, and systems that are built in that organic mold work for a while, but eventually they rot, and the stench they give off is the stench of impending doom.

A software system is always built from other components, and the questions that you need to ask are: How trustworthy is the component I am using? How stable is the API? Do I understand how to use this component? Let me break those down for you.

Trustworthiness of software is not simply a matter of knowing whether someone wrote it for the purpose of stealing information, though if you are taking factors for your elliptic curve code from a three-letter agency, you might want to think really hard about that. To say that software is trustworthy is to know that it has a track record—hopefully, measured in years—of being well tested and stable in the face of abuse. People find bugs in software all the time, but we all know a trustworthy piece of software when we use it, because it rarely fails in operation.

Stability of APIs is something I have alluded to in other responses, but it bears, or should I say seems to require, frequent repetition. If you were driving a car and the person giving you directions revised them every block, you would think that person had no idea where he or she was going, and you would probably be right. Similarly, a piece of software where the APIs have the stability of Jell-O indicates the people who built those APIs did not really know what they were doing at the start of the project, and probably still do not know now that the software has a user base. I frequently come across systems

that seem to have been written to solve a problem quickly—and in a way that gets Google or Facebook to fork over a lot of cash for whatever dubious service has been created with it. An API need not be written in stone, but it should be stable enough that you can depend on it for more than a point release.

Understanding the use of a component is where the github generation seems to fall on its face most often. Some programmers will do a search based on a problem they are trying to solve; find a Web page or entry in stack overflow that points to a solution to their problem; and then, without doing any due diligence, pull that component into their system, regardless of the component's size, complexity, or original intended purpose. To take a trivial example, I typed “red black tree” into github's search box. It then spat out, “We've found 259 repository results.” That means there are 259 different implementations of a red black tree present. Of course, they span various languages:

Repositories	Language
56	C
43	Java
41	C++
17	JavaScript
13	Python
9	Ruby
8	Go
8	C#
4	Haskell
3	Common Lisp

How are we to evaluate all (any?) of these implementations? We can sort them by user ratings (aka “stars”), as well as forks, which is how many times someone has tried to extend the code. Neither of these measurements is objective in any way. We still do not know about code size, API stability, performance, or the code's intended purpose, and this is for a relatively simple data structure, not for some huge chunk of code such as a Web server.

To know if a piece of code is appropriate for your use, you have to read about how the author used it. If the author produced documentation (and, yes, I will wait until you stop laughing), then that might give an indication of his or her goal, and you can then see if that matches up with yours. All of this

is the due diligence required to navigate the sea of software that is churned out by little typing fingers every day.

Lastly, you are quite right about one thing: you can outsource work, but at the end of the day it is much more difficult to outsource responsibility.

KV

Dear KV,

What do you do when your debugger fails you? You have talked in the past about the tools you use to find bugs without resorting to print statements, such as `printf()` in C, and their cousins in other languages, but there comes a time when tools fail, and I find I must use some form of brute force to find the problem and solve it.

I am working with a program where when we dump the state of the system for an operation that is supposed to have no side effects, the state clearly changes; but, of course, when the debugger is attached to the program, the state remains unchanged. Before we resort to print statements, maybe you could make another suggestion.

Brute Forced

Dear Brute,

Tools, like the people who write them, are not perfect, and I have had to resort to various forms of brute-force debugging, forsaking my debugger for the lowly likes of the humble print statement.

From what you have written, though, it sounds like another form of brute force might be more suitable: binary search. If you have a long-running oper-

Tools, like the people who write them, are not perfect, and I have had to resort to various forms of brute-force debugging.

ation that causes a side effect, the easiest way to find the part of the operation causing you trouble is to break down the operation into parts. Can you trigger the error with only half the output? If so, which half? Once you identify the half that has the bug, divide that section in half again. Continue the halving process until you have narrowed down the location of the problem and, well, not quite voila, but you will definitely have made more progress than you would by cursing your debugger—and it will take less time than adding a ton of print statements if the segment of the system you are debugging is truly large.

Often print statements will mask timing bugs, so if the bug is timing related, adding a print statement may mislead you into thinking the bug is gone. I have seen far too many programmers ship software with debug and print statements enabled, although the messages go into `/dev/null`, simply because “it works with debug turned on.” No, it does not “work with debug turned on”; the debug is masking the bug and you are getting lucky. The user of the software is going to be unlucky when the right moment comes along and, irrespective of the print statements, has a timing error. I hope you are not working on braking systems or avionics, because, well, boom.

If your goal is to find the bug and fix it, then I can recommend divide and conquer as a debugging approach when your finer tools fail you.

KV

Q Related articles on queue.acm.org

Outsourcing: Devising a Game Plan

Adam Kolawa

<http://queue.acm.org/detail.cfm?id=1036501>

Debugging on Live Systems

George Neville-Neil

<http://queue.acm.org/detail.cfm?id=2031677>

Postmortem Debugging in Dynamic Environments

David Pacheco

<http://queue.acm.org/detail.cfm?id=2039361>

George V. Neville-Neil (kv@acm.org) is the proprietor of Neville-Neil Consulting and co-chair of the *ACM Queue* editorial board. He works on networking and operating systems code for fun and profit, teaches courses on various programming-related subjects, and encourages your comments, quips, and code snips pertaining to his *Communications* column.

Copyright held by author.



The Business of Software Vendor: Vidi, Vici

Some hidden costs of outsourcing.

OUTSOURCING IS COMMON in the computing world these days. It is said to save costs, eliminate the need for additional in-house employees, and provide adaptable staffing levels that can flex with the ups and downs of business cycles. But there are costs and dangers to outsourcing and companies can leave themselves open to business practices that can be quite predatory.

Most vendors view the growth and protection of their revenue stream to be the most important thing. Even when they do behave ethically, this can cause conflicts of interest between what is best for the vendor, what is best for the vendor account manager, and what is best for the customer. It can lead to some nasty behavior.

I once worked on a large project alongside a consulting company that actually had meetings to decide just how inefficiently they needed to work to maximize their income. A job done too quickly that resulted in a high-quality system would end the project and the checks would stop. A job done too poorly would get them fired and the checks would stop. Somewhere between these two scenarios was an optimal point where they could drag out the project and maybe land a lucrative maintenance contract fixing the system. Customers were not invited to these meetings.

Who to Blame?

Even more perniciously, I have seen vendors deliberately identify “targets”



in the customer’s environment who could be blamed when something went wrong. The “target” might be a middle manager in the client organization; one low enough to clearly see what was going on and act as a whistleblower, but also high enough to be listened to. The vendor would collect email messages, memos, meeting minutes, and so forth, to be used against the person. Then if the target started making waves, the memos were presented to senior executives in the client organization and the whistleblower portrayed as “...not a team player...” These smear campaigns were often successful and resulted in the targets getting fired. It worked out well for the vendor—they would get rid of a potentially dangerous irritant and divert the blame for problems at the same time. Perfect!

Stubbed Out

“I want you to get everyone writing stubs for the system,” the vendor managing partner told me. On a project some years ago, I was in charge of a large group of programmers and I was puzzled by the directive, “What are the stubs supposed to do?” I asked. The managing partner replied disdainfully, “Stubs are dummy programs that don’t do anything, their functionality hasn’t been decided yet.” “Heck, I know what stubs are,” I responded, “but we need a reason to write them: to set up a system architecture, test infrastructure interactions, to check out program load capability... We don’t write stubs just to write stubs...”

The managing partner’s eyes narrowed. “You’re not a team player, are you?” he said. It was not a question and I had just become a target. He wanted

to report to the client that code was being written, whether it was needed or not. But he was right: I was not a player on that kind of team.

Organizational Issues

These examples operate at the personal level and can lose people their livelihoods. But problems can occur at the organizational level:

► **Project Estimation**—vendors can take over the project estimation processes under the guise of “improving project management” requiring clients without their own internal capability to rely on the vendor’s assessment of what projects will cost.

► **Project Tracking**—vendors may appropriate all project tracking and status reporting so that the customer sees only what the vendor wants them to see.

► **Governance**—unless customers exert a strong hand in managing the portfolio, they rely solely on the vendors for results and the vendors may have a vested interest in it not being *too* successful.

► **Vendor Conflict**—I have seen situations where different vendors were contracted to build different parts of a system and another vendor to test it. It is a recipe for finger-pointing and blame avoidance that might end up with more money going to the vendors.

Revolving Vendor

The goals of the vendor and the customer may not be the same. Vendors are in business to maximize the benefit to *them* and they may do things that are less than optimal for the client. Vendors may recruit inexperienced talent, because they do not have to pay these people as much. But once a programmer has spent a year or two with a client, they are now “experienced” and can command a higher billing rate elsewhere. So the vendor may move out a person who knows what the system does and move in a novice who does not.

If developers come from overseas, their visa may be linked to the vendor and the developers must behave the way the vendor wants them to behave.

Fixed-price contracts that are running over budget give vendors a strong incentive to put lower-cost (and less-talented) resources on the project just when a project needs to keep the more experienced and skilled developers.

The net effect of these vendor strategies is the customer pays a lot more for systems that are of low quality.

Owning and Using

The net effect of these vendor strategies is the customer pays a lot more for systems that are of low quality. But that is just money; there are broader consequences.

In a previous column,¹ I discussed the difference between *owning* and *using* systems. Companies need to *own* the systems that are the executable version of the knowledge of how the company works and why it exists. Companies that outsource the development and maintenance of these systems outsource their future. Now the company does not control that knowledge—they have paid someone else to manage it. Losing control of these systems and the knowledge they contain, they have also lost control of their markets. Vendors can peddle this information by selling their experienced developers’ services to the client’s competitors. They can siphon off what their developers have learned and can build and sell their own solutions and when everyone uses the same packages no one has any competitive advantage. The diluting of a company’s specific knowledge can reduce innovation; the gene pool of variations that catalyze new ideas is thinned out when everyone uses the same systems.

Apprenticeship Has Sailed?

Where do we expect our future talent to come from? I have worked with companies where people say with pride that, while the “grunt” programming is being outsourced, activities further up the food chain are being done by experienced in-house people. But how did they become experienced? How did they acquire the necessary business knowledge and technical exper-

tise? They usually started as a novice programmer and, by working hard, displaying ingenuity, and seizing opportunities, rose through the ranks to become an invaluable resource. If the entry positions are being farmed out to vendors, companies cannot grow the next generation of senior technical staff and become even more vendor-dependent.

Action and Reaction

Outsourcing is still in vogue, but there are some interesting trends occurring.² There is growth in hybrid client/vendor governance, a move to insourcing development, more in-house integration, and contractual stipulations on vendor cooperation. These are all healthy trends.

But there is a deeper issue at stake here operating at the executive level of many modern companies. The headlong charge to outsource essential development is framed as cost reduction but it is driven by something akin to executive cowardice. Building modern systems can be very difficult. Creating and managing an organization that can build these systems is challenging. Looking beyond the next quarter’s financial results requires executive vision. Sacrificing short-term quick hits in favor of long-term growth and corporate health requires technical and leadership backbone. But 25 years ago, leaders of many companies did step up to this challenge and now they do not. Now, to avoid the hard work and challenge of today, executives are signing checks and turning over the keys of the kingdom to vendors.

But vendors only work the way their customers and the market allow them to. Rather than being controlled by them, companies can genuinely partner with their vendors, obtain good systems for a reasonable price, and own and manage the systems knowledge that defines their business. ■

References

1. Armour, P.G. Owning and using. *Commun. ACM* 57, 6 (June 2014), 29–30.
2. Overby, S. 10 IT outsourcing trends to watch in 2014. *CIO Magazine*; <http://bit.ly/1g41ggx>.

Phillip G. Armour (armour@corvusintl.com) is a senior consultant at Corvus International Inc., Deer Park, IL, and a consultant at QSM Inc., McLean, VA.

Copyright held by author.

Viewpoint

Disrupting and Transforming the University

Higher education institutions must modify their business models in response to technology-driven influences.

ARE UNIVERSITIES ABOUT to be disrupted the way Kodak, Borders, and Blockbuster, all recently in bankruptcy, were disrupted? Blended courses, online learning, and MOOCs are moving at light speed compared to the typical university. This Viewpoint will highlight the opportunities and threats from technology and recommend that higher education institutions morph their business models in response to these challenges. Examples of the transformation in higher education include:

► The Minerva Project proposes to create a top-tier for-profit research university. Students live together on different campuses around the world and top professors stream online classes to student seminars (*New York Times*, 4/21/2013).

► Generation Rwanda is starting an entirely massive open online course (MOOC) university with tuition under \$1,500 a year. It plans a 400-person university in Rwanda with MOOCs providing the content and teaching fellows handling discussions. Southern New Hampshire University will test and certify associates degrees. Currently 1% of Rwanda's population has a college degree (*Technology Review*, 3/15/2013).

► Georgia Tech announced a professional online Master of Science degree



in computer science earned through MOOCs in conjunction with Udacity and AT&T. The program estimates tuition at \$6,600 for the three years of course work with 4,500 centers for proctoring examinations. There are recent reports that the school has received twice the expected number of applications for the program (*Wall Street Journal*, 10/29/2013).

While these proposals may or may not be successful, technology is enabling many new ideas for higher education. This Viewpoint describes the promises (and threats) of innovations like these based on my experience teaching along with my research on disruptive technologies.^{2,5}

A Typology

Table 1 summarizes a wide variety of new instructional practices enabled by technology. Online education includes fully online classes, blended learning and MOOCs, and these can be mixed and matched to create a huge variety of student experiences.^a

Opportunities and Threats

Table 2 describes opportunities and threats from disruptive technologies applied to learning. The table distinguishes among content producers/consumers who create and use MOOCs and other online materials, and content consumers who use content created by others.

The development of MOOCs and other kinds of online education expands the scope of the university far beyond students on campus and its alumni; schools can have a global reach. My first MOOC had students from nearly 100 countries. Along with this expansion of scope will come increased competition; MOOCs and online courses spread a school's brand around the world. But that means universities will compete with each other to attract students for a variety of programs and courses, and we will see competition at the level of the individual course and individual course module.

a A blended format generally refers to a course where there are no lectures with class time devoted to discussion or solving problems. This format is associated with the "flipped classroom" from the Khan Academy's successful program to change traditional approaches to teaching.

Technology-enhanced instruction places more responsibility on the student for learning and less on the faculty member.

Generation Rwanda and the Minerva Project preview what could become the greatest threat to universities: new ventures consuming content with cost structures that are a fraction of the costs of existing universities. The content producer develops and distributes content; it has to worry about the high investments needed to develop courses. The university that primarily acquires content benefits from these efforts. However, there must be a balance so that the content producers and consumers remain viable financially. There will be a new funding model as organizations like Coursera license MOOCs to a school and collect a royalty shared with the school that created the MOOC.⁴ San Jose State University is using MOOCs for remedial work (*New York Times*,

4/29/2013). A group of universities is offering MOOCs as a free, for-credit first course toward a university degree to encourage students to start college (see <http://mooc2degree.com>).

The physical campus will see changes. Some students will elect to be on campus for less than an academic year, taking online classes from home and working part-time for part of the semester. Colleges will need dormitories that are more like hotels than apartments. Most faculty will reduce the number of physical class hours when teaching a blended course, reducing the need to build new classrooms. At the same time the university will need new kinds of space for course preparation and delivery and for student interaction.

It is also likely that projects like MOOC U will be the end of the traditional for-profit college.³ A certificate and eventually a degree from MOOC and/or online classes from the top faculty in the country will soon be a better credential than a degree from one of the existing for-profits, and it will certainly cost less.

For students, the technology brings incredible flexibility and many more options, but at a cost: technology-enhanced instruction places more responsibility on the student for learning and less on the faculty member. The Smith School at Maryland online MBA program makes it possible for students to study for a degree from virtually any-

Table 1. Types of instruction and technology-enabled teaching.

Type of Instruction	Technology-Enabled Teaching Examples
Instruction at for-profit universities.	Generally offerings of asynchronous online courses mostly taught by non-Ph.D. faculty with little or no direct faculty-student interaction.
To reach students unable to come to a physical campus, for example, an online MBA program.	1. Synchronous online courses with online interaction between Ph.D. faculty and students plus asynchronous homework; possibly combined with short residence sessions. 2. A program like Georgia Tech's featuring MOOCs and physical proctoring of exams.
To improve the quality of courses that meet physically: Part I.	Blended courses with physical class meetings of shorter duration than a traditional class with video-lectures and multimedia homework; generally expect Ph.D. faculty.
To improve the quality of courses that meet physically: Part II.	Integrate MOOCs in to the traditional class taught by creative and capable Ph.D. faculty.
Instruction for underserved populations and countries.	Free MOOCs with asynchronous videos and interactive sessions via Google Hangouts for a small number of participants and the faculty.
New models for the university: the threat to traditional schools.	1. Project Minerva with streaming video by star faculty and local discussion groups. 2. MOOC degree programs. 3. A MOOC university.

place. It will be possible to configure custom degree programs so that students will not be confined only to completely faculty-defined courses of study.

Do we know that technology-enhanced learning is better than traditional learning? Daphne Koller, co-founder of Coursera, argues that the technology will improve the quality of education, not replace the teacher. “The online experience allows students to watch at their own pace, to achieve mastery in a topic before the class moves on to the next one, to really practice until they get it (*Wall Street Journal*, 5/15/2013).” At this point there are few rigorous, credible studies of synchronous and asynchronous learning or MOOCs at the college level.^b San Jose State found that in a blended circuits course meeting for discussion and using MIT’s online circuits course outside of class time, 91% of students passed compared to 59% in the traditional class (*New York Times*, 4/30/2013). My experience suggests a shorter class focused on discussion as opposed to lecturing creates a more stimulating discussion, but both of these observations are far from controlled experiments and the question remains to be answered.

Faculty can add value to existing courses given the availability of MOOCs and other materials. An economics professor might include parts of MOOCs from two Nobel prize-winning economists in class. The greatest fear is that a few star faculty members from top universities teach video courses while the rest of the faculty becomes discussion leaders or teaching assistants. A faculty member creates a course by determining its content, choosing from available online materials and readings, designing class sessions, preparing assignments, and bringing relevant research into the discussions. The best strategy for faculty members is to be creative in using

^b The U.S. Department of Education website has several meta studies of online education. While there has been a large number of studies, many suffer from weak controls and lack rigor. The Ithaka project conducted a randomized study of the use of an interactive statistics course developed by Carnegie Mellon University and found no major differences on learning outcomes between students in a traditional course and the CMU course.¹

Table 2. Opportunities and threats from technology-enabled learning.

Entity	Opportunity
The University	
Content consumer	Expand brand globally, generate revenue from content, contribute to raising the global level of education, reconfigure campus.
Content producer/consumer	
Established	Supplement courses with new material from top faculty, reconfigure campus.
Start-up	Create a new school without a costly faculty or physical plant.
Students	Increased flexibility; opportunity to spend less time on campus, variety of subjects available, improved learning outcomes, enhance educational opportunities around the world.
Faculty	Opportunity to use new content to add value to courses
Entity	Threat
The University	
Content consumer	High investment cost, overruns, competition from peers developing content and from established and startup universities with lower cost structures; competition at the level of the individual course and the individual course module.
Content producer/consumer	
Established	Competition from startups and schools with different models for education and lower cost structures.
Start-up	Establishing brand and reputation for quality.
Students	Accepting more responsibility for learning, whether learning outcomes will be better, whether nontraditional degrees and certificates will count with employers and others.
Faculty	Becoming TAs for star content producers, less need for Ph.D. faculty.

new and traditional approaches to add value to their courses and to learn how to be mentors and coaches for their students. In a recent interview Koller said: “What we hope and believe is that the role of teachers will change. A teacher will have more time to spend teaching as opposed to spending time in content development and preparation and in grading endless repetitions of the same assignments. Students will come to class to actually have meaningful, engaged discussion with others students and instructors.” (*Wall Street Journal*, 5/15/2013).

Recommendations for a Strategy

Universities in the U.S. and other countries today are able to transform the teaching and learning process through the use of technology. But this technology is fundamentally disruptive to our current model of higher education.⁶ It will indeed be a tragedy if universities deny that a disruptive technology will affect them and end up in the position of Kodak,

Blockbuster, Borders, or the newspaper industry. Universities should meet these challenges by being bold in thought and action rather than by making incremental changes. Some suggestions include:

- ▶ Establish associate provost and associate dean positions for blended and online courses and MOOCs. Converting hundreds of courses to blended requires a strong leader and champion.

- ▶ Move aggressively into the developing markets for different programs. There is likely to be a strong first-mover advantage from being early to the market and establishing a school’s brand.

- ▶ Expedite the development of blended and online courses by providing faculty incentives for transitioning their courses. This effort will be massive and will require substantial resources and funds.

- ▶ Offer degree programs online with a focus on high quality and rigorous courses with faculty-student interaction.

- ▶ Move quickly to convert part-time programs to blended courses, reducing the amount of time students spend in class.

- ▶ Create and market MOOCs with the university's brand and create a favorable balance of trade in course royalties.

- ▶ Develop courses and degree programs based on MOOCs that award credit for students who pay tuition and take proctored exams.

- ▶ Develop support staff and infrastructure to assist faculty in the transition.

- ▶ Hire star teaching faculty who can bring research to their classes and keep them constantly updated. The new star will be someone who connects with and challenges students through interactive sessions and forums.

- ▶ Base hiring on teaching as well as research potential.

- ▶ Review promotion and tenure policies to give more credit for teaching. See if the faculty member brings research into her courses.

- ▶ Enable students to create custom degree programs, mixing and matching courses from their home institution with online courses and MOOCs from other schools.

- ▶ Provide the flexibility for students to spend time on campus as well as at home during undergraduate years.

- ▶ Plan physical facilities that fit technology-enhanced learning. The large lecture is dead; blended courses will be in the 30–50 student size range; students will spend less time in class, freeing classroom capacity.

- ▶ Create dormitories that provide for shorter stays from a few months to a semester.

- ▶ Consider partnering with organizations that offer MOOCs and companies that provide support for developing online programs.

- ▶ Find a replacement for the credit hour; student requirements for a degree and faculty workloads are tied to the credit hour. Yet what does a credit hour mean in a blended course?

Conclusion

Technology is both transformational and disruptive for universities. Project Minerva and Generation Rwanda are examples of how one can create a university that is enabled by technol-

Every university needs to be concerned because it is not clear how different types of schools will be impacted.

ogy. Georgia Tech's master's degree in computer science shows how to use MOOCs to dramatically reduce tuition costs for students enrolled for a degree. Who is threatened by technology-enabled learning? Every university needs to be concerned because it is not clear how different types of schools will be impacted. On balance, it is a time of great opportunity to improve the quality of education, put students in a position to be more responsible for their own learning, and dramatically extend the reach of each institution. The technology will let us leverage the human capital and knowledge on our campuses and use it to raise the level of education around the world. Will universities devote their resources and energy to meeting this challenge? **C**

References

1. Bowen, W., Chingos, M., Lack, K., and Nygren, T. Interactive learning online at public universities: Evidence from randomized trials. *Ithaca S+R*, 2012.
2. Christensen, C., Horn, M., and Johnson, C. *Disrupting Class*. McGraw-Hill, New York, 2011.
3. Cusumano, M. Are the costs of 'free' too high in online education? *Commun. ACM* 56, 4 (Apr. 2013), 26–29.
4. Dellarocas, C. and Van Alstyne, M. Economic and business dimensions money models for MOOCs. *Commun. ACM* 56, 8 (Aug. 2013), 15–28.
5. Lucas, H.C., Jr. *The Search for Survival: Lessons from Disruptive Technologies*. Praeger, Santa Barbara, CA, 2012.
6. Vardi, M. Will MOOCs destroy academia? *Commun. ACM* 55, 11 (Nov. 2012), 5.

Henry Lucas (hlucas@rhsmith.umd.edu) is a professor of information systems at Smith School of Business, College Park, Maryland.

I wish to thank William Dill, former dean of the Stern School of Business at NYU and former president of Babson College, for his insightful comments on an earlier version of this Viewpoint.

Copyright held by author.

Calendar of Events

October 19–21

Conference on Systems, Programming, and Applications: Software for Humanity, Portland, OR, Sponsored: SIGPLAN, Contact: Andrew P. Black, Email: black@cs.pdx.edu

October 19–22,

The Annual Symposium on Computer-Human Interaction in Play

Toronto, CA, Sponsored: SIGCHI, Contact: Lennart Nacke, Email: acagamic@googlemail.com

October 19–22

32nd IEEE International Conference on Computer Design, Seoul, Republic of Korea, Contact: Naehyuck Chang, Email: naehyuck@elpl.snu.ac.kr

October 20–21

Symposium on Architectures for Networking and Communications Systems, Los Angeles, CA, Sponsored: SIGARCH, SIGCOMM, Contact: Viktor Prasanna, Email: prasanna@usc.edu

October 20–22

The 16th International ACM SIGACCESS Conference on Computers and Accessibility, Rochester, NY, Sponsored: SIGACCESS, Contact: Sri Kurnianwan, Email: srikur@soe.ucsc.edu

October 20–24

Conference on Systems, Programming, and Applications: Software for Humanity, Portland, OR, Sponsored: SIGPLAN, Contact: Andrew P. Black, Email: black@cs.pdx.edu

October 27–28

The 13th ACM Workshop on Hot Topics in Networks, Los Angeles, CA, Sponsored: SIGCOMM, Contact: John Heidemann, Email: johnh@isi.edu

Viewpoint

A Turing Tale

Assessing the accuracy of popular descriptions of Alan Turing's influences and legacy.

MUCH HAS BEEN written about Alan Turing during the past decades and by a variety of people, including historians, philosophers, and logicians. Becoming a Turing scholar today not only requires archival research but also the study of several secondary sources. Doing the latter leads to the observation that many texts contain flaws.

In this Viewpoint, I compare and contrast some key arguments put forth by three Turing scholars—Andrew Hodges, Martin Davis, and Jack Copeland—highlighting the conceptual difference between a “universal Turing machine” and a “stored program” computer. My findings complement Thomas Haigh’s January 2014 *Communications* Historical Reflections column, “Actually, Turing Did Not Invent the Computer.”⁷

In his 1936 paper, “On Computable Numbers,” Turing introduced his automatic machines, which do not contain a finite output (nor an input) as is the case with the later-devised “Turing machines.” Turing wanted each of his machines to compute and print a real number (such as π and $\frac{1}{4}$). For example, the machine computing $\frac{1}{4}$ prints the digits 0 and 1 and then forever prints the digit 0 in accordance with $\frac{1}{4}$ ’s binary representation: 0.01000...

During the course of three decades, Turing, Emil Post, Alonzo Church, Stephen Kleene, Martin Davis, Saul Gorn, and others recast the concept of Turing’s 1936 automatic machine. Several years were needed for the term “universal Turing machine” to acquire an



invariant meaning.^{5,12} Martin Davis presented a modern definition in his 1958 book *Computability and Unsolvability*³ and a definition for the layman in his recent book *The Universal Computer: The Road from Leibniz to Turing*⁴—two definitions I abide with here and with which modern textbooks in computer science comply.

The meaning attached to the words “stored program” also changed in the post-war years and it is unlikely those

words meant exactly the same to every historical actor. In this Viewpoint, they refer to a large store inside the computer, containing both numbers and instructions. According to the current state of the art in the history of computing, the words “stored program” were introduced in 1949 by IBM engineers in Poughkeepsie, NY.⁸

Although all three Turing scholars have their own unique narrative thrust, I will discuss Hodges’s 1983 biography

first and then scrutinize Davis's and Copeland's work together. Davis and Copeland have more in common than meets the eye.

Hodges

In his authoritative biography, Hodges put Turing's life in a pluralistic context, as the following points illustrate:

► Also in a world without Turing, his universal machine would have come to light in one form or another and in no small part due to Emil Post, even though Post's "worker" model did not include a "universal machine" construction.⁹

► One hundred years before Turing, Babbage had already planned for storing numbers in a machine that was universal, as he and Ada (Countess of Lovelace) were well aware.⁹

► In America, Eckert and Mauchly perceived the idea of storing instructions inside the machine, in electronic form.⁹

► Von Neumann may or may not have been influenced by Turing when working on the ENIAC-to-EDVAC transition.⁹

Hodges stressed throughout his book that Turing was not taken seriously by most of his contemporaries in the arena of computer building.⁹ Turing's 1936 paper meant a lot to him and to some of his close colleagues, as Charles G. Darwin's repeated statements in 1946 about the ACE machine illustrate.⁹ That said, Turing's paper had little impact on the computer-building community at large.⁹

In what respect, then, did Turing stand out in the 1940s?

Turing had the remarkable ability to unify seemingly disparate theoretical *and* practical concepts. He needed just one tape in his 1936 paper and just one electronic memory in the 1940s. In Hodges's words: "This [unification of data and instructions] was the new idea, ... For it threw all the emphasis on to a new place—the construction of a large, fast, effective, all-purpose electronic 'memory.' And in its way it made everything much more simple, less cluttered in conception."⁹

The idea to unify was, however, not solely Turing's, nor did it require knowledge of Turing's 1936 paper *per se*. But, Turing's unification was, unlike that of most of his contem-

Turing had the remarkable ability to unify seemingly disparate theoretical *and* practical concepts.

poraries, also theoretical in nature. Based on his 1936 universal machine, Turing was able to see that one machine could do the job of several special-purpose machines.⁹ This grander picture of computing was something Turing was not able to convey clearly in the 1940s to his contemporaries who were eagerly, and successfully, building modern computers. That, in brief, is what I take to be Hodges's central technical theme.

Hodges distinguished between Babbage's universal physical machine and Turing's universal physical machine. Babbage had not stored instructions internally in his machine while Turing planned to do just that. As Hodges implicitly conveyed, storing instructions externally (on, say, paper tape) or internally (in computer memory) does not matter in terms of Turing's universality.⁹ Men like Turing and von Neumann very likely understood this theoretical connection during the 1940s, while many contemporaries did not.^a

Davis and Copeland

In contrast to Hodges, both Davis and Copeland depict history as a stream,

as a Turing Tale, starting with Turing's abstract 1936 paper and ending with the modern computer.² They completely neglect that grasping the emerging practical implications of Turing's 1936 paper took several years, even for logicians.

To set the stage for Turing, Davis refers to a tiny excerpt from Babbage's writings—which states that his analytical engine "could do everything but compose country dances"—to conclude that Babbage had a limited view on universality.⁴

Davis also puts Turing on a pedestal by ridiculing the following 1955 statement of the computer pioneer Howard Aiken: "If it should turn out that the basic logics of a machine designed for the numerical solution of differential equations coincide with the logics of a machine intended to make bills for a department store, I would regard this as the most amazing coincidence that I have ever encountered."⁴

Davis rightfully observes that Aiken did not grasp Turing's notion of universality. But Davis takes one step too many when he portrays Aiken as someone who was therefore lagging behind on the current events of his time. In Davis's words: "If Aiken had grasped the significance of Alan Turing's paper, published two decades earlier, he would never have made such a preposterous statement."⁴

A more careful interpretation of history, by contrast, characterizes Aiken as one of several computer professionals who simply did not have to rely on Turing's "universal machine" concept to advance computer technology—an observation that is apparently difficult to make by an eminent logician who has experienced the past differently. Davis was in fact still writing down his newly acquired insights between Turing's theory and computing practice in his book *Computability and Unsolvability*^b when Aiken made his "preposterous" statement in 1955. Davis's 1958 book opened the eyes of many mathematically inclined computer programmers,¹¹ a large percentage of which had not heard of Turing until then.

Davis also notes that "Turing's universal computer was a marvel-

a I am giving Turing and von Neumann the benefit of the doubt here. This is similar to Hodges's conjecture that Turing also knew all along that "program modification does not extend the range of possible operations" either.⁹ Hodges noted Herman Goldstine apparently did not see this connection, thereby suggesting Turing's 1936 paper was received in different (and less thorough) ways among those who did come across his work. (Goldstine had eagerly read Turing's 1936 paper by January 8, 1946.⁶ My thanks to Thomas Haigh and Mark Priestley for bringing Herman Goldstine's letter to my attention.) I opine that many people today mistakenly take "program modification" to be part and parcel of Turing's 1936 universal-machine construction.

b As Davis confirmed to me (Ghent, November 8, 2011).

ous conceptual device,” which is of course true, and then continues as follows: “But could one actually build such a thing?...These questions were in Turing’s mind from the very first.”⁴ Hodges, by contrast, has cautioned his readership not to blindly believe that Turing set about constructing a universal machine before the war.⁹ Although Turing was a very creative and rather unusual mathematician, he, too, needed time to connect his 1936 work on logic to rapidly changing technology.

Copeland goes even further than Davis by conflating the “universal machine” and the “stored program” concepts. (Copeland is not alone. See, for example, Priestley’s scrutiny of Ceruzzi’s work.¹³) Copeland misleadingly describes “the stored-program universal computer” as a “single invention” from 1936²—a statement that both Hodges and Davis have complained about.^{4,10}

“Computer science textbooks,” Copeland says, “often attribute Turing’s stored-program concept to von Neumann.” But, Copeland insists, von Neumann “never claimed it as his own.”² Copeland then continues: “On the contrary, von Neumann said that Turing’s ‘great positive contribution’ was to show that ‘one, definite mechanism can be *universal*.’”^{c,2}

Von Neumann referred here to Turing’s 1936 universal machine, which is not the same as the “stored program” computer of the 1940s. A “stored program” is only a means to constructing a practical realization of a “universal Turing machine.” Turing and von Neumann were able to make this observation, exactly because they were well versed in both theory and practice. Von Neumann’s letter^c does not support Copeland’s reasoning.

Finally, lack of primary sources forces both Davis and Copeland to repeatedly refer to praising, second-hand, comments. Davis relies on *Time* magazine to make a case for Turing.⁴ Copeland, in turn, writes: “Many people have acclaimed von Neumann as the ‘father of the computer,’ von Neumann’s friend Stanley Frankel observed, “but I am sure that he would never have made that mistake himself.”²

c Von Neumann’s letter to Wiener, 1946.

When, how, and why did Turing’s work influence other actors during the course of history?

Here we have one of several references^d in Copeland’s book to contemporaries of von Neumann. Such recollections should be handled with care because they come from people who participated in a success story that was already several decades old and were in no better position than most of us today to identify what only very few men like Turing and von Neumann knew.

Conclusion

All three Turing scholars—Hodges, Davis, and Copeland—depict the 1940s as a heterogeneous decade. The first half was one of secrecy: Turing was involved in practical issues concerned with electronic computation (albeit for a special kind) and was aware of the completion and significance of the Colossus computers. During the second half of the 1940s, Turing and a small number of by now highly expert colleagues were—despite being unable to tell anyone of the sources of their expertise—actively and publicly involved in computer projects in the U.K.

In retrospect, then, the reader can argue that I have put too much weight on the assessment of Turing’s contemporaries. I think most readers will agree with me when I say Turing was a genius. What is relevant here is the following type of question: Did this genius live on a remote island, or did everybody depend on him to advance computer technology? If the former were true, then scholars would hardly be interested in Turing to begin with. Davis and Copeland want us to believe the latter extreme, while Hodges places Turing’s life in

d Frankel’s letter to Randell, 1972.

between both opposites. Historians who want to improve upon Hodges’s historiography should ask: When, how, and why did Turing’s work influence other actors during the course of history? The assessment of Turing’s contemporaries and successors thus matters a great deal.

I conclude that Hodges’s 1983 biography is far more accurate than anything that has been written since. From the 1970s onward, popular claims have been made, describing Turing as the “inventor of the computer”; but, see Burks’s fitting rebuttal¹ and van Rijsbergen’s and Vardi’s sober reflections on Turing’s legacy^{14,15}—a legacy that lies more in programming than in computer building.⁵ **□**

References

- Burks, A.W. The invention of the universal electronic computer—How the electronic computer revolution began. *Future Generation Computer Systems* 18 (2002), 871–892.
- Copeland, B.J. *Turing: Pioneer of the Information Age*. Oxford University Press, 2012.
- Davis, M. *Computability and Unsolvability*. McGraw-Hill, 1958.
- Davis, M. *The Universal Computer: The Road from Leibniz to Turing*. CRC Press, 2nd edition, 2012.
- Daylight, E.G. Towards a historical notion of “Turing—The father of computer science.” *History and Philosophy of Logic*. Accepted for publication; see: <http://www.dijkstrascry.com/TuringPaper>.
- Goldstine, H.H. untitled (A letter from Goldstine to Womersley, January 8, 1946). Herman H. Goldstine papers in the collection of the American Philosophical Society in Philadelphia, Series 1, Box 3.
- Haigh, T. Actually, Turing did not invent the computer. *Commun. ACM* 57, 1 (Jan. 2014), 36–41.
- Haigh, T., Priestley, M., and Rope, C. Reconsidering the stored program concept. *IEEE Annals of the History of Computing* (Jan.–Mar. 2014), 4–17.
- Hodges, A. *Alan Turing: The Enigma*. Burnett Books, 1983.
- Hodges, A. Book review: The essential Turing. *Notices of the AMS* 53, 10 (Nov. 2006), 1190–1199.
- Knuth, D.E. and Daylight, E.G. *The Essential Knuth, volume 2013*. Lonely Scholar, 2013.
- Petzold, C. *The Annotated Turing: A Guided Tour through Alan Turing’s Historic Paper on Computability and the Turing Machine*. Wiley Publishing, Inc., 2008.
- Priestley, M. *A Science of Operations: Machines, Logic and the Invention of Programming*. Springer, 2011.
- van Rijsbergen, C.J. Turing and the origins of digital computers. In *Aslib Proceedings* 37, pp. 281–285. Emerald Backfiles, June/July 1985. Paper presented at an Aslib Evening Meeting, Aslib, Information House, March 27, 1985.
- Vardi, M.Y. Who begat computing? *Commun. ACM* 56, 1 (Jan. 2013), 5.

Edgar G. Daylight (egdaylight@dijkstrascry.com)—also known as Karel Van Oudheusden—has a Ph.D. from KULeuven in Belgium and is a researcher in software engineering and the history of computing at Utrecht University in the Netherlands.

The author encourages readers to view his blog post at <http://www.dijkstrascry.com/ATuringTale> for more detailed information pertaining to the references cited in this Viewpoint.

Are you looking for your next IT job? Do you need Career Advice?

The **ACM Career & Job Center** offers ACM members a host of career-enhancing benefits:

- A **highly targeted focus** on job opportunities in the computing industry
- **Access to hundreds** of industry job postings
- Resume posting **keeping you connected** to the employment market while letting you maintain full control over your confidential information
- **Job Alert system** that notifies you of new opportunities matching your criteria
- **Career coaching** and guidance available from trained experts dedicated to your success
- **Free access** to a content library of the best career articles compiled from hundreds of sources, and much more!



Visit **ACM's Career & Job Center** at:
<http://jobs.acm.org>



Association for
Computing Machinery

Advancing Computing as a Science & Profession

The **ACM Career & Job Center** is the perfect place to begin searching for your next employment opportunity!

Visit today at <http://jobs.acm.org>

Article development led by [acmqueue](http://acmqueue.queue.acm.org)
queue.acm.org

Public, verifiable, append-only logs.

BY BEN LAURIE

Certificate Transparency

ON AUGUST 28, 2011 a mis-issued wildcard HTTPS certificate for google.com was used to conduct a man-in-the-middle attack against multiple users in Iran. The certificate had been issued by a Dutch certificate authority (CA) known as DigiNotar, a subsidiary of VASCO Data Security International. Later analysis showed that DigiNotar had been aware of the breach in its systems for more than a month—since at least July 19. It also showed that at least 531 fraudulent certificates had been issued. The final count may never be known, since DigiNotar did not have records of all the mis-issued certificates. On Sept. 20, 2011, DigiNotar was declared bankrupt.

The damage caused by this breach was not confined to Iran. When the DigiNotar roots were eventually revoked, two weeks after the initial discovery, they included one used by the Dutch government to provide Internet services. This revocation prevented the Dutch from buying and selling cars, electronically clearing customs, and purchasing electricity on the international market, among many other things. Also, of course, every Web server with a certificate issued by

DigiNotar had to scramble to get a new certificate.

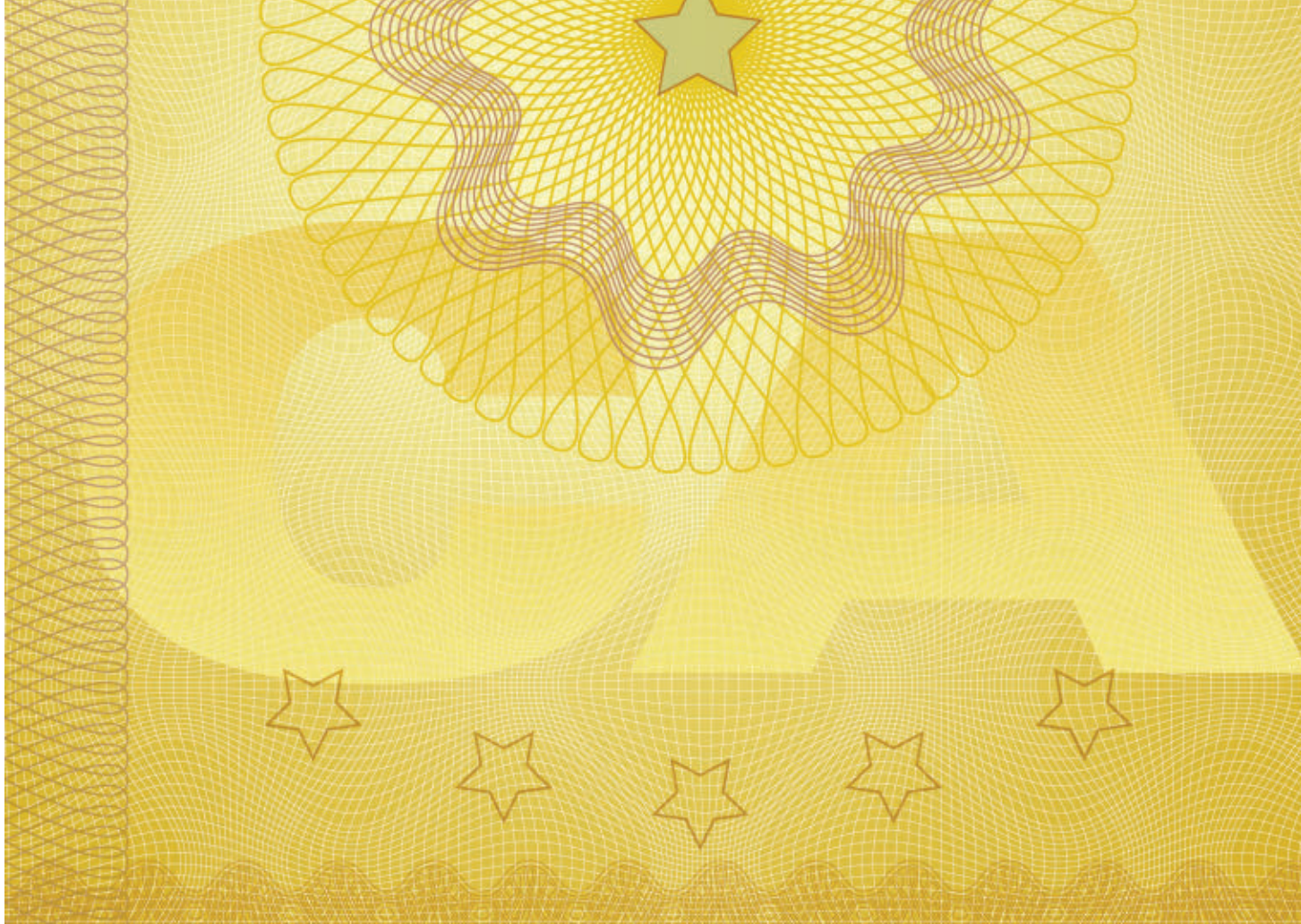
This was not the first time a CA had been compromised, nor was it the last. The Comodo Group (fraudulent certificates issued for Google, Yahoo!, Skype and others),¹ TürkTrust (unauthorized google.com certificate),² and ANSSI (certificates issued via an intermediate allegedly for local network monitoring) have all had reported breaches or internal mis-issuance. More such incidents are sure to come.

It is easy to blame the CAs' poor security for these breaches, but the fact is the state of the art in software engineering and system design cannot render anyone absolutely safe from attack. While the CAs surely must take some of the blame (particularly those that know what has happened but keep it quiet in the hope they will get away with it!), no one yet knows how to build a completely foolproof system. So, how can we make things better?

Alternatives to the CA

Perhaps it is instructive to take a step back and consider the problem we are really trying to solve: ultimately, we want to ensure Web users are actually talking to whom they think they are talking to, and that no one else can intercept the conversation. That is really an impossible goal—how can a computer know what the user is thinking—but for now let's reduce the problem to a slightly different one: how to ensure the Web user is talking to the owner of the domain corresponding to the URL being used. Granted, this is a rather weak strategy, but it does work in at least some circumstances. Most people know the right URL for the big websites such as Google, PayPal, and eBay. Likewise, if users follow a correct link from an honest source (which is, in practice, most links), they are protected.

The solution the computer world has relied on for many years is to introduce into the system trusted third parties (CAs) that vouch for the binding between the domain name and



the private key. The problem is that we have managed to bless several hundred of these supposedly trusted parties, any of which can vouch for any domain name. Every now and then, one of them gets it wrong, sometimes spectacularly.

What are the alternatives? And what are the constraints on them? Let's talk about constraints first.

Constraints. The constraints will definitely vary depending on the nature of the system. Here, the focus is on the Web from the point of view of Google Chrome. The following are some constraints to consider in devising a secure system for Chrome:

- ▶ *Migration path.* There must be some plausible way to get there from here. Solutions that require the whole world to change on a flag day, for example, are just not going to fly.

- ▶ *Generally applicable.* No one is special. Everyone must be able to participate. In other words, the solution must scale.

- ▶ *(Almost) no added latency.* Chrome is very passionate about page load times, so they cannot be made noticeably slower. A corollary is no synchro-

nous out-of-band communications: everything needed before a page loads must arrive in the same channel as the page itself. Experience shows that if we have to consult another source, it will fail and it will be slow. (For a discussion of that experience, see <https://www.imperialviolet.org/2014/04/19/revchecking.html>.)

- ▶ *Do not make the situation worse.* For example, "fixing" one set of trusted third parties by introducing another does not seem like a step forward.

- ▶ *Do not push the decision onto the end user.* We already know users do not understand certificate warnings. Expecting them to make sense of even more complex schemes is unlikely to be the answer.

Given the shortcomings of CAs and the system of trusted third parties to ensure Internet security, several alternatives have surfaced. At Google we have found only one (spoiler alert)—Certificate Transparency—that can overcome all the constraints and present a reasonable solution to the security problem. Before considering why that is the case, let's look at some of the other alternatives.

Pinning. One alternative is for websites to advertise which certificates (or CAs) are correct for them, with the browser rejecting any certificate that is not on that list. Right now, pins for some sites are built into Chrome, but proposals exist to allow anyone to advertise a pin (for example, <https://datatracker.ietf.org/doc/draft-ietf-websec-key-pinning/> and <http://datatracker.ietf.org/doc/draft-perrin-tls-tack/>).

Pinning fails for subtle reasons. What happens when something goes wrong? Clearly, a pin cannot simply be replaced by a new pin, or the whole point is defeated. So, pins expire after some preset time. If you lose your key before the pin expires, then your site does not work until it does expire. Short expiration times provide little protection from interlopers, but long pin times could mean some serious downtime in the event of disaster.

Right now, if this happens, the recourse is to contact Chrome and ask to change your pin, but this is not a scalable, inclusive solution. Furthermore, at least while pins are built into Chrome,


it introduces a new trusted third party (such as Google).

Notaries. Another popular alternative is to use notaries. The best-known notaries are the Perspectives project (<http://perspectives-project.org/>) and Convergence (<http://convergence.io/>). Google also ran one for a while, called the SSL Certificate Catalog,⁴ but decided not to continue to support it after starting work on Certificate Transparency. The idea is to scan the Internet periodically, ideally from multiple points of view, inserting all certificates into a notary log, which can later be asked, “Have you seen this certificate before?” If the answer is “no” or “not very often,” then the certificate might be viewed with some suspicion.


This idea has quite a few problems. The biggest one is that an answer of “no” could simply indicate the site has just changed its certificate. Should sites be inaccessible every time they renew their certificates? That seems unwieldy. Second, the notary approach involves an out-of-band check, which breaks one of the deployability rules. Third, a bold attacker might deploy a fake certificate widely, which would make the notaries think everything is fine. Finally, it introduces a new trusted third party.

DNSSEC. Another alternative is to base trust in Domain Name System Security Extensions (DNSSEC). Two mechanisms exist for this purpose: DNS-based Authentication of Named Entities (DANE; <https://tools.ietf.org/html/rfc6698>) and Certification Authority Authorization; (CAA; <https://tools.ietf.org/html/rfc6844>). Strictly speaking, CAA makes DNSSEC optional but strongly recommended. Both DANE and CAA associate particular certificates or CAs with hosts in the obvious way by having appropriate DNS records for the host’s name. The difference is the way the records are used: DANE records are to be checked by clients when connecting to servers; CAA records are to be checked by CAs when issuing certificates. If the CA is not included in the CAA record for the new certificate, then it should refuse to issue.

Both proposals have problems that are inherent in DNSSEC, but CAA also has the further issue of not really solving part of the problem—namely, mis-



Given the shortcomings of CAs and the system of trusted third parties to ensure Internet security, several alternatives have surfaced. At Google we have found one that works (spoiler alert)—Certificate Transparency.



issues by subverted or malicious CAs. Clearly they will simply not bother to consult the CAA record.

More importantly, however, DNSSEC, like the existing public-key infrastructure (PKI), introduces trusted third parties into the equation, which may or may not fulfill their duties. The trusted third parties in this case are DNS registries and registrars. Sadly, their security record is substantially less impressive than that of CAs.

Some argue that DNSSEC has an inherent protection against subversion of these trusted third parties, because the public nature of DNS makes any meddling immediately visible. The problem with this theory is that DNS is a distributed system—my view is not your view, and there is nothing ensuring our two views are consistent. It is thus relatively easy for attackers to present a split world showing one set of DNS records to their victims and another set to those who seek to check the integrity of DNS.

Another problem is simply that DNSSEC so far is not widely deployed, so we would be gating one improvement on another for which we have already waited well over a decade (indeed, I was busy fixing the “last” problem in DNSSEC more than eight years ago; see RFC 5155 <http://tools.ietf.org/html/rfc5155>).

Finally, experiments indicate that at least 4% of clients cannot get DNSSEC records at all because of routers that take over the DNS and do not support DNSSEC, because of captive portals and the like, and because of blocking TCP on port 53 (DNS is usually served over UDP, but larger records require fallback to TCP), and other causes.

Note, however, that DANE would be useful in the context of SMTP. SMTP servers are already fully at the mercy of DNS and currently use only opportunistic Transport Layer Security (TLS). DANE would definitely be an improvement.

Bitcoin-based solutions. I have written extensively on what is wrong with Bitcoin (for example, it is the least green invention ever, and all of history could be destroyed by a sufficiently powerful adversary if it were truly decentralized, which it is not). Nevertheless, people continue to worship irrationally at the altar of Bitcoin, and this worship extends to DNS and keys—for

example, DNSChain (<https://github.com/okTurtles/dnschain>).

Apart from being an extremely costly solution (in terms of wasted energy, in perpetuity), it also introduces new trusted third parties (those who establish the “consensus” in the block chain) and has no mechanism for verification.

Certificate Transparency

The drawbacks of all these alternatives have led us to pursue a different approach, called Certificate Transparency. The core idea behind Certificate Transparency is the public, verifiable, append-only log. Creating a log of all certificates issued that does not need to be trusted because it is cryptographically verifiable (and it turns out this is possible, as explained in more detail later) allows clients to check that certificates are in the log, and servers can monitor the log for mis-issued certificates. If clients decline to connect to sites whose certificates have not been logged, then this is a complete solution. It becomes impossible to mis-issue a certificate without detection.

This mechanism allows us to meet all the constraints listed earlier. There is a migration path: certificates continue to be issued and revoked as they always have been, and over time more and more clients check for log inclusion, and more and more servers monitor the logs. Even before all clients check, the fact that some do confers a herd immunity on the remainder.

Everyone can participate. It is not difficult to get a certificate into a log, and since the log itself makes no judgment on the correctness of the certificate, there is no change to the revocation of bad certificates, which is still done by the CAs.

Latency is not added because log-inclusion proofs are compact and included in the TLS handshake.

No trusted third party is introduced. Although the log is indeed a third party, it is not trusted; anyone can check its correct operation and, if it misbehaves, prove that it did.

Finally, Certificate Transparency does not push the decision onto the user. The certificate is either logged or it is not. If it is logged, then the corresponding server operator (or other

interested parties) can see it and take appropriate action if it is not valid. If it is not logged, then the browser simply declines to make the connection.

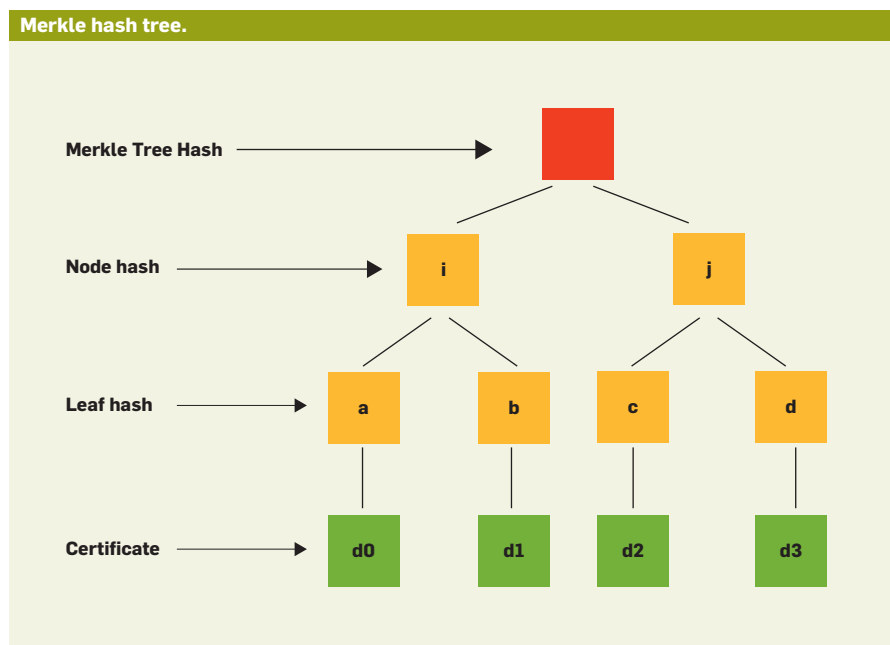
How it works. How do you build a verifiable, append-only log? This turns out to be relatively straightforward in essence, although there are some design trade-offs and some interesting problems to overcome. One obvious approach would be for clients of the log simply to verify, by periodically downloading the entire log, that it satisfies the append-only property. A client could also compare its copy of the log with other clients’ copies to verify they are all seeing the same log. Then everyone would know the log is indeed public and append-only.


This is very wasteful of bandwidth and storage, however. A better approach is to use Merkle trees, as shown in the accompanying figure. A Merkle tree’s leaves are the items in the log. Each branch in the tree is a cryptographic hash of the nodes below the branch (this follows the somewhat odd convention that trees grow downward; the root is at the top, the leaves are at the bottom). Clearly, from the properties of cryptographic hashes, each node is a summary of all the nodes below it: none can be modified without changing the hash. Thus, the root of the tree is a summary of all of the leaves. This means it is now possible for clients to efficiently verify they have seen the same tree by simply comparing a hash.

That is not all we want from a log, however. We would like to verify things that are claimed to be in the log are actually in the log and the log has the append-only property (or to put it another way, the entirety of yesterday’s log is included in today’s log). Fortunately, a Merkle tree is an efficient way of accomplishing this. To show that a particular leaf is in a log, all that is needed is the hash of that leaf and a list of neighboring hashes working up the tree. The root can be recalculated from this list, and, because the hash is cryptographic, this means the leaf must be present in the tree (otherwise, it would not be possible to produce a list of hashes that combine with the leaf hash to produce the root hash). Similarly, yesterday’s root can be linked to today’s root by a list of hashes representing the new items added to the tree.

One more ingredient is needed for a complete system. If a log misbehaves, then we need to be able to prove it has done so. Luckily, this is again rather easy. The log simply needs to sign things. In principle, in fact, the log could sign just one thing: the root of the tree. This is also a signature for everything included in the tree.

This leads to the first trade-off. Logs should be both highly available and consistent. In order for TLS clients to ensure certificates are actually in the log, each certificate must include some kind of proof of inclusion. (Techni-





TTLS

cally, the proof can come anywhere in the conversation with the server, but since rolling out new server software will take many years, the only option that can be universally deployed right now is to include the proof directly in the certificate.)

Our initial thought was to include a Merkle proof—that is, a signed tree head and the hashes leading from the entry for the certificate to the signed head. This runs sharply into the available/consistent trade-off, however: in order to make the log highly available, you need to run multiple instances that are geographically distributed. That means the process of synchronizing the instances can potentially take some time, and they must be synchronized to achieve the append-only property. Therefore, before a CA can issue a certificate, it must first submit it to the log and wait until the log has synchronized all its instances (in order to assign the new certificate a position in the log) and returned a Merkle proof for the new version of the log. Although this can be quite fast most of the time,

in the face of network and server outages it could take quite some time—hours or possibly even days.

CAs found this delay in their issuance pipelines to be unacceptable. In fact, there are points on the trade-off spectrum that are even slower—for example, the log could return both a proof of inclusion and proof that log monitors had not only seen the new certificate, but also had time to take action if it had been incorrectly issued.

Luckily, there are also points on the spectrum that are faster. The version that was finally implemented has the log server return a signed timestamp for the new certificate, known as a signed certificate timestamp (SCT). This signed timestamp is a promise for future inclusion in the log. Each log has a maximum merge delay (MMD) parameter, which says how long after issuing the signed timestamp it is permitted to wait before inclusion.

On the face of it, this would seem to allow the log to respond instantly, since all it needs to do is generate a signature, a process that takes under

a millisecond on modern hardware. But this is not quite true. Since the SCT is a promise to include the certificate, the log absolutely cannot afford to lose the certificate once it has issued an SCT. Therefore, the log must have some redundancy in the storage of incoming certificates; the certificate should probably be stored in multiple data centers. This does allow the consistency requirement to be relaxed. A log can store incoming certificates in some subset of its redundant instances and later resolve any inconsistencies before deciding on an order for new certificates and generating a new version of the log. This kind of redundancy is substantially faster than consistent redundancy.

The next trade-off is the MMD. Obviously, those monitoring logs would like the MMD to be as short as possible, since a log colluding in deliberate misissuance will presumably delay logging for as long as possible. Note that if the log does not collude or lose control of its key, the CA (or the CA's attacker) cannot influence the time taken by the

log to integrate new certificates. Log operators, on the other hand, need the MMD to be long enough to allow logs to establish a consistent state, perhaps even in the face of software errors. We are still deciding what an acceptable MMD is, but clearly it has to be at least hours and possibly as long as a couple of days.


The third trade-off is the number of logs that each certificate should be logged in. There are two reasons for favoring the use of more logs. First, if a log does go bad, clients will stop trusting it. If all the SCTs for a certificate are from bad logs, then the certificate will no longer work. Second, the more SCTs are required in a certificate, the harder it is for an attacker to avoid detection, since the attacker would have to control both the CA (or the CA's key) and all the required logs (or their keys).

Our current thinking is that each certificate should be logged in at least two logs and an increasing number as the certificate lifetime goes up—five logs for certificates with lifetimes of more than 39 months. The reasons for reducing the number of logs are straightforward: increased size of the TLS handshake; increased time taken to create certificates (though note that where the number of logs exceeds the required number of SCTs, firing off requests to all logs in parallel and taking the first n to respond should generally be fast); and the increased size and bandwidth requirements of individual logs caused by redundant logging.

Finally, a fourth trade-off: what should be admitted into a log? An attractive response is “anything,” but logs are useful only if their size is manageable. Someone has to watch them, and if they become so large that no one can feasibly do this, then the logs may as well not exist. The straightforward answer is to admit only those certificates that can be chained to a CA that the clients recognize. There is arguably no point in logging certificates that do not satisfy this criterion, since browsers will not accept them anyway. (It would be nice if the usefulness of self-signed certificates could somehow be bolstered with this kind of log, but the spam problem, coupled with the lack of any way to effectively revoke self-signed certificates, is evidence that no



Logs are useful only if someone is watching them, so it is important to know the participants in this ecosystem and exactly what they do.



one has found a way to do so; however, see the discussion on other types of logs toward the end of this article.) Besides, even if the browsers did accept them, there is currently no scalable way to revoke them.

The ecosystem. Logs are useful only if someone is watching them, so it is important to know the participants in this ecosystem and exactly what they do.

First, certificates must somehow get into the logs. Although CAs will likely do this initially, the standard for Certificate Transparency also allows SCTs to be presented in a TLS extension. This option requires modified server software but is already experimentally supported by the Apache HTTPD server. Given software that supports the TLS extension, site operators can do their own logging, and, because they can update the SCTs whenever they feel like it—which is not possible for SCTs baked into certificates—they can also use fewer of them and run less risk that the certificate might not be accepted.

Second, clients must check the certificates they see really are in the log. Because of the requirement that no out-of-band communications are allowed, this means trusting the log when the certificate is presented but later verifying its honesty by demanding a proof of inclusion.

Third, interested parties (site operators, CAs, and researchers, for example) need to monitor logs to ensure CAs are not doing anything improper—issuing certificates for sites they should not be, for example, or issuing certificates that have extensions or flags set that should not be, or issuing certificates that do not conform to standards. Monitoring also ensures logs are not violating the append-only property or breaking their MMDs, or other such misbehavior.

Finally, everyone who interacts with logs should check with each other that they have all seen the same log—that is, that the log is not presenting different views to different people. This would enable a log to persuade clients it had logged certificates in one view, while showing a view without those certificates to the websites they were purportedly for.

Gossip. All of these tasks are straightforward, except the last. Moni-

toring logs, obtaining consistency and inclusion proofs and so forth can be done by directly querying the log, but checking for consistent views is more difficult. To do this, the various log clients will *gossip*. In the long run, this could occur over a variety of protocols—XMPP, SMTP, peer-to-peer connections, among others—but our first suggestion is to piggyback gossip on TLS. Whenever a client connects to a server, it sends a few items to the server, which the server may verify or merely cache; in return the server sends a few items back from its cache. This establishes what is effectively a peer-to-peer network between the clients.

Because this exchange is piggybacked on a connection made for some other purpose (presumably fetching a Web page), only a few items should be sent in order to conserve bandwidth and avoid excessive latency. When the connections are specifically established for gossip (for example, directly to a log server or using some peer-to-peer protocol with other clients), there is no reason to worry about that, and clients and servers can choose to send large numbers of items—perhaps everything they know.

What items do they send? That is a matter for debate (and simulation), but we can be reasonably sure what the minimum requirement is: an STH (signed tree head). Every client should be able to reassure itself that it has seen the same logs as every other client. Logs are summarized by STHs, so gossiping them is clearly the least a client would wish to do.

STHs have some nice properties for gossiping. First, they are signed, so a bad actor cannot inject spam into the protocol. Every participant can trivially reject messages that did not originate at a log. Second, given two STHs from the same log, it is possible to prove consistency between them, and thus discard the older one. This means caches are $O(\text{number of logs})$ in size.

Would more gossip be desirable? Possibly it would be useful to gossip STH consistency proofs for recent STHs, thus reducing load on logs. Servers might also want to gossip their own SCT inclusion proofs along with the corresponding STH.

Exactly what is gossiped, and

when, is an open question at the time of writing. It is being explored through simulation.

Other uses of transparency. Certificate Transparency originally motivated the work we have done on verifiable logs, but there are other useful applications:

► *Binary transparency.* This allows you to make logs of applications that are downloadable on the Internet. Like Certificate Transparency, Binary Transparency does not prevent the binaries from being malicious, but it does reassure users that the binaries they are getting are visible to the world for analysis, making the deployment of targeted malware much harder.

► *DNSSEC transparency.* DNSSEC is an attractive alternative to the CA-based world of authentication, but it has its own list of potential weak points—in particular, domain registries and registrars. Transparency for keys held in DNSSEC would ensure these can be monitored for correct operation.

► *Revocation transparency.* Once a bad certificate is identified, it should be revoked. Existing mechanisms allow dishonesty in that process, too—for example, selectively setting the revocation status to unrevoked for malicious purposes.

► *ID to key mappings.* This could consist of, for example, email to PGP (Pretty Good Privacy), instant messaging ID to OTR (off-the-record; see <https://otr.cypherpunks.ca/>).

► *Trusted timestamps.* Protocols exist for digital notaries, but they currently require trust in the notary. A notary that logged all timestamps would not need to be trusted.

Other constructions. When thinking about Revocation Transparency, my colleague Emilia Käsper and I invented a new construct: a sparse Merkle tree (<http://www.links.org/files/RevocationTransparency.pdf>). The idea is if you want to have a verifiable mapping, you can store the elements in the map as the leaves of a very large Merkle tree—say, one with 2^{256} leaves (that is, 256 levels deep). Although such a tree would normally be impossible to calculate, the observation is that most of the leaves are empty, which means most of the next level up would have the same value—the hash of two empty leaves; likewise for the level above that, and so on. This means it is, in fact, pos-

sible to calculate the root of the tree and make proofs of inclusion and so forth, so long as the tree is sparse. This structure can be used as an adjunct to a verifiable log to provide an efficient, verifiable map.

Status. Certificate Transparency is under active development at Google. We have two logs running in production, with a third planned by year's end. Others (for example, ISOC, Akamai, and various CAs) are also planning to run public logs. We have open-source implementations of all the key components. Chrome supports Certificate Transparency and will make it mandatory for Extended Validation (EV) certificates in January 2015. More than 94% of CAs (by volume of certificates issued) have agreed to include SCTs in their EV certificates.

Once we have seen the system working well for EV certificates, we plan to roll out Certificate Transparency for *all* certificates. We also intend to pursue some of the other uses for verifiable logs and maps. G

Related articles on queue.acm.org

Network Forensics

Ben Laurie

<http://queue.acm.org/detail.cfm?id=1016982>

The Case Against Data Lock-in

Brian W Fitzpatrick and JJ Lueck

<http://queue.acm.org/detail.cfm?id=1868432>

A Decade of OS Access-control Extensibility

Robert N.M. Watson

<http://queue.acm.org/detail.cfm?id=2430732>

References

- Comodo Group. Mar. 31, 2011 update; <https://www.comodo.com/Comodo-Fraud-Incident-2011-03-23.html>.
- Langley, A. Enhancing digital certificate security. Google Online Security Blog, 2013; <http://googleonlinesecurity.blogspot.de/2013/01/enhancing-digital-certificate-security.html>.
- Langley, A. Further improving digital certificate security. Google Online Security Blog, 2013; <http://googleonlinesecurity.blogspot.co.uk/2013/12/further-improving-digital-certificate.html>.
- Laurie, B. Improving SSL certificate security. Google Online Security Blog, 2011; <http://googleonlinesecurity.blogspot.co.uk/2011/04/improving-ssl-certificate-security.html>.

Ben Laurie is a software engineer, protocol designer, and cryptographer. He is a founding director of The Apache Software Foundation, a core team member of OpenSSL, a member of the Shmoo Group, a director of the Open Rights Group, Director of Security at The Bunker Secure Hosting, Founder-member of FreeBMD, Visiting Fellow at Cambridge University's Computer Laboratory, and a committer at FreeBSD. Laurie works for Google in London and is currently focused on certificate transparency.

Copyright held by owner/author. Publication rights licensed to ACM. \$15.00

Assessing legal and technical solutions to secure HTTPS.

BY AXEL ARNBAK, HADI ASGHARI,
MICHEL VAN EETEN, AND NICO VAN EIJK

Security Collapse in the HTTPS Market

HYPERTEXT TRANSFER PROTOCOL SECURE (HTTPS) has evolved into the de facto standard for secure Web browsing. Through the certificate-based authentication protocol, Web services and Internet users first authenticate one another (“shake hands”) using a TLS/SSL certificate, encrypt Web communications

end-to-end, and show a padlock in the browser to indicate a communication is secure. In recent years, HTTPS has become an essential technology to protect social, political, and economic activities online.

At the same time, widely reported security incidents—such as DigiNotar’s breach, Apple’s #gotofail, and OpenSSL’s Heartbleed—have exposed systemic security vulnerabilities of HTTPS to a global audience. The Edward Snowden revelations—notably around operation BULLRUN, MUSCULAR, and the lesser-known FLYING PIG program to query certificate metadata on a dragnet scale—have driven the point home that HTTPS is both a major target of

government hacking and eavesdropping, as well as an effective measure against dragnet content surveillance when Internet traffic traverses global networks. HTTPS, in short, is an absolutely critical but fundamentally flawed cybersecurity technology.

While the Heartbleed incident illuminated severe flaws in a widely used crypto-library of HTTPS (OpenSSL), the focus here is on the systemic security vulnerabilities in the HTTPS authentication model that precedes end-to-end encryption. Although some of these vulnerabilities have been known for years, the 2011 security breach at the small Dutch certificate authority (CA) known as DigiNotar was a watershed moment, demonstrating these

theoretical man-in-the-middle vulnerabilities in the wild. Meanwhile, large CAs such as Comodo and Verisign have experienced breaches as well but did not suffer similar consequences as DigiNotar. In fact, some large CAs actually *benefited* from the increased sense of HTTPS insecurity.

Policymakers and technologists are increasingly advocating various solutions to address the security collapse of HTTPS. The European Union is half-way through adopting the first comprehensive legislation on HTTPS in the world. It will acquire immediate binding force in the legal systems of 28 European member states. As most large CAs operate (also) under E.U. jurisdiction, the legislation will impact HTTPS governance globally. In the U.S., on the other hand, attention has focused on technological solutions and industry self-regulation.

To evaluate both legal and technological solutions, an understanding of the economic incentives of the stakeholders in the HTTPS ecosystem, most notably the CAs, is essential.^{2,3} This article outlines the systemic vulnerabilities of HTTPS, maps the thriving market for certificates, and analyzes the suggested regulatory and technological solutions on both sides of the Atlantic. The findings show existing yet surprising market patterns and perverse incentives: not unlike the financial sector, the HTTPS market is full of information asymmetries and negative externalities, as a handful of CAs dominate the market and have become “too big to fail.” Unfortunately, the proposed E.U. legislation will reinforce systemic vulnerabilities, and the proposed technological solutions are far from being adopted at scale. The systemic vulnerabilities in this crucial technology are likely to persist for years to come.

Systemic Vulnerabilities in the HTTPS Authentication Model

Essentially, HTTPS is a two-step process. First, a trust relationship (a handshake) is established between a website and an end user’s browser. This is done with the help of a Transport Layer Security/Secure Sockets Layer (TLS/SSL) certificate containing basic information for authentication purposes. If the Web browser trusts the certificate and the issuing CA, then this authen-

tication handshake succeeds. Second, successful authentication leads to a TLS/SSL-encrypted channel between the website and browser, called a *tunnel*.¹ Thus, the handshake authentication serves as the stepping stone for the confidentiality and integrity that HTTPS seeks to deliver. If the handshake succeeds, then the browser informs the user by, for example, depicting a padlock or a green address bar. If the TLS/SSL certificate or the issuing CA cannot be trusted, then the browser will show a security warning to the end user. The described data flows are shown in Figure 1.

A website that wants to provide HTTPS communications to users needs to obtain a TLS/SSL certificate from a CA. Basically, these certificates are small computer files that contain information on hostname (website), certificate owner (website owner), certificate issuer (CA), validity period, and public key.¹ The method for verification of the identity of a website owner, among others, drives the costs of a certificate and is the key difference between domain validation (DV), organization validation (OV), and extended validation (EV) certificates.²

The stakeholders. HTTPS market involves four central stakeholders, as depicted in Figure 1—Website owners; certificate authorities; Web browsers; and end users.

Website owners decide whether to deploy HTTPS or not, and how securely to implement it on their servers. Deployment is a binary affair from the point of view of the end user. An outdated implementation, as long as the browser accepts it, appears similar to the state-of-the-art implementation. If embedded content from third-party websites (for example, behavioral tracking across websites for advertising) is a part of the revenue model of a website owner, then that operator has a strong incentive not to deploy HTTPS at all. Both deployment and secure implementation vary widely.²⁴

Certificate Authorities. CAs sell TLS/SSL certificates, which come in three categories: root, intermediate/subordinate, and untrusted. *Root CAs* are trusted by default by browsers, after they have solicited for such a status with the browsers and complied with the varying browser CA trust policies.

Intermediate/subordinate CAs are either directly verified by one root CA or they are part of a chain of trust of several intermediate CAs that ultimately ends with one root CA. Certificates of *untrusted CAs* are not issued by a CA linked to a root CA but are mostly self-signed by the owner of a website. Self-signed certificates evoke the “untrusted connection” security warning when served by a website to browsers. CAs are owned by such varying entities as multinational corporations, nation-states, universities, and hacker communities—anyone can start a CA operation relatively easily.

Web-browser vendors. These vendors play a key role in the HTTPS ecosystem. For example, they decide whether to trust a CA inherently, how to respond to a (suspected) CA compromise, and how to implement related trust revocation protocols such as the OSCP (Online Certificate Status Protocol). Over the years, various browsers have developed different certificate policies, leading to varying numbers of root and intermediate CAs inherently trusted per browser.^{3,7}

End users. Because their communications and valuable information are on the line, end users have an interest in seeking HTTPS communications with websites, but they depend to a large degree on security decisions made by the other stakeholders and can exert very little control over HTTPS.^{4,9}

Known CA breaches. On Friday, September 2, 2011, a nocturnal press conference of the Dutch Minister of Internal Affairs marked the beginning of the *DigiNotar* affair. It was triggered by unauthorized access, reportedly by a hacker sympathizing with the government of Iran in mid-July 2011, to the root CA capacity of DigiNotar. When the breach became public three months later, it emerged that in this long period of obscurity 531 false certificates had been created for widely used and highly sensitive domain names such as *.google.com, *.facebook.com, update.windows.com, and *.cia.gov.¹² A small player in the global market with a strong presence in the niche for Dutch e-government services, DigiNotar had root status with all major browser vendors, leading those browsers to trust, by default, corrupt certificates for months.

According to the forensic report, 30 critical updates had not been performed, logging was insufficient, and no antivirus protection was in place at the time of the intrusion.¹³ The damage was probably enormous but cannot be determined with certainty because of the unreliability of the log files. ENISA (European Network and Information Security Agency) speaks of breached communications of “millions of citizens,” particularly connected to the *.google.com certificate, and notes that some experts believe the lives of Iranian activists have been put at risk.⁹ Upon publication of the breach, the trust in the entire range of DigiNotar activities was revoked by all the major browsers.

Comodo. The range of breaches at market-leading CA Comodo also received considerable media attention.¹⁴ The best documented breach was the compromise of Comodo’s UTN-USERFirst-Hardware certificate. According to the Electronic Frontier Foundation (EFF) SSL Observatory, 85,440 public HTTPS certificates were signed directly by UTN-USERFirst-Hardware, and indirectly, the certificate had delegated authority to 50 more intermediate CAs.⁸

Verisign. Another dominant CA, Verisign, was hacked in 2010. The breach was not discovered until February 2012, after new Security and Exchange Commission (SEC) regulations mandated companies to notify investors of intrusions. In reporting its discovery, news agency Reuters quoted a former CTO who said Verisign “probably can’t draw an accurate assessment” of the damage, given the time elapsed since the attack and the vague language in the SEC filing.⁹

Trustwave used its root CA status to enable third parties to issue SSL server certificates for the purpose of monitoring employees. While providing man-in-the-middle capabilities to private entities via sub-CAs does not technically breach the HTTPS trust model, it undermines it. This is especially true when end users are not informed of the monitoring. Trustwave claims this is common practice among root CAs.⁶ This illustrates the “compelled-CA attack” in real life: CAs are in a unique position to enable surveillance of end users.²³

Steven B. Roosa and Stephen Schultze²⁰ report on several other breaches, including GlobalSign, KPN/Getronics, StartSSL, and TurkTRUST. From the known CA breaches, several patterns emerge.

Systematic vulnerabilities of the HTTPS authentication model. The term *systemic vulnerabilities* refers to those vulnerabilities inherent in the HTTPS ecosystem, as opposed to incidental vulnerabilities that have occurred at a particular stakeholder during an isolated incident. Many security experts agree that the security of the HTTPS authentication model and thus the HTTPS ecosystem is systemically flawed as a result of these vulnerabilities.¹

Weakest link. A crucial technical property of the HTTPS authentication model is that any CA can sign certificates for any domain name. In other words, literally *anyone* can request a certificate for a Google domain at any CA anywhere in the world, even when Google itself has contracted one particular CA to sign its certificate. CAs have certain institutional limits to issuing certificates (for example, validation procedures) but no technical ones. If this second google.com certificate is obtained from one of the hundreds of intermediate CAs that link to root CAs trusted by browsers, users will get the familiar HTTPS notification (signaling all is OK).

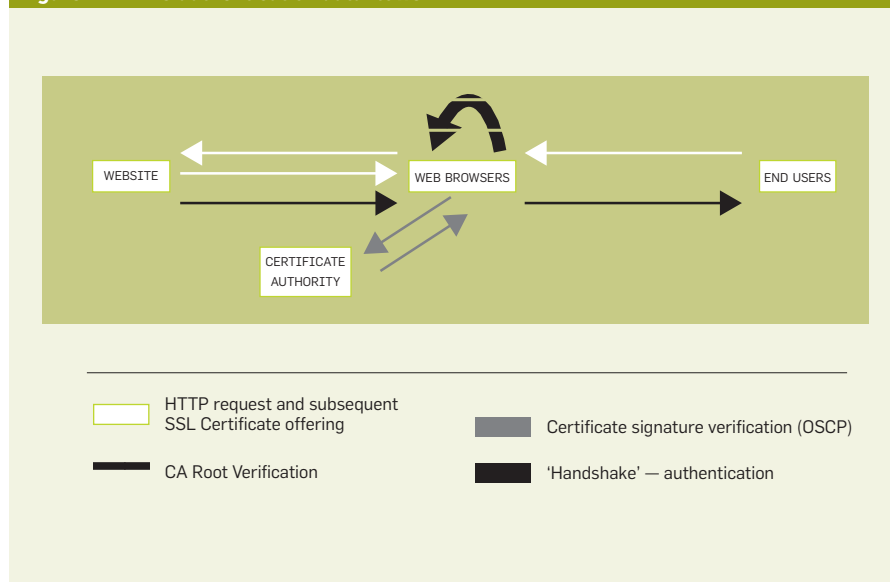
While this ability to sign for any domain name has spurred a flourishing global market for certificates, it

has profound implications for the security of the HTTPS ecosystem, commonly referred to as the *weakest-link* problem: if one CA suffers a breach, the entire ecosystem is under attack.^{9,20} The scenarios for failure are manifold, from CA compromise, misconfiguration, and malpractice to state compulsion.²³

Information asymmetry and ineffective auditing schemes. The recurring information asymmetries are a striking systemic vulnerability, making it very difficult for other stakeholders to know about the security of CAs. The current regulatory regime in the E.U. and auditing obligations worldwide have proven ineffective. The qualified certificate practices of DigiNotar were regulated and passed the periodic audits based upon internationally recognized industry standards. The regulatory and auditing schemes deliver perceived security and enable liability dumping.²⁰

Liability dumping. Websites, browsers, and CAs push damages from security breaches downstream toward end users. CAs, for example, disclaim all liability for losses suffered via inappropriately issued certificates.^{20,25} Because of the negative externalities at play, liability dumping is a common practice, and it is widely criticized for providing wrong incentives or actual security provision.^{1,22} End users bear the burden of these security vulnerabilities and breaches, even though most users are probably unaware of this and

Figure 1. HTTPS authentication data flows.



cannot reasonably be held responsible for evaluating security practices in the HTTPS authentication model.

Mapping the HTTPS Market

To understand these systemic flaws better, a thorough understanding of the market dynamics of HTTPS is essential.¹ It is only in light of such data-driven findings that one can start to reflect on the need for legal and

technical interventions in the current HTTPS ecosystem.

Several studies have surveyed the SSL certificate market. Two of the largest have been the EFF SSL Observatory in 2010 and the University of Michigan's HTTPS ecosystem scans in 2012–2014. Both projects systematically scanned all the IPv4 address space, looking for publicly facing HTTPS servers. They retrieved the SSL

certificates presented by these servers and later parsed and validated them to determine whether different browsers and operating systems would trust that certificate.

In an earlier study³ we used the EFF dataset, which contains approximately 1.5 million trusted certificates, in empirically establishing the number of CAs, the firms that own them, their market shares, and the pricing

Figure 2. Price and market share of DV certificates.

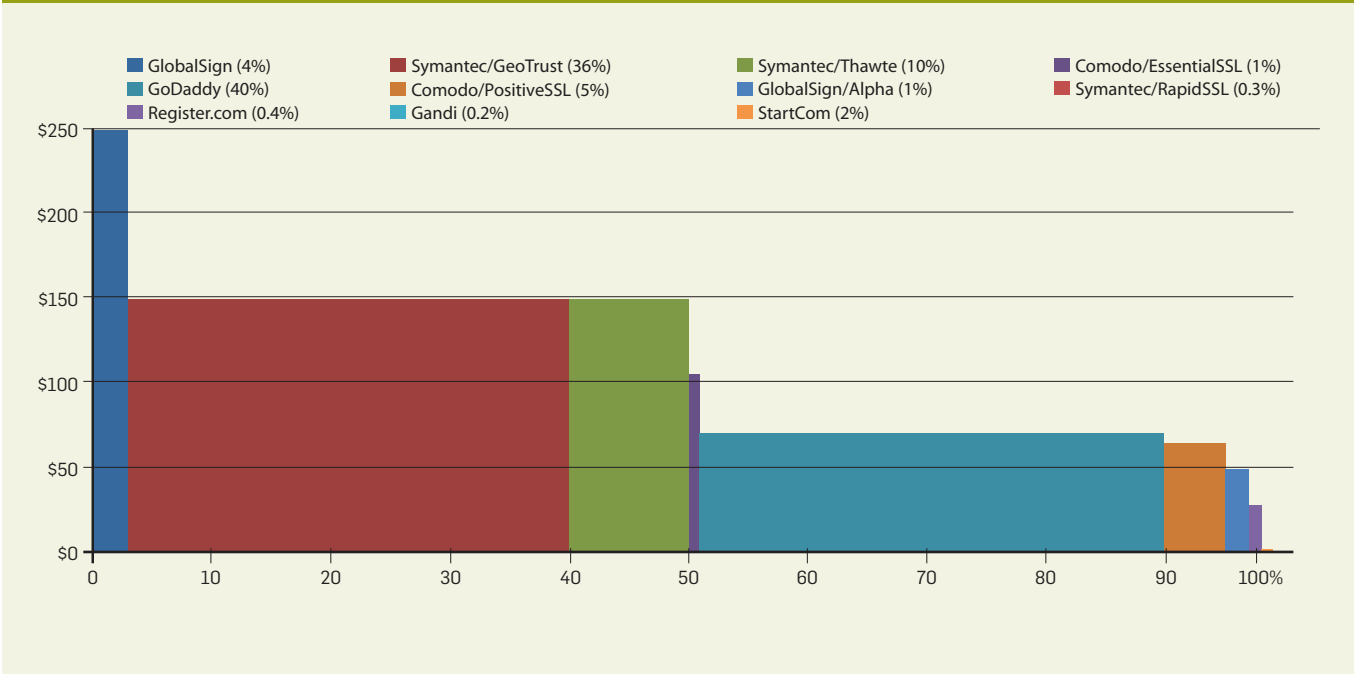
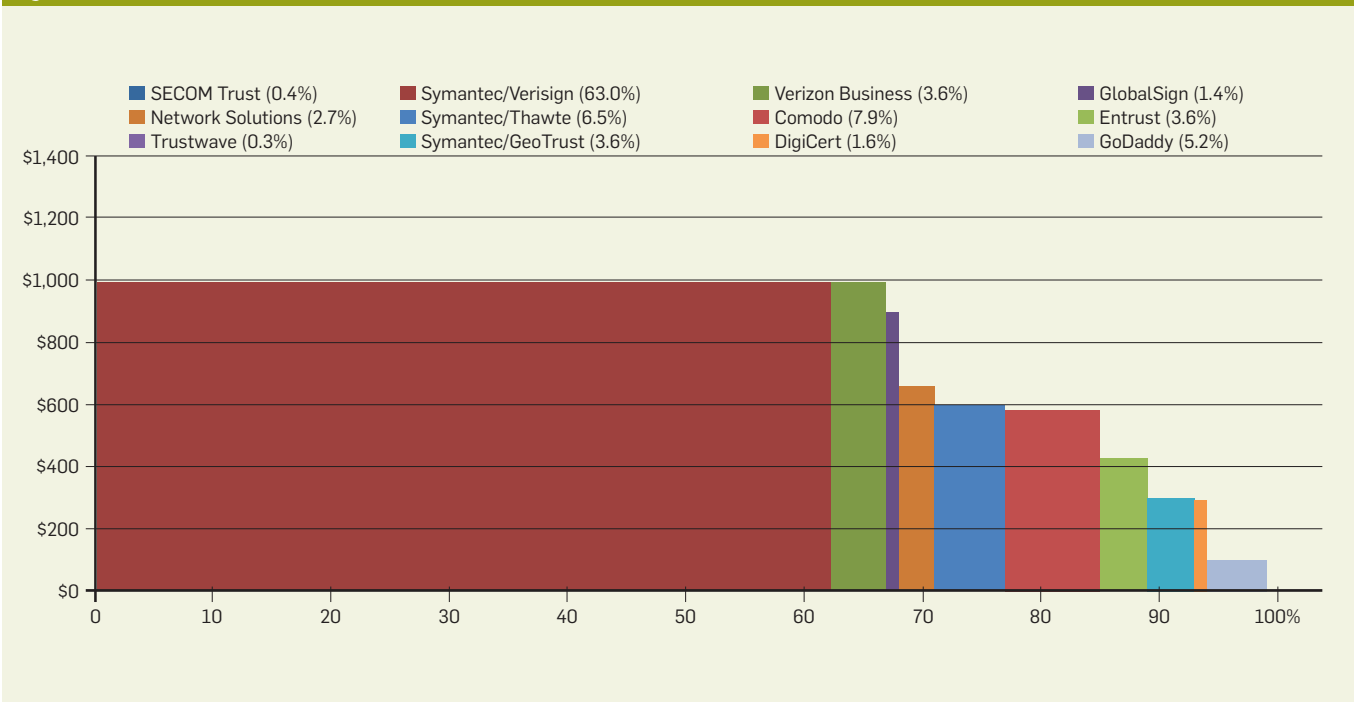


Figure 3. Price and market share of EV certificates.



strategies. We compared our findings against the HTTPS ecosystem scan dataset, which has approximately three million trusted certificates. Durumeric et al.⁷ use this dataset to analyze the HTTPS ecosystem. While the latter scan has collected more certificates than the EFF dataset, this difference mostly reflects a linear growth pattern over time in the number of certificates in use on the Web, and to a limited extent improved scanning methodology. There is a difference of 400,000 certificates if the growth trend in the ecosystem scan data is extrapolated back in time to the EFF data-collection period. Despite these differences, the following patterns are consistent across both datasets.

Many CAs. Foremost, the number of organizations that can issue browser-trusted certificates is high. There are between 1,000 and 2,000 trusted CAs, including root and intermediate CAs. Multiple CAs might be owned by the same organization for a variety of operational and business needs, so the number of issuing organizations is lower. Mapping CAs to organizations leads to an estimated 250 to 700 trusted certificate-issuing organizations, located in 57 countries worldwide. Heterogeneity is often good for an ecosystem, especially in terms of resilience. Because of the weakest-link nature of the HTTPS system, however, this also means many more single points of failure in case of CA compromise or misconfiguration. What is particularly troubling is that a number of the trusted CAs are run by authoritarian governments, among other less trustworthy institutions. Their CAs can issue a certificate for any website in the world, which will be accepted as trustworthy by browsers of all Internet users.

HTTPS market concentration. Second, the market for SSL certificates is highly concentrated, despite the large number of issuers. In fact, both data sets find that around 75% of SSL certificates in use on the public Web have been issued by just three companies: Symantec, GoDaddy, and Comodo. Symantec, the largest commercial CA, owns multiple brands, including Verisign, GeoTrust, Thawte, RapidSSL, and TC TrustCenter. The distribution is heavily skewed, with smaller CAs

Price ranges of different certificates.

Certificate type	Min price	Max price	Average (std. dev.)
DV	\$0	\$249	\$81 (74)
OV	\$38	\$1,172	\$258 (244)
EV	\$100	\$1,520	\$622 (395)

having little or no presence on the public Internet. Power-law distributions, although not surprising in Internet service markets, pose a major risk for the HTTPS ecosystem: if one of the large CAs is compromised, its root status cannot be revoked by browser vendors without massive collateral damage. One particular CA of GoDaddy had signed 26% of all valid HTTPS certificates in use in March 2013. That means if it were compromised, 26% of all websites that rely on HTTPS would need to be immediately issued new certificates.⁷ Otherwise, browsers ought to present certificate warnings or block access to those sites, posing an impossible trade-off for the user between access and security. In other words, such large CAs are truly “too big to fail.”

Weak price competition. Mapping the prices for different certificate brands provides a sense of the degree to which the market is dominated by price competition. Figure 2 shows the price and market share for DV certificate offerings. Symantec/GeoTrust certificates (for example, QuickSSL Premium) sell for \$149 but have a much larger market share than Gandi SSL certificates selling at \$16. OV and EV markets show similar dynamics, as presented in the accompanying table.

The situation is extreme in the EV market, as shown in Figure 3. The market leader, Verisign, sells certificates for approximately \$1,000 and has a 63% share. GoDaddy, offering certificates at a fraction of that price (\$100), captures a mere 5% of the market. (These comparisons have certain limitations, most notably that prices are as advertised by vendors in March 2013, while market shares were from the EFF 2010 dataset.³ The more recent and longitudinal HTTPS ecosystem scan data shows that similar market shares hold over time, with a slight shift of a few percentage points

away from Symantec to cheaper providers.) The differences are intriguing, as certificates themselves are perfect substitutes (within each validation category). The differences might be explained by features bundled with the certificates, discussed in the next section. In sum: the SSL market shows few signs of intense price competition.

Analysis of HTTPS Market Incentives

Various researchers and industry observers have claimed that a “race to the bottom” exists in the HTTPS market: a market dominated by fierce competition pushing prices toward marginal cost, with perverse incentives for security.^{1,22} Some have pointed to this as an explanation for the poor security practices at DigiNotar and other compromised CAs.^{15,18,20,25}

One would indeed expect such a race. Certificates are perfect substitutes, suggesting a completely commoditized market. Also, buyers cannot meaningfully distinguish secure from less secure offerings; and even if they could, buying from a more secure CA cannot protect the site owner against the threat of an attacker fraudulently signing the domain with a certificate from a compromised CA.

The empirical data, however, clearly suggests otherwise, showing market concentration and little price competition. In one sense, it is good news that the market is not driven by a race to the bottom, given the perverse security incentives associated with such a race. Rather than certificates themselves, however, the HTTPS market is driven by:³

- ▶ Bundled security services such as scans of the buyer’s site for malware.
- ▶ Enterprise certificate management services such as support for management and billing of large numbers of certificates.

► Brand reputation as a liability shield against shareholders, regulators, or others who may hold the buyer accountable in the face of security issues.

► Trust or security signals aimed at third parties and end users such as site seals, warranty amounts, and the high price of a certificate itself.

► Higher continuity in case of security failures at the CA, because of the too-big-to-fail dynamic of market-leading CAs.

Knowledgeable buyers understand security in this market is a weakest-link problem and thus determined by the weakest CA. They also understand three of the four market leaders got hacked in recent years and some of the “security” features of these services do not really provide actual security. Nonetheless, buying from the market leaders is still rational, given the liability shield and higher continuity. The price differences are not enough to overrule these advantages. They may be large in a relative sense, but they are modest in absolute terms, compared with other cost components in large firms.

Given the market leaders successfully differentiate their products via, among other things, security-related features, buyers appear to be willing to pay for security. Two classic problems, however, as mentioned earlier, affect the proper alignment of incentives:

► *Information asymmetry prevents buyers from knowing what CAs are really doing.* Buyers are paying for the perception of security, a liability shield, and trust signals to third parties. None of these correlates verifiably with actual security. Given that CA security is largely unobservable, buyers’ demands for security do not necessarily translate into strong security incentives for CAs.

► *Negative externalities of the weakest-link security of the system exacerbate these incentive problems.* The failure of a single CA impacts the whole ecosystem, not just that CA’s customers. All other things being equal, these interdependencies undermine the incentives of CAs to invest, as the security of their customers depends on the efforts of all other CAs.

The most powerful incentive for security seems to be reputation effects, but this does not necessarily make

them more sensitive to the reputation damage caused by breaches. While they have more to lose compared with smaller brands, large CAs are less threatened by the ultimate reputation effect: being removed from the root stores.

Ironically, the security problems that have plagued the HTTPS ecosystem over the past few years, including the breaches at market leaders, may in fact benefit these same market leaders. The breaches have increased the demand for security, and this demand seems to latch onto whatever security signals are available, regardless of their relationship to actual security. All of this may impact the attempts to fix the systemic vulnerabilities of the system. The dominant players might be reluctant—or less eager—to push for adoption of one of the proposed technological solutions. This is not to suggest that market leaders will act against them, but rather that the status quo works quite well for them.

Improving HTTPS Governance

In the aftermath of these CA breaches, policymakers and technologists have suggested regulatory and technical solutions to the systemic vulnerabilities of HTTPS. Let’s evaluate these solutions in light of the market-incentive analysis.

Regulatory solutions. The HTTPS authentication model is by and large unregulated in both the U.S. and the E.U. This is bound to change in the near future. Each entity has opted for a completely different approach: the U.S. gives priority to technological solutions and lets industry self-regulate in the meantime. The European Commission (the executive branch of the E.U.), on the other hand, proposed the Electronic Identification and Trust Services Regulation in June 2012. Unlike the more common E.U. *directives* that require implementation in national law, *regulations* acquire direct binding force of law in all E.U. member states upon adoption in Brussels. In April 2014, the European Parliament adopted substantial amendments to the commission proposal, leaving the regulation only for the E.U. Council (national governments of the E.U.) to approve.

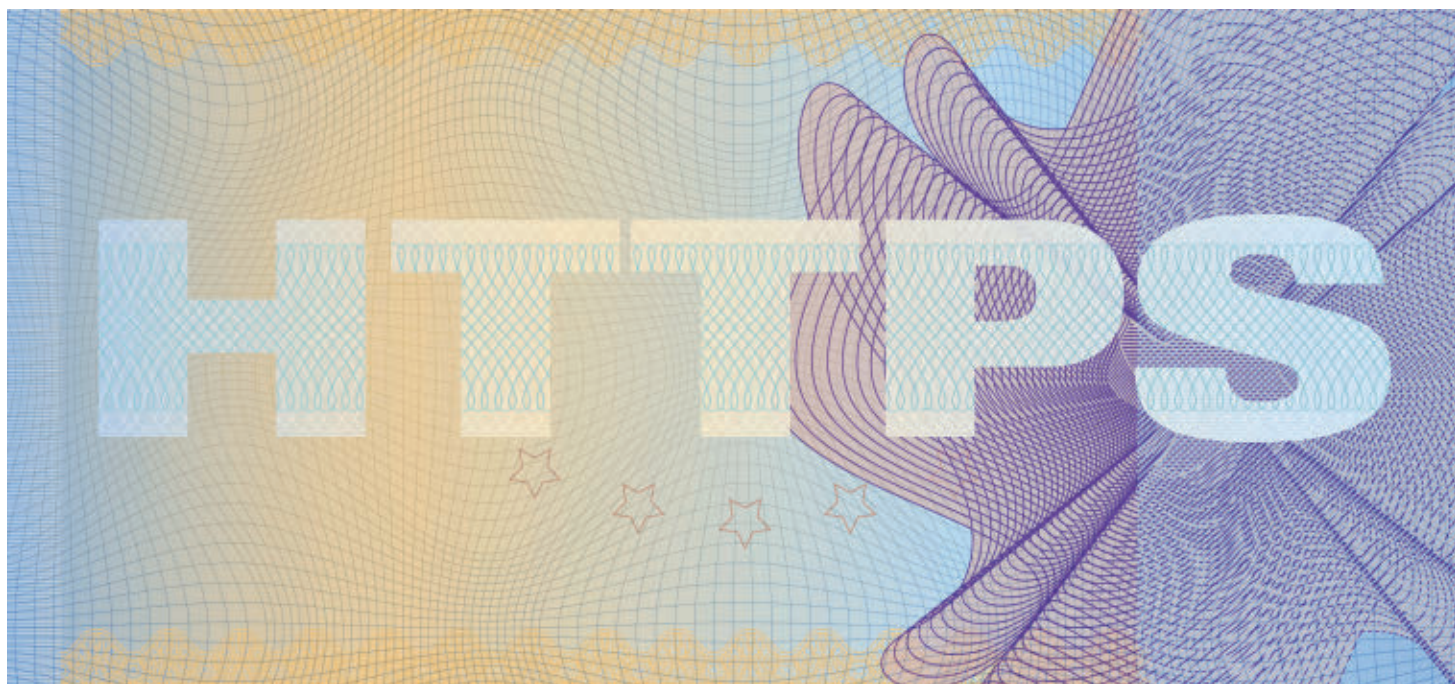
Here, we outline the scope, underly-

ing values, security requirements, security breach notification, and liability regime of the E.U. proposal,¹⁰ as well as the recent proposals by Mozilla for “chain of trust transparency.”^{2,3}

Scope. The E.U. proposal regulates *trust service providers*, including CAs.¹⁰ All major CAs appear to fall within both U.S. and E.U. jurisdiction.³ While inherently local, regulation may therefore be an effective instrument to address the observed market failures and positively influence HTTPS security globally. Other critical stakeholders in the HTTPS ecosystem, however, such as browser vendors and website operators, remain unregulated in the proposal. This limited scope impacts the proposed security measures considerably.

Underlying values. The E.U. proposal focuses on availability interests to boost trust in e-commerce, neglecting confidentiality and integrity concerns connected to the systemic HTTPS vulnerabilities already outlined. Apart from failing to observe privacy and communications secrecy obligations under the E.U. Charter of Fundamental Rights, the proposal completely ignores the Snowden revelations. The BULLRUN and MUSCULAR disclosures have made clear that HTTPS significantly raises the costs of mass dragnet surveillance and has been a primary target of intelligence agency subversion. Large Internet companies have now started or accelerated efforts to encrypt communication paths both with users and within their own networks using TLS. The April 2014 E.U. Parliament amendments not only ignore these developments, but also make explicit that the HTTPS provision is “entirely voluntary” for Web services (recital 67).

Security requirements. The E.U. proposal introduces new obligations for CAs to adopt security requirements. Their details will be determined by the European Commission in a so-called implementing act. While such delegation to the executive branch provides some flexibility to adapt requirements to new technological developments, the E.U. proposal fails to specify regulatory priorities or underlying values. Moreover, the April 2014 parliament amendments literally state that “industry-led initiatives (for example, CA/Browser Forum)” influence such re-



quirements (recital 67). Naming a CA industry group as influential in a law that seeks to address failing security practices of CAs indicates control by dominant market players.

Security breach notification (SBN). In theory, SBNs help minimize the damage after a breach has occurred and provide incentives for organizations to invest in information security upfront. The E.U. proposal introduces an SBN regime stating that notification needs to occur “within 24 hours” to relevant authorities if the breach “has a significant impact,” a concept that is not defined in the law. The general public is informed when a breach harms the “public interest” (also undefined). Again, the European Commission will determine those details, but the parliament proposal states that CAs should be subject to “light-touch and reactive ex-post supervisory activities” and that there exists “no general obligation to supervise non-qualified service providers” (that is, CAs offering certificates for HTTPS).

Aforementioned information asymmetries and CA breaches render defensible a strict regime for notifications—which types of breaches should be made public by default, for example. Experiences with SBN legislation in the U.S., moreover, suggest SBNs need to be complemented with punitive (for example, sanction and liability regimes) and proactive

enforcement (for example, as part of annual reporting) to create real incentive to notify—and avoid non-compliance by less well-intentioned companies.^{1,22} In addition, reputation losses might not affect major CAs that do not risk being thrown out of root stores for nonreporting. Reporting not only breaches, but also the vulnerabilities that led to them, would be a major step forward, as would a scheme of responsible disclosure. Such lessons are not included in the E.U. proposals or considerations. Moreover, the parliament has further weakened the SBN regime by mandating light-touch and ex-post supervision. Again, these amendments indicate capture of the regulatory process by dominant CAs.

Liability. As already observed, liability for security breaches is disclaimed across the HTTPS ecosystem and transferred through terms and conditions to end users. The 2012 E.U. Commission proposal sought to address such liability dumping by imposing a strict liability regime on CAs for “any direct damage,” with CAs bearing the burden of proving they handled the situation non-negligently. The 2014 parliament amendments reverse this burden of proof; customers and users now have to prove malicious intent or negligence at CAs post-breach. Moreover, CAs are allowed to transfer liability in their

terms and conditions to end users. Astonishingly, the parliament explicitly codifies liability dumping. Again, there are traces of regulatory capture at the E.U. parliament.

The weakest-link problem of HTTPS creates more fundamental problems with security through liability: small CAs will be unable to conduct business with large corporations processing vast amounts of sensitive data. Consider DigiNotar with its annual budget of a few million U.S. dollars; it could never cover damages for the rogue certificates that were issued for Google, Facebook, Skype, cia.gov, among others, in the midst of its security breach. Smart CAs will thus circumvent liability by creating subsidiary special-purpose companies that bear full liability and can easily file for bankruptcy. Indeed, DigiNotar quickly went bankrupt post-breach, while its parent company Vasco has escaped unscathed.

Tackling fundamental issues with liability regimes requires carefully crafted policies or broad mandates for enforcement. Liability should be matched with security requirements and distributed among all stakeholders: domain owners should have incentives to protect their assets through HTTPS offering and implementation,² while browsers should strengthen their CA policies (as discussed later). The European

Commission failed to consider such fundamental drawbacks, and the parliament amendments make matters worse by codifying liability dumping and reversing the burden of proof.


Chain of trust transparency. Unrelated to the E.U. proposals, Mozilla has proposed the so-called “chain of trust transparency.” As discussed earlier, one cannot assure that HTTPS communications are subject to systematic but unnoticed surveillance without transparency,²³ but today it is only starting to emerge through various (research) projects such as the browser plug-in CertPatrol for Firefox.

In a recent amendment to its CA policy, Mozilla requires that subordinate CA certificates “either be technically constrained or be publicly disclosed and audited.”¹⁹ Subordinate CAs, in other words, must either be constrained to issue certificates for only a (small set of) domain name(s)—on internal networks, for example—or their chain of trust must be publicly disclosed and audited. The aim is to hold subordinate CAs to similar standards as root CAs and make a root CA accountable for all the sub-certificates it signs. Existing subordinate CA certificates were given until May 15, 2014, to comply, so it is too early to observe how Mozilla enforces noncompliance. Nonetheless, chain of trust transparency warrants at least consideration and, from a theoretical perspective, encouragement throughout the HTTPS ecosystem.²¹ So far, it has not been part of any regulatory proposal.


Technology solutions. A host of technological solutions to the systemic vulnerabilities of the current system are being developed. Among the most prominent are Convergence, Perspectives, DANE, Sovereign Keys, Certificate Transparency, Public Key Pinning, and TACK. From the perspective of governance, we can make several general observations:

- ▶ All proposals attempt to solve the weakest-link problem by introducing another authority to check whether the certificate that is validated through the normal HTTPS process is indeed the correct one.

- ▶ All proposals reduce the information asymmetry of buyers and users, versus the CAs, by systematically un-



Not unlike the financial sector, the HTTPS market is full of information asymmetries and negative externalities, as a handful of CAs dominate the market and have become “too big to fail.”



covering suspect certificates.

- ▶ All proposals can function on top of the current CA system, leaving it in place or depending on it; a subset can also replace it.

- ▶ All proposals can follow incremental adoption paths (albeit some are a lot more difficult than others), and all need support from browsers.

None of these solutions is close to large-scale adoption. That said, they do seem promising in terms of addressing the current weaknesses, especially the weakest-link problem, for which regulatory solutions appear ineffective. Therefore, in the long run they are preferable, and it is relevant to assess how they relate to the incentives of the HTTPS stakeholders. Some scholars predict multiple proposals will eventually be adopted.⁵

As argued earlier, the insecure status quo can be beneficial for market leaders. In light of this, one might assume that CAs are not particularly keen on actively helping any of these proposals along, especially the ones that theoretically could make them obsolete. In practice, however, some CAs are involved in developing potential solutions—for example, DigiCert and Comodo are experimenting with Certificate Transparency.¹⁶ Other proposals require nontrivial activities on the side of the domain owner, which may be done by their CA as a complementary service to current business models.

Furthermore, each proposal is intensely debated in relation to browser performance. Any form of large-scale adoption requires default support by browser vendors. Google and Mozilla have been particularly active in this area.

While none of these solutions is easy to scale, there are benefits for early adopters, a key requirement for any solution to take off. Whether the costs are worth it depends on the kinds of threats HTTPS stakeholders want to defend themselves against. An average cybercriminal might not be interested in breaching a CA and manipulating network traffic already encrypted through HTTPS, as financially attractive information can be acquired through more cost-effective attacks.^{11,17} From previous breaches, it appears that state-sponsored at-

tackers and large corporations, rather than profit-driven criminals, are more likely to engage in the complex man-in-the-middle attacks in the realm of HTTPS. For some user groups and domains, such adversaries make early adoption attractive.

Conclusion

Recent breaches at CAs have exposed several systemic vulnerabilities and market failures inherent in the current HTTPS authentication model: the security of the entire ecosystem suffers if any of the several hundreds of CAs is compromised (weakest link); browsers are unable to revoke trust in major CAs (“too big to fail”); CAs manage to conceal security incidents (information asymmetry); and ultimately customers and end users bear the liability and damages of security incidents (negative externalities).

Understanding the market and value chain for HTTPS is essential to address these systemic vulnerabilities. The market is highly concentrated, with very large price differences among suppliers and limited price competition. Paradoxically, the current vulnerabilities benefit rather than hurt the dominant CAs, because among others, they are too big to fail.


In terms of solutions, the E.U. has opted for a regulatory response, while the U.S. prefers industry self-regulation and technological solutions. In general, the technological solutions aim to solve the weakest-link security problem of the HTTPS ecosystem. Several proposals are promising, but none is near large-scale adoption. Industry self-regulation has only augmented market failures, rather than solve them.

The proposed E.U. regulation does not consider the role of all stakeholders in the HTTPS ecosystem, thus reinforcing systemic vulnerabilities by creating new long-term institutional dependencies on market-leading CAs. The April 2014 E.U. Parliament amendments make matters much worse. The E.U. Parliament seems to have been successfully captured by CA lobbying efforts.

Regardless of major cybersecurity incidents such as CA breaches, and even the Snowden revelations, a sense of urgency to secure HTTPS seems nonexistent. As it stands, major CAs

continue business as usual. For the foreseeable future, a fundamentally flawed authentication model underlies an absolutely critical technology used every second of every day by every Internet user. On both sides of the Atlantic, one wonders what cybersecurity governance really is about.

Acknowledgments

The authors thank Bernhard Amann, Ian Brv own, Peter Eckersley, Edward Felten, Sharon Goldberg, Joris van Hoboken, Ralph Holz, Chris Hoofnagle, Kees Keuzenkamp, Samad Khatibi, Arman Noroozian, Bruce Schneier, Stephen Schultze, Christopher Soghoian, Sid Stamm, Marcelo Thompson, and participants of TPRC 2012, WEIS 2013, two workshops at the Berkman Center in Spring 2014, 29c3, a UC Berkeley TRUST Seminar January 2013, and an HKU Law & Tech Talk, February 2013. The authors are solely responsible for this article. 

Related articles on queue.acm.org

Securing the Edge

Avi Freedman

<http://queue.acm.org/detail.cfm?id=640149>

The Seven Deadly Sins of Linux Security

Bob Toxen

<http://queue.acm.org/detail.cfm?id=1255423>

Security in the Browser

Thomas Wadlow and Vlad Gorelik

<http://queue.acm.org/detail.cfm?id=1516164>

References

- Anderson, R.J. *Security Engineering: A Guide to Building Dependable Distributed Systems*. Wiley, 2008.
- Arnbak, A. and van Eijk, N. Certificate Authority collapse: regulating systemic vulnerabilities in the HTTPS value chain. Research Conference on Communication, Information and Internet Policy (TPRC), 2012; http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2031409.
- Asghari, H., van Eeten, M.J., Arnbak, A.M. and van Eijk, N.A. Security economics in the HTTPS value chain, 2013; http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2277806.
- Bakos, Y., Marotta-Wurgler, F. and Trossen, D. Does anyone read the fine print? Testing a law and economics approach to standard form contracts. In *Proceedings of the Fourth Annual Conference on Empirical Legal Studies*, 2009.
- Bonneau, J. Fixing HTTPS: new models for distributing transport security policy. Center for Information Technology Policy (CITP) Seminar, 2014; https://docs.google.com/presentation/d/1dxWwKUOVj01MnQJkkyxCS03VfP_kmPeAmneJ9Kld-M/edit?usp=sharing.
- Constantin, L. Trustwave admits issuing man-in-the-middle digital certificate; Mozilla debates punishment. *ComputerWorld* (Feb. 8, 2012); http://www.computerworld.com/s/article/9224082/Trustwave_admits_issuing_man_in_the_middle_digital_certificate_Mozilla_debates_punishment.
- Durumeric, Z., Kasten, J., Bailey, M. and Halderman, J.A. Analysis of the HTTPS certificate ecosystem.

In *Proceedings of the Internet Measurement Conference*, 2013.

- Eckersley, P. Iranian hackers obtain fraudulent HTTPS certificates: How close to a Web security meltdown did we get? Electronic Frontier Foundation, 2011; <https://www.eff.org/deeplinks/2011/03/iranian-hackers-obtain-fraudulent-https>.
- ENISA. Operation Black Tulip: Certificate Authorities lose authority, version 2 (Dec. 2011); <http://www.enisa.europa.eu/media/news-items/operation-black-tulip>.
- European Union. Electronic identification and trust services for electronic transactions in the internal market. Amended proposal, 2014; 2012/0146(COD), A7-0365/201; <http://www.europarl.europa.eu/sides/getDoc.do?type=TA&language=EN&reference=P7-TA-2014-0282#title3>
- Florencio, D. and Herley, C. Where do all the attacks go? Workshop on Economics of Information Security (2011); <http://research.microsoft.com/pubs/149885/WhereDoAllTheAttacksGo.pdf>.
- Fox-IT. DigiNotar Certificate Authority breach (Sept. 5, 2011); <http://www.rijksoverheid.nl/documenten-en-publicaties/rapporten/2011/09/05/diginotar-public-report-version-1.html>.
- Fox-IT. Black Tulip—Report of the investigation into the DigiNotar Certificate Authority breach; <http://www.rijksoverheid.nl/documenten-en-publicaties/rapporten/2012/08/13/black-tulip-update.html>.
- InfoSecurity. Comodo admits two more registration authorities hacked, 2011; <http://www.infosecurity-magazine.com/view/16986/comodo-admits-two-more-registration-authorities-hacked>.
- Kelkman, O.M. DNSSEC Musings: DigiNotar, DANE and Deployment. NLnet Labs, 2013; http://conference.apnic.net/_data/assets/pdf_file/0005/58901/dnssec-diginotar-dane_1361864377.pdf.
- Langley, A. Certificate Transparency. ImperialViolet; <http://www.imperialviolet.org/2012/11/06/certrans.html>.
- Langley, A. Real World Crypto 2013. ImperialViolet; <http://www.imperialviolet.org/2013/01/13/rwc03.html>.
- Mills, E. Google users in Iran targeted in SSL spoof. CNET (Aug. 30, 2011); http://news.cnet.com/8301-27080_3-20099421-245/google-users-in-iran-targeted-in-ssl-spoof/.
- Mozilla. Mozilla CA certificate policy, version 2.1 (Feb. 14, 2013); <http://www.mozilla.org/projects/security/certs/policy/>.
- Roosa, S.B., Schultze, S. The “Certificate Authority” trust model for SSL: a defective foundation for encrypted Web traffic and a legal quagmire. *Intellectual Property & Technology Law Journal* 22. 11 (2010), 3.
- Roosa, S.B. and Schultze, S. Trust Darknet: control and compromise in the Internet’s Certificate Authority Model, 2013; <http://ssrn.com/abstract=2249042>.
- Shapiro, C. and Varian, H. *Information Rules*. Harvard Business School Press, 1998.
- Soghoian, C. and Stamm, S. Certified lies: detecting and defeating government interception attacks against SSL. In *Financial Cryptography and Data Security*. Springer, 2012, 250-259.
- Trustworthy Internet Movement. SSL-Pulse. Survey of the SSL implementation of the most popular websites, 2014; <https://www.trustworthyinternet.org/ssl-pulse/>.
- Vratonjic, N., Freudiger, J., Bindschaedler, V. and Hubaux, J.-P. The inconvenient truth about Web certificates. In *Proceedings of the Workshop on Economics of Information Security*, 2011.

Axel Arnbak is a cybersecurity and information law researcher at the University of Amsterdam, Research Fellow at the Berkman Center (Harvard University) and at CITP (Princeton University).

Hadi Asghari is a researcher in the economics of cybersecurity at Delft University of Technology, Faculty of Technology, Policy and Management.

Michel Van Eeten is a professor of governance of cybersecurity at Delft University of Technology, Faculty of Technology, Policy and Management.

Nico Van Eijk is professor of media and telecommunications law and director of the Institute for Information Law, IVIR, Faculty of Law, University of Amsterdam.

Copyright held by owners/author(s). Publication rights licensed to ACM. \$15.00.

Q Article development led by [acmqueue](http://queue.acm.org)
queue.acm.org

Routing security incidents can still slip past deployed security defenses.

BY SHARON GOLDBERG

Why Is It Taking So Long to Secure Internet Routing?

THE BORDER GATEWAY PROTOCOL (BGP) is the glue that holds the Internet together, enabling data communications between large networks operated by different organizations. BGP makes Internet communications global by setting up routes for traffic between organizations—for example, from Boston University’s network, through larger ISPs such as Level3, Pakistan Telecom, and China Telecom; then on to residential networks such as Comcast or enterprise networks such as Bank of America.

While BGP plays a crucial role in Internet communications, it remains surprisingly vulnerable to attack. The past few years have seen a range of routing incidents that highlight the fragility of routing with

BGP. They range from a simple misconfiguration at a small Indonesian ISP that took Google offline in parts of Asia,³² to a case of BGP-based censorship that leaked out of Pakistan Telecom and took YouTube offline for most of the Internet,² to a routing error that caused a large fraction of the world’s Internet traffic to be routed through China Telecom,⁶ to highly targeted traffic interception by networks in Iceland and Belarus.³⁴

People have been aware of BGP’s security issues for almost two decades and have proposed a number of solutions, most of which apply simple and well-understood cryptography or whitelisting techniques. Yet, many of these solutions remain undeployed (or incompletely deployed) in the global Internet, and the vulnerabilities persist. Why is it taking so long to secure BGP?

The answer to this question lies in the fact that BGP is a global protocol, running across organizational and national borders. As such, it lacks a single centralized authority that can mandate the deployment of a security solution; instead, every organization can autonomously decide which routing security solutions it will deploy in its own network. Thus, the deployment becomes a coordination game among thousands of independently operated networks; this is further complicated by the fact that many security solutions do not work well unless a large number of networks deploy them.

Routing Primer

BGP enables networks to route to destination *IP prefixes*. An IP prefix is a set of Internet Protocol addresses with a common prefix that is n bits in length. For example, the set of IP addresses {8.0.0.0, 8.0.0.1, ..., 8.255.255.255} is written as 8.0.0.0/8, where the notation /8 (“slash eight”) implies that the first eight bits (the prefix) are common to all addresses in the set (in this case, those beginning with the numeral 8.). IP prefixes can have variable lengths, and the addresses in one IP prefix may



be entirely contained in another IP prefix. For example, the prefix 8.8.8.0/24, which is allocated to Google, is entirely contained in prefix 8.0.0.0/8, which is allocated to Level3; we say that IP prefix 8.0.0.0/8 *covers* IP prefix 8.8.8.0/24.

Longest-prefix-match routing. To decide how to forward an IP packet, an Internet router identifies the *longest* IP prefix that covers the destination IP address in the packet. For example, a packet with destination IP address 8.8.8.8 would be forwarded on the route to the longer 24-bit IP prefix 8.8.8.0/24 rather than to the shorter eight-bit IP prefix 8.0.0.0/8.

Autonomous systems. BGP allows autonomous systems (ASes) to discover routes to destination IP prefixes. ASes are large, autonomous networks operated by different organizations. Each AS is assigned a different AS number (for example, Google [AS 15169], China Telecom [AS 4134], Comcast [AS 7922], Boston University [AS 111], Verizon Wireless [AS 22394 and AS 6167]) and is allocated a set of IP prefixes. An AS is the *origin* for a prefix that is allocated to it.

ASes are interconnected, creating a graph where nodes are ASes and edges are the links between them, as in Figure 1. ASes discover routes to IP prefixes through the AS-level graph via BGP *announcements* they receive from their neighbors. Each BGP announcement contains the AS-level path the neighbor AS uses to reach the destination IP prefix. In Figure 1,^{17,41} IP prefix

66.174.161.0/24 is allocated to Verizon Wireless, whose AS 22394 *originates* the prefix into the routing system by sending the following BGP announcement to AS 6167:

```
22394
66.174.161.0/24
```

AS 6167 selects the route and forwards all traffic for prefix 66.174.161.0/24 to its neighbor AS 22394. AS 6167 then appends its own name to the path and announces the path to its neighbors AS 2828 and AS 3356 as:

```
6167, 22394
66.174.161.0/24
```

Level3's AS 3356 selects the path and announces it onward to its neighbor AT&T AS 7018 as:

```
3356, 6167, 22394
66.174.161.0/24
```

This process continues, and the AS-level path to prefix 66.174.161.0/24 propagates through the network.

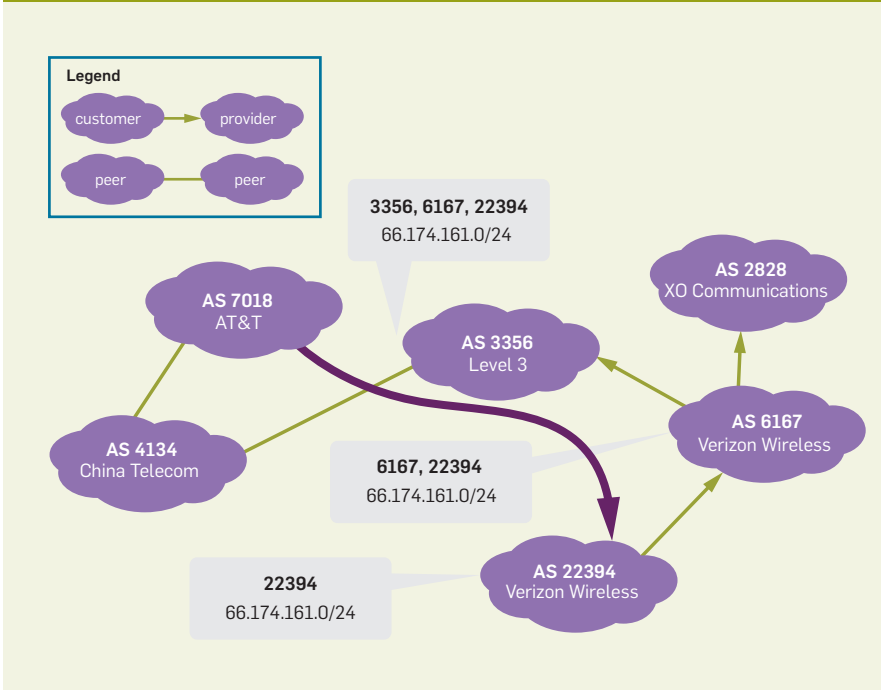
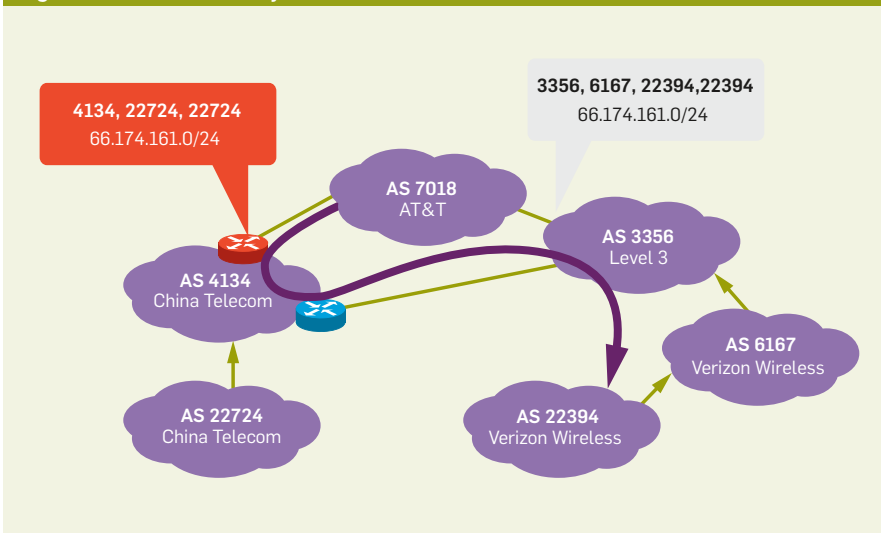
Business relationships and routing policies. If an AS learns multiple routes to a particular IP prefix, then it chooses a single most-preferred route using its local routing policies. BGP provides ASes with considerable flexibility in how they select their routes. Routing decisions are typically independent of the performance of the route at a given instant; instead, they are based

on route length (that is, the number of ASes on the AS-level path) and the price of forwarding traffic to the neighbor that announced the route.

The price of forwarding traffic depends on the *business relationships*^{9,19,20} between neighboring ASes. While many business relationships exist, two are particularly relevant here. The first is a *customer-provider* relationship, where the customer AS pays the provider AS to both send and receive traffic; Level3 and Verizon Wireless have a customer-provider relationship, represented by a directed edge in Figure 1 from the customer (Verizon Wireless) to the provider (Level3). The second relevant business relationship is *settlement-free peering*, where two ASes agree to transit each other's traffic for free; Level3 and AT&T have a peering relationship, represented by an undirected edge in Figure 1.

An AS will almost always avoid forwarding traffic from one neighbor to another if it cannot generate revenue by doing so; for example, China Telecom's AS 4134 in Figure 1 will not carry traffic from its peer, Level3 (AS 3356), to its other peer, AT&T (AS 7018), because neither neighbor pays China Telecom for this service. As such, China Telecom will not send a BGP announcement to AT&T (AS 7018) for the route to the prefix it learned from Level3 (AS 3356) in Figure 1.

This economically motivated behavior^{9,19,20} is often generalized as the following *rule of thumb*: AS *a* will typically

Figure 1. Excerpt of the AS-level graph.^{17,41}Figure 2. China Telecom hijacks Verizon Wireless.¹⁷

announce a route to neighboring AS n only if: (1) n is a customer of a ; (2) the route is for a prefix originated by a ; or (3) the route is through a customer of a .

Attacks on BGP

BGP was designed in the early 1990s—a simpler time, when the Internet was less contentious. As a result, BGP lacks basic authentication mechanisms, making it highly vulnerable to attack. We illustrate these vulnerabilities using several real-life routing incidents.

Hijacks. BGP lacks mechanisms to authenticate the allocation of IP prefixes to autonomous systems; a prefix

hijacker exploits this by originating a prefix that was not allocated to its AS. Hijacks can be classified into two types: *prefix* and *subprefix*.

Prefix hijacks. In a prefix hijack, the hijacking AS originates the exact same prefix as the AS(es) that is legitimately allocated the victim IP prefix. The bogus BGP announcement originated by the hijacking AS will be disseminated throughout the routing system, and the other ASes will use their local policies to choose between routes to the legitimate origin AS(es) and bogus routes originated by the hijacking AS.

For 18 minutes on April 8, 2010,

China Telecom launched prefix hijacks for 15% the Internet's prefixes.^{6,17} While there is no evidence this incident resulted from anything other than a misconfiguration, it provides an instructive example of a “classic” prefix hijack.¹⁷ Figure 2 shows one of the hijacks: China Telecom's AS 22724 hijacks Verizon Wireless's prefix 66.174.161.0/24. The bogus route originated by AS 22724 propagates through the AS-level graph and is eventually selected by AT&T because it is shorter than the legitimate route originating by Verizon Wireless's AS 22394. Meanwhile, Level3 selects the legitimate route, because it is shorter than the bogus route. Thus, network traffic splits between the hijacking AS and the legitimate origin AS, with the nature of the split depending on routing policies used by individual ASes and the topology of the AS-level graph.

Subprefix hijacks. A far nastier attack, the subprefix hijack can potentially allow the hijacker to intercept 100% of the network traffic destined for the victim IP prefix. In a subprefix hijack, the hijacking AS originates a subprefix of the victim's IP prefix—that is, a prefix that is covered by the victim IP prefix.

Perhaps the most famous subprefix hijack occurred on February 24, 2008, when Pakistan Telecom took YouTube offline. The incident² began when Pakistani authorities demanded YouTube to be censored within Pakistan. To accomplish this, Pakistan Telecom's AS 17557 launched a subprefix hijack by originating the subprefix 208.65.153.0/24 of YouTube's prefix 208.65.153.0/22 to its customer ASes in Pakistan (for example, Aga Khan University, Lahore Stock Exchange, Allied Bank Pakistan), as in Figure 3.^{37,41} This meant traffic destined for YouTube's servers in AS 36561 would instead be forwarded to the *longer* IP prefix originated by Pakistan Telecom's AS 17557, where traffic could then be dropped.

Events took an unexpected turn when Pakistan Telecom's bogus BGP announcement leaked out of Pakistan. PCCW, a large ISP that provides global network connectivity to Pakistan Telecom, received the bogus routing announcement, selected the bogus route, and announced it to its own neighbors. Because the bogus route

was for a longer prefix (/24) than the legitimate route (/22), longest-prefix-match routing meant the bogus route was *always* more preferred by the legitimate route, and within minutes, at least two-thirds of the Internet was sending its YouTube traffic to Pakistan.² The incident was eventually resolved via manual intervention of network operators at YouTube, PCCW, and other ISPs worldwide.

Detecting hijacks. Prefix hijacks might seem to be easy to detect, just by checking that a particular prefix is originated by more than one AS. A single prefix, however, might be originated by multiple ASes for legitimate reasons (for example, multiple ASes in a disparate part of the AS-level topology might originate a single prefix to reduce latency, so other ASes can get “closer” to the prefix). In some situations, only the legitimate holder of a prefix can be absolutely certain that a prefix is being hijacked. The identification of hijacks using anomaly-detection techniques is an active area of research.^{3,21}

Route leaks are a separate class of commonly observed routing incidents.²⁸ These leaks are especially interesting because they do not involve the announcement of a bogus route. Instead, the perpetrator announces a legitimate route that it is actually using, but announces it to *too many* of its neighbors. The perpetrator is then overwhelmed by a flood of traffic from neighbors that select the leaked route.

Figure 4 illustrates such an incident involving Moratel (AS 23947), a local ISP based in Indonesia.^{32,33} Moratel is not designed to transit large volumes of traffic from an international communications provider such as PCCW (AS 3491) to an important content provider such as Google (AS 15169). Per the rule of thumb in the first section, Moratel therefore should not announce its route to prefix 8.8.8.0/24 to its provider PCCW.

On November 6, 2012, however, a misconfiguration at Moratel did just that, “leaking” the route

23947, 15169
8.8.8.0/24

to PCCW. Understanding why this had impact requires knowledge of PCCW’s local routing policies. Many

routers,^{19,20} likely including those in PCCW’s AS, are configured to prefer a route through a neighboring customer over one through a neighboring settlement-free peer. By forwarding traffic through its customers, an AS can generate more revenue. As such, PCCW’s routers preferred the customer route through Moratel over the usual settlement-free peering route directly to Google’s AS 15169. As a result, Moratel received a huge volume of network traffic from PCCW, which quickly took parts of Moratel’s network offline and rendered 8.8.8.0/24 unreachable for PCCW and some of its neighbors, including AS 4436.

Impact of routing incidents. Inci-

dents of this type can impact routing in different ways, which can be classified as *blackholes* or *interception*.

Blackhole. In a blackhole, network traffic stops at the perpetrator AS and never reaches its legitimate destination; blackholes happen because BGP routing decisions are typically independent of the instantaneous performance of the route. Blackholes result in network outages that are visible to end users. The Moratel incident is a classic example of a route leak leading to a blackhole. Hijacks can also cause blackholes; the Pakistan Telecom/YouTube incident created a blackhole because all of Pakistan Telecom’s neighbors had selected its bogus route,

Figure 3. Pakistan Telecom hijacks YouTube.^{2,37,41}

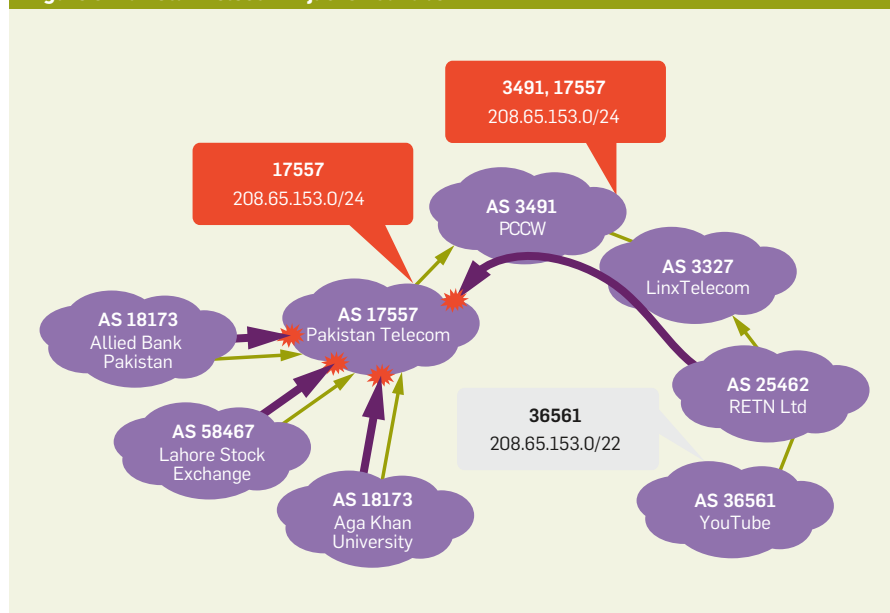
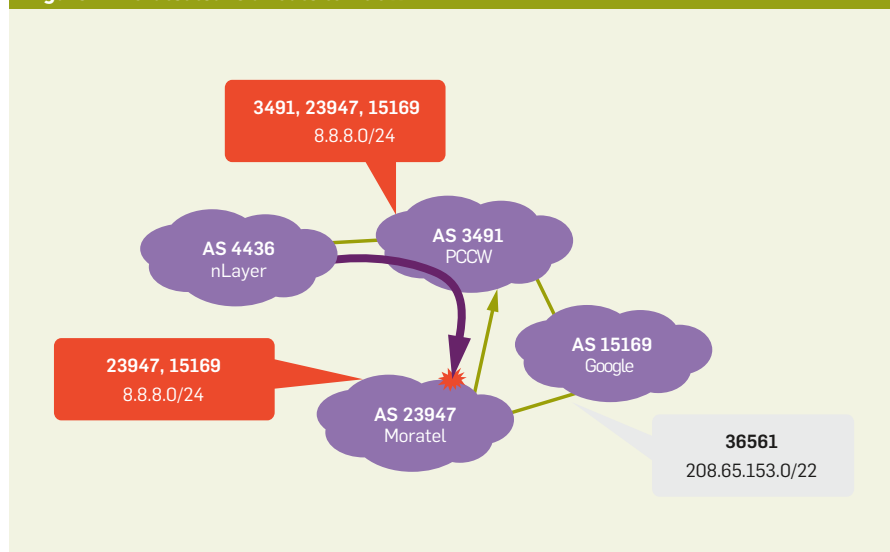


Figure 4. Moratel leaks a route to PCCW.^{32,33}





leaving Pakistan Telecom without a working route to YouTube and forcing it to drop traffic for YouTube's prefix.

Interception. Traffic interception occurs when the perpetrator AS intercepts traffic for the victim IP prefix and then silently passes it on to the legitimate origin AS. Interception is invisible to end users. Both route leaks and hijacks can lead to traffic interception, as long as the perpetrator has a working route to the legitimate origin AS and enough network capacity to transit the extra traffic it attracts. The 2010 China Telecom hijack is one example. Figure 2 shows how one of China Telecom's routers announced the bogus hijacked routes to its neighbors, while other China Telecom routers maintained a working route to the legitimate origin of the prefix.^{2,17} Traffic then traveled from the hijacking router, through China Telecom's high-capacity network, back out onto the wider Internet, and finally to the legitimate origin AS for the victim IP prefix. Similar incidents were observed last year by Renesys, which reported several short-lived hijacks that caused traffic for targeted IP prefixes to be intercepted by ASes based in Iceland and Belarus.³⁴

Defenses

Many of these incidents can be eliminated through security solutions based on simple cryptography or whitelisting techniques. This section looks at these solutions—prefix filtering, RPKI (Resource Public Key Infrastructure), and BGPSEC—and highlights the challenges involved in deploying them on

the global Internet.

Prefix filtering is a whitelisting technique used to filter out bogus BGP announcements. It is based on the rule of thumb of the first section, which implies that an AS (for example, Pakistan Telecom in Figure 3) will announce BGP routes to its provider (PCCW) only if those routes are: for its own allocated prefixes; or through its own customers (Aga Khan University, Lahore Stock Exchange, among others). As such, the provider can usually enumerate the small set of IP prefixes that are announced by its customer; that is, the set of IP prefixes allocated to Pakistan Telecom and its customer Pakistani ASes. The provider can therefore keep a *prefix list* of these IP prefixes for each customer and discard BGP announcements from a customer when they are not for prefixes on the list.

Benefit: *Prefix filtering is simple and effective.* Because a prefix filter is a simple whitelist, it does not usually present a large computational burden to routers. Prefix filtering has been used by various ISPs since the late 1990s and is a highly effective defense against hijacks and leaks perpetrated by customer ASes. Indeed, our research shows if every Internet provider with at least 25 customer ASes were to deploy prefix filters properly, this would prevent at least 48% of the Internet's ASes from launching routing leaks or hijacks.¹² Moreover, if PCCW had properly configured prefix filters in April 2008, the Pakistan Telecom's hijack of YouTube might never have happened. The same is true of the November 2012 Moratel route leak.

Challenge: *Prefix filtering works only on customer links.* Prefix filters, however, typically filter BGP announcements only from *customer* ASes; this is because prefix filters are built on the assumption that the filtered AS will announce only a *small* number of IP prefixes to the filtering AS. Prefix filtering is not typically used to filter BGP announcements from providers or settlement-free peers. For example, the 2010 China Telecom incident in Figure 2 could not have been prevented by prefix filtering, since China Telecom announced the bogus route along a settlement-free peering edge between China Telecom (AS 4134) and AT&T (AS 7018).

Challenge: *Lopsided incentives.* The incentives for deploying prefix filters are somewhat lopsided. For example, the “victims” of the 2008 Pakistan Telecom/YouTube incident were YouTube and all the impacted ASes that could not reach YouTube's hijacked prefix. However, the only AS that could have prevented the incident by using prefix filtering is PCCW itself; deploying prefix filters on the other victim ASes would do nothing to prevent the hijacked route from propagating through the Internet. Thus, the AS deploying the prefix filter (for example, PCCW) does not have particularly strong incentives to do so, other than protecting *the rest of the Internet* from attacks by its *own customers*.

RPKI: Cryptographic origin validation. The issues with prefix filtering have led to the development of many alternative security solutions. The approach that currently has the most traction is the RPKI.²⁶ Deployed since the start of this decade, the RPKI provides a trusted mapping from allocated IP prefixes to ASes authorized to originate them in BGP. To do this, the RPKI establishes a cryptographic hierarchy of *authorities* that allocate and suballocate IP address space, as well as authorize its use in BGP.

The RPKI is rooted at the RIRs (regional Internet registries). Figure 5 shows how ARIN (American Registry for Internet Numbers) allocates the prefix 8.0.0.0/8 to Level3, which suballocates prefix 8.8.8.0/24 to Google;⁵ these allocations are accomplished using cryptographic certificates. The holder of a cryptographic certificate for a prefix can then sign an ROA (route or-

igin authorization) authorizing a prefix (or its subprefix) to be originated in BGP; in Figure 5, for example, Google issues an ROA authorizing its AS 15169 to originate 8.8.8.0/24.

Benefit: Offline cryptography. RPKI does not require any modifications to BGP message formats; nor does it require any cryptography to be performed online during routing. Instead, each day an AS syncs its local cache to the public repositories that store RPKI objects, cryptographically verifies the RPKI objects in its local cache, and pushes the resulting whitelist (mapping IP prefixes and their authorized origin AS(es) to border routers in its AS.²⁶

Benefit: Protection from hijacks. Routers use this whitelist to filter hijacked BGP routes (that is, those with an unauthorized origin AS). For example, in Figure 2 AT&T can use the RPKI to determine the route

3356, 6167, 22394, 22394
66.174.161.0/24

is legitimate; AS 22394 is the origin of the route, and there is ROA in the RPKI of Figure 5 authorizing AS 22394 to originate 66.174.161.0/24. Meanwhile, the route originating at China Telecom's AS 22724 in Figure 2 is bogus, since there is no ROA authorizing AS 22724 to originate 66.174.161.0/24.

Benefit: Effective incentives. The RPKI also avoids the two problems that plague prefix filtering: it can be used to filter BGP announcements made by *any* neighbor (not just neighboring customers), and it avoids lopsided deployment incentives. During the first phase of RPKI deployment, an AS that wants to protect the routes it *originates* can populate RPKI repositories with ROAs for its originated routes. (Today, RPKI contains ROAs for about 4% of the routes announced in BGP.³¹) During the second phase of RPKI deployment, an AS can use RPKI to discard bogus routes, thus protecting the routes it selects. (Currently we are in the very early steps of this phase, with a few ASes worldwide are experimenting with the RPKI-based filtering.)

Challenge: RPKI takedowns and misconfigurations. A key challenge to RPKI deployment stems from abuse of RPKI itself.^{5,7,8,30} RPKI is designed as a threat model where BGP is under

attack but RPKI is trusted. Can RPKI itself be attacked, misconfigured, or lawfully compelled to misclassify a legitimate BGP route as bogus? (DNS is subject to lawful orders to take down domains;^{14,35} could RPKI be used to take down IP prefixes? This has already come up in several court cases.^{13,22,29}) Since routers use RPKI to filter bogus BGP routes, then the routers will lose access to the misclassified route. This means RPKI creates a new attack vector that can be used to blackhole routes. These issues are known to the RPKI standards community, and there are ongoing efforts to harden RPKI against this type of abuse through the development of configuration tools^{24,31,36} and fail-safe mechanisms;^{16,23} it is too early to tell what the outcome of these efforts will be.

Challenge: RPKI can be circumvented. Unfortunately, the RPKI cannot prevent some classes of attacks.

The first is a route leak. The RPKI is designed to detect routes with an unauthorized origin AS, but in a route leak, the perpetrator leaks a legitimate route with an authorized origin AS. For example, even if nLayer (AS 4436) in Figure 4 had been filtering routes based on the RPKI, it would still select the “leaked” Moratel route, since Google is a legitimate origin for prefix 8.8.8.0/24.

The second is a *path-shortening attack* in which an attacker announces a short bogus path to a prefix that terminates at the authorized origin AS.

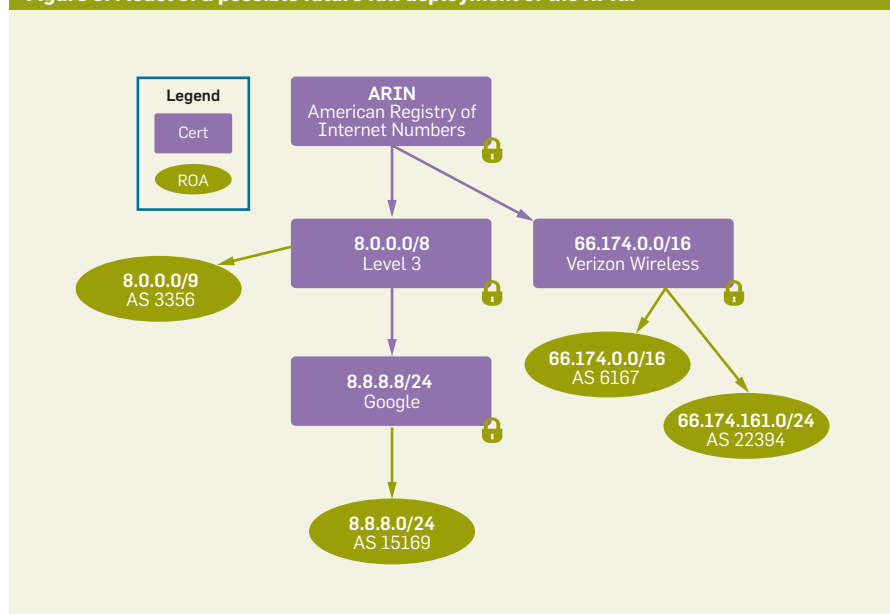
For example, even if RPKI were fully deployed, China Telecom (AS 4134) could still intercept traffic if it announced the route

4134, 22394
66.174.160.0/24

to AT&T in Figure 2. To see why, notice the route has a legitimate origin AS (AS 22394), but the route is actually bogus: there is no edge between AS 4134 and AS 22394. Thus, even if AT&T used RPKI to filter routes, it would still select the bogus route to China Telecom because it has a legitimate origin AS and is shorter than the legitimate route via Level3.

Fortunately, however, research^{1,12,27} suggests fewer ASes are likely to select a leaked or shortened route than one that is subprefix hijacked. During a subprefix hijack, the hijacker exploits longest-prefix-match routing to (potentially) convince *all* of the ASes on the Internet to select the bogus route. Meanwhile, both route leaks and path-shortening attacks do not exploit longest-prefix-match routing. Instead, they cause traffic to split between legitimate routes and the leaked/shortened route, with a majority of the traffic taking legitimate routes;^{12,27} the nature of the split is determined by routing policies and the AS-level topology (since ASes closer to the attacker are more likely to select the attacker's route).

Figure 5. Model of a possible future full deployment of the RPKI.⁵



BGPSEC: Cryptographic path validation. The community has considered a number of solutions that can eliminate the attacks that can be launched against the RPKI. Excellent surveys of these solutions are available.^{3,21} Here, we focus on BGPSEC, the protocol currently being standardized by the Internet Engineering Task Force (IETF).²⁵ Building on the RPKI's guarantees that a BGP route has an authorized origin AS, BGPSEC also provides *path validation*.

BGPSEC builds on the RPKI by adding cryptographic signatures to BGP messages. It requires each AS to sign each of its BGP messages digitally. The signature on a BGPSEC message covers (1) the prefix and AS-level path; (2) the AS number of the AS *receiving* the BGPSEC message; and includes (3) all the signed messages received from the previous ASes on the path. For example, in Figure 2, AT&T's AS 7018 would receive the following BGPSEC message from Level3's AS 3356:

```
[66.174.161.0/24 : 7018; 3356; 6167;
22394]3356
[66.174.161.0/24 : 3356; 6167; 22394]6167
[66.174.161.0/24 : 6167; 22394]22394
```

where the notation $[m]_A$ means message m signed by AS A . Upon receipt of a BGPSEC announcement, an AS validates the signatures and filters the route if the signatures are invalid.

Benefit: No path-shortening attacks. BGPSEC eliminates path-shortening attacks. In Figure 2, China Telecom (AS 4134) announced the path

```
4134, 22394
66.174.161.0/24
```

to AT&T. With BGPSEC, this attack would fail. China Telecom (AS 4134) would not receive the BGPSEC announcement

```
[66.174.161.0/24 : 4134; 22394]22394
```

from Verizon Wireless (AS 22394), since AS 22394 and AS 4134 are not neighbors, and thus could not form a 'shortened' bogus path that passes the digital signature checks required by BGPSEC.

Challenge: Online cryptography. Unlike the solutions discussed thus far, BGPSEC is an *online* cryptographic

RPKI does not require any modifications to BGP message formats; nor does it require any cryptography to be performed online during routing.

protocol; routers must cryptographically sign and verify every BGP message they send. This high computational overhead, which could require routers to be upgraded with crypto hardware accelerators, could slow down BGPSEC deployment.

Challenge: The transition to BGPSEC. All the security solutions considered here face the challenge that each AS will decide whether or not to deploy them based on their own local business objectives. This challenge is particularly acute with BGPSEC, because an AS cannot validate the correctness of an AS-level path (and therefore filter bogus routes) unless all the ASes on the path have applied their signatures to the message. This means the security benefits of BGPSEC apply only after *every* AS on the path has deployed BGPSEC. This is in stark contrast to the other two solutions discussed here—prefix filtering and RPKI—where only the AS doing the filtering needs to deploy the security solution. This creates a chicken-and-egg problem; the security benefits of BGPSEC apply only after a large number of ASes have deployed BGPSEC, but there is little security incentive for anyone to be the first to deploy BGPSEC.

There are a number of ways around this chicken-and-egg problem. One idea is that a set of early-adopter ASes would deploy BGPSEC (for example, for regulatory compliance, because of subsidies, or for public-relations purposes) and then trigger a cascade of BGPSEC deployment.^{4,10} One argument in favor of deploying BGPSEC is that because BGPSEC necessarily influences routes selection, an AS that has deployed BGPSEC could attract more revenue-generating traffic from its customers that prefer to select BGPSEC-secured routes. Our simulation results suggest these economic incentives, along with several other conditions, can create a cascade that leads to BGPSEC adoption at a majority of ASes on the Internet.¹⁰

Beyond economic incentives, however, is the question of what security benefits are provided during the transition to BGPSEC, when some ASes have adopted it but others have not. The answer is, unfortunately, less positive. Given the routing policies

that are likely to be most popular¹¹ during the transition to BGPSEC, our recent work argues that BGPSEC can provide only meager improvements to security over what is already possible with the RPKI.²⁷ This is because ASes may prioritize economic considerations over security concerns. For example, given a choice between an *expensive*, BGPSEC-secured route through a provider and a *cheap, insecure* BGP route through a customer, an AS might choose the cheap, insecure path. Thus, even ASes that have deployed BGPSEC can suffer from *protocol downgrade attacks*, where an attacker convinces them to select a bogus path instead of a legitimate BGPSEC-secured path.

Conclusion

Today we live in an imperfect world where routing-security incidents can still slip past deployed security defenses, and no single routing-security solution is a panacea against routing attacks. Research suggests, however, the combination of RPKI with prefix filtering could significantly improve routing security; both solutions are based on whitelisting techniques and can reduce the number of ASes that are impacted by prefix hijacks, route leaks, and path-shortening attacks. There are still several deployment challenges to overcome, since prefix filtering is limited by lopsided deployment incentives, while RPKI introduces a new dependence on centralized authorities.

This article has concentrated on protocol-based attacks on BGP. Recent research^{38,39} and media revelations^{15,18,40} indicate routers themselves could be compromised in a manner that circumvents *protocol-based* defenses such as prefix filtering, RPKI, and BGPSEC. Thus, while we continue to make progress toward protocol-based defenses for routing security, the next frontier of routing security could very well be hardening the software and hardware used in Internet routers.

Acknowledgments

Thanks to my collaborators on the research I have drawn upon here: Kyle Brogle, Danny Cooper, Phillipa Gill, Shai Halevi, Ethan Heilman, Pete Hummon, Alison Kendlar, Robert Lychev,

Anchal Malhotra, Leonid Reyzin, Jennifer Rexford, Michael Schapira, and Tony Tauber. This work has been funded by the NSF (1017907), Cisco, and the Sloan Foundation. □

Related articles on queue.acm.org

What DNS is Not

Paul Vixie

<http://queue.acm.org/detail.cfm?id=1647302>

The Network is Reliable

Peter Bailis and Kyle Kingsbury

<http://queue.acm.org/detail.cfm?id=2655736>

Splinternet Behind the Great Firewall of China

Daniel Anderson

<http://queue.acm.org/detail.cfm?id=2405036>

References

- Ballani, H., Francis, P. and Zhang, X. A study of prefix hijacking and interception in the Internet. In *Proceedings of the ACM SIGCOMM 2007 Conference*, 265–276.
- Brown, M. Pakistan hijacks YouTube. Renesys blog; http://www.renesys.com/blog/2008/02/pakistan_hijacks_youtube_1.shtml.
- Butler, K., Farley, T., McDaniel, P. and Rexford, J. A survey of BGP security issues and solutions. In *Proceedings of the IEEE 98*, 1, (2010), 100–122.
- Chan, H., Dash, D., Perrig, A. and Zhang, H. Modeling adoptability of secure BGP protocol. In *Proceedings of the ACM 2006 SIGCOMM Conference*, 279–290.
- Cooper, D., Heilman, E., Brogle, K., Reyzin, L. and Goldberg, S. On the risk of misbehaving RPKI authorities. In *Proceedings of the 12th ACM Workshop on Hot Topics in Networks* (2013).
- Cowie, J. China's 18-minute mystery. Renesys blog, 2010; <http://www.renesys.com/blog/2010/11/chinas-18-minute-mystery.shtml>.
- FCC Communications Security, Reliability and Interoperability Council III (CSRIC). Secure BGP deployment. *Communications and Strategies*; (2012); http://transition.fcc.gov/bureaus/pshs/advisory/csr/c3/CSRICIII_9-12-12_WG6-Final-Report.pdf.
- FCC Communications Security, Reliability and Interoperability Council, Working Group 6. Secure BGP deployment, final report, 2013.
- Gao, L., Rexford, J. Stable Internet routing without global coordination. *IEEE/ACM Transactions on Networking* 9, 6 (2001), 681–692.
- Gill, P., Schapira, M. and Goldberg, S. Let the market drive deployment: A strategy for transitioning to BGP security. In *Proceedings of the ACM SIGCOMM 2011 Conference*, 14–25.
- Gill, P., Schapira, M. and Goldberg, S. A survey of interdomain routing policies. *ACM SIGCOMM Computer Communication Review* 44, 1 (2013), 28–34.
- Goldberg, S., Schapira, M., Hummon, P. and Rexford, J. How secure are secure interdomain routing protocols? In *Proceedings of the ACM SIGCOMM 2010 Conference*, 87–98.
- Goldman, E. Sex.com—An update. Technology and Marketing Law blog, 2010; http://blog.ericgoldman.org/archives/2006/10/sexcom_an_update.htm.
- Government Printing Office. H.R.3261 - Stop Online Piracy Act, 2011.
- Greenwald, G. How the NSA tampers with US-made Internet routers. *The Guardian* (May 12, 2014); <http://www.theguardian.com/books/2014/may/12/glenn-greenwald-nsa-tampers-us-internet-routers-snowden>.
- Heilman, E., Cooper, D., Reyzin, L. and Goldberg, S. From the consent of the routed: Improving the transparency of the RPKI In *Proceedings of the ACM SIGCOMM 2014 Conference*.
- Hiran, R., Carlsson, N. and Gill, P. 2013. Characterizing large-scale routing anomalies: a case study of the China Telecom incident. In *Passive and Active Measurement*. Springer, Berlin Heidelberg, 2013, 229–238.
- Horchert, J., Appelbaum, J. and Stöcker, C. 2013. Shopping for spy gear: Catalog advertises NSA toolbox. *Der Spiegel* (Dec. 29, 2013); <http://www.spiegel.de/international/world/catalog-reveals-nsa-has-backdoors-for-numerous-devices-a-940994.html>.
- Huston, G. Interconnection, peering and settlements, Part I. *Internet Protocol Journal* 2, 1 (1999). Cisco.
- Huston, G. Interconnection, peering and settlements, Part II. *Internet Protocol Journal* 2, 2 (1999). Cisco.
- Huston, G., Rossi, M. and Armitage, G. Securing BGP: a literature survey. *IEEE Communications Surveys and Tutorials* 13, 2 (2011), 199–222.
- Internet Governance Project. M.L. Mueller. In important case, RIPE-NCC seeks legal clarity on how it responds to foreign court orders; <http://www.internetgovernance.org/2011/11/23/in-important-case-ripe-ncc-seeks-legal-clarity-on-how-it-responds-to-foreign-court-orders/>.
- Kent, S. and Mandelberg, D. Suspenders: a fail-safe mechanism for the RPKI. Internet Engineering Task Force, 2014; <http://tools.ietf.org/html/draft-kent-sidr-suspenders-01>.
- LACNIC Labs. RPKI looking glass; www.labs.lacnic.net/rpkitools/looking_glass/.
- Lepinski, M., ed. BGPSEC protocol specification. IETF Network Working Group, 2014; <http://tools.ietf.org/html/draft-ietf-sidr-bgpsec-protocol-05>.
- Lepinski, M. and Kent, S. RFC 6480: an infrastructure to support secure Internet routing. Internet Engineering Task Force, 2012; <http://tools.ietf.org/html/rfc6480>.
- Lychev, R., Goldberg, S. and Schapira, M. BGP security in partial deployment. Is the juice worth the squeeze? In *Proceedings of the ACM SIGCOMM 2013 Conference*, 171–182.
- McPherson, D., Amante, S., Osterweil, E. and Mitchell, D. eds. Draft. Route leaks and MITM attacks against BGPSEC. IETF Network Working Group, 2013; <http://tools.ietf.org/html/draft-ietf-grow-simple-leak-attack-bgpsec-no-help-03>.
- Miller, R. Court ruling: Israeli and US terrorism victims now "own" Iran's Internet. Joshuapundit blog (June 25, 2014); <http://joshuapundit.blogspot.com/2014/06/court-ruling-israeli-and-us-terrorism.html>.
- Mueller, M. and Kuerbis, B. Negotiating a new governance hierarchy: an analysis of the conflicting incentives to secure Internet routing. *Communications and Strategies* 81 (2011), 125–142.
- National Institute of Standards and Technology. RPKI deployment monitor; <http://www.x.antd.nist.gov/rpki-monitor/>.
- Paseka, T. Why Google went offline today and a bit about how the Internet works. Cloudflare blog (Nov. 6, 2012); <http://blog.cloudflare.com/why-google-went-offline-today-and-a-bit-about>.
- PeeringDB. 2014; <https://www.peeringdb.com/>.
- Peterson, A. Researchers say U.S. Internet traffic was re-routed through Belarus. That's a problem. *Washington Post* (Nov. 20, 2013).
- Piscitello, D. Guidance for preparing domain name orders, seizures and takedowns. Thought paper. ICANN (Mar. 2012).
- RIPE Network Coordination Centre. RPKI validator; <http://localcert.ripe.net:8088/trust-anchors>.
- RIPE Network Coordination Centre. YouTube hijacking: A RIPE NCC RIS case study. RIPE NCC Blog, 2008; <http://www.ripe.net/internet-coordination/news/industry-developments/youtube-hijacking-a-ripe-ncc-ris-case-study>.
- Schuchard, M., Thompson, C., Hopper, N. and Kim, Y. Taking routers off their meds: why assumptions of router stability are dangerous. In *Proceedings of the Network and Distributed System Security Symposium*, 2012.
- Schuchard, M., Thompson, C., Hopper, N. and Kim, Y. 2013. Peer pressure: exerting malicious influence on routers at a distance. In *IEEE 33rd International Conference on Distributed Computing Systems*, 2013, 571–580.
- Storm, D. 17 exploits the NSA uses to hack PCs, routers and servers for surveillance. *ComputerWorld* (Jan. 3, 2014); <http://blogs.computerworld.com/cybercrime-and-hacking/23347/17-exploits-nsa-uses-hack-pcs-routers-and-servers-surveillance>.
- Wang, L., Park, J., Oliveira, R. and Zhang, B. Internet AS-level topology archive; <http://irl.cs.ucla.edu/topology/>.

Sharon Goldberg is an assistant professor of computer science at Boston University.

Copyright held by owners/author(s). Publication rights licensed to ACM. \$15.00.

DOI:10.1145/2629572

Use this map query interface to search the world, even when not sure what information you seek.

BY HANAN SAMET, JAGAN SANKARANARAYANAN, MICHAEL D. LIEBERMAN, MARCO D. ADELFIGIO, BRENDAN C. FRUIN, JACK M. LOTKOWSKI, DANIELE PANOZZO, JON SPERLING, AND BENJAMIN E. TEITLER

Reading News with Maps by Exploiting Spatial Synonyms

DO YOU TRAVEL? Do you want to know what is happening in the place and vicinity you are traveling to? Do you want to keep up with the latest news in the place and neighboring vicinity you left, especially if it is where you may have once lived or worked? If you answered yes to any of these questions, then our NewsStand, denoting Spatio-Textual Aggregation of News and Display, and related systems, are for you.

NewsStand⁴⁶ is an example application of a general framework for enabling people to search for information with a map-query interface. As such, it is a variant of systems we have been developing for the

past 30 years at the University of Maryland that we call “spatial browsers,” as in Samet et al.³⁹ and Samet et al.⁴¹ The advantage of the map-query interface is that a map, coupled with the ability to vary the zoom level at which it is viewed, provides inherent granularity to a search process that facilitates approximate search. This capability distinguishes it from prevalent keyword-based conventional search methods that provide a limited facility for approximate searches that are realized primarily by permitting a match through a subset of the keywords. However, users often lack a firm grasp of which keyword to use, and would thus welcome the search to also account for synonyms. For queries to spatially referenced data, termed “spatial queries to spatial data,” the map-query interface is a step in this direction. Consider the action of pointing at a location (such as through the appropriate positioning of a pointing device or gesturing appropriately) and making the interpretation of the precision of this positioning specification dependent on the zoom level. This is equivalent to permitting use of spatial synonyms.

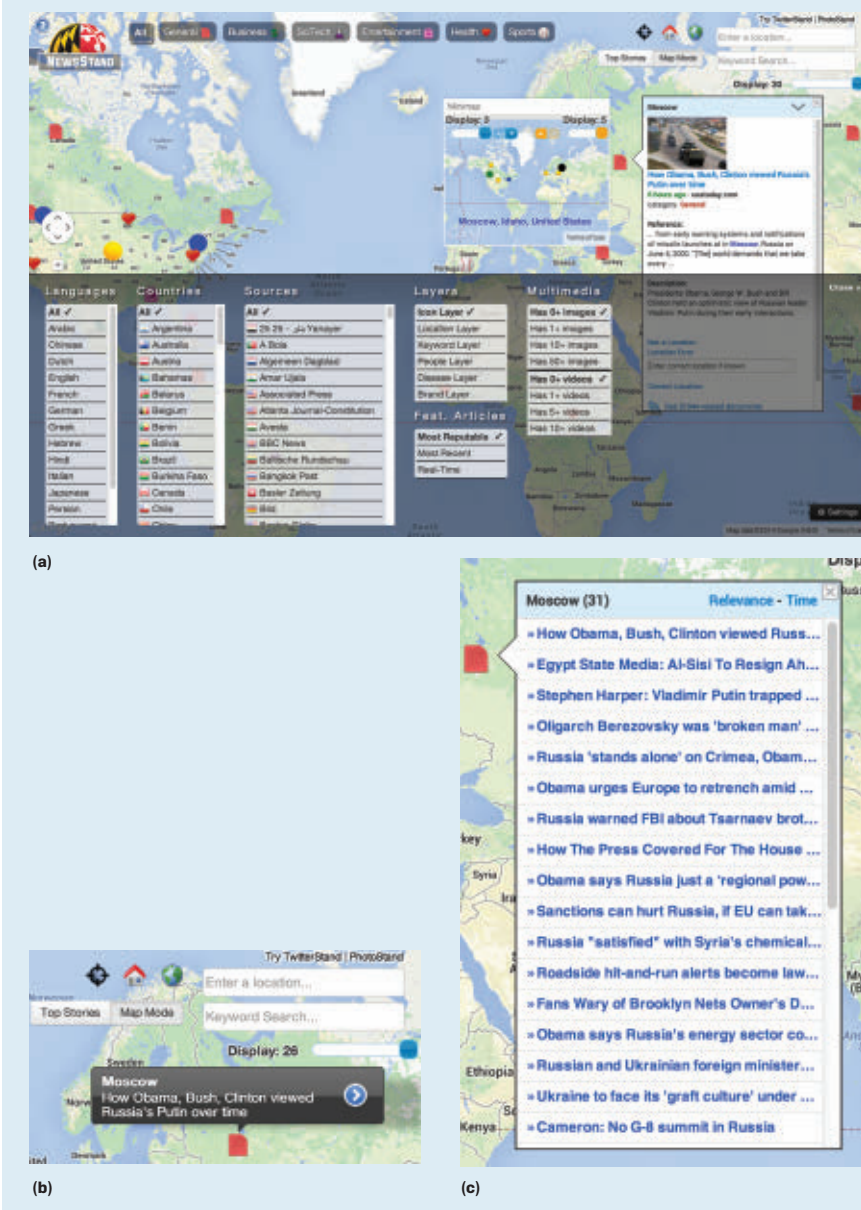
Being able to use spatial synonyms is important, as it enables users to search for data when they are not exactly sure what they seek or what the answer to their query should be. For example, suppose the query seeks a “rock concert in Manhattan.” The presence of “rock concerts” in Harlem, Brooklyn,

» key insights

- **The NewsStand map query interface monitors the output of more than 10,000 RSS news sources within minutes of publication and associates articles with the locations they mention.**
- **A map coupled with the ability to vary the zoom level at which it is viewed and interpreted provides inherent granularity to the search process, facilitating an approximate search and enabling use of spatial synonyms.**
- **Textual specification of location is preferable to geometric specification for users of mobile devices but must overcome potential ambiguity.**



Figure 1. NewsStand Map Mode: (a) Example screenshot for “What is happening at location X on March 26, 2014?”; (b) representative headline in Moscow for the Obama/Putin relationship topic; and (c) representative headlines for topics associated with Moscow.



or New York City are all good answers when no such events can be found in Manhattan, as they correspond to spatial synonyms: Harlem by virtue of being contained in Manhattan; Brooklyn by virtue of both proximity and being a sibling (both are boroughs of New York City); and New York City by virtue of a containment relationship. Conventional search engines handle spatial queries by dynamically incorporating information gleaned from query-and-click logs, whereby if enough users searching for Manhattan end up clicking on pages associated with Harlem or New York, then over time, the search

engine infers the spatial scope of the documents to be proximate or relevant to New York. More recently, search engines (such as Google’s Knowledge Graph and Microsoft’s Satori) have been using large knowledgebases to understand the spatial focus of keyword search queries, as well as, to a limited extent, the spatial focus of the documents. Notwithstanding such improvements to search engines for understanding locations in documents, the primary utility of the search engines is still based on popularity in the sense that the PageRank algorithm and click logs ensure webpages provided

to the user are ordered by measures that incorporate some aspect of their frequency. In particular, the classic PageRank algorithm uses static data, while click logs correspond to dynamic data. The frequency basis ensures the results are the same as those provided to other users. This property can be characterized as the “democratization of search” in the sense that all users receive equal treatment. A cruder way to describe the resulting effect is that it does not discriminate among users in the sense they all get the same bad (or good) answers. That is, the effect of using the PageRank algorithm and click logs to order results (effectively choosing which results to present to the user) is that if nobody ever looked for some data (or its neighbor in a spatial sense) before or linked to it, then it will never be found and, hence, will never be presented to the user. In some cases, this is fine. However, for synonyms, it has a strongly negative effect on the quality of search results, as it means if nobody linked to similar pages due to their content being equivalent but for the use of the same words, or clicked on a spatial neighbor, then the search engine will never find the similarity. As such, the PageRank algorithm will never be able to find similar pages as it crawls the Web when building an index to the Web pages, and no useful click logs will be found.

NewsStand and related systems we have built at the University of Maryland address the synonym problem for spatial queries. Note that all spatial queries can be broken down into two classes:

Location-based. Takes a location X , traditionally specified using lat/long coordinate values as an argument, and returns a set of features associated with X ; and

Feature-based. Takes a feature Y as an argument and returns the set of locations with which Y is associated.

These queries can also be characterized as a pair of functions, with one the inverse of the other. Feature-based queries are also known as “spatial data mining.”³ Although features are usually properties (also known as attributes) of spatially referenced data (such as crop types, soil types, zones, and speed limits), they and the underlying spatially referenced data domain can be

more broadly interpreted. NewsStand adapts them to the domain of unstructured data consisting of collections of news articles with textually specified locations; the features are the topics. Adapting these concepts results in a location-based query returning all topics and articles mentioning a specific place or region X and a feature-based query returning all places and regions mentioned in articles about topic T or just article Y . Note that NewsStand does not require users to specify T in advance, in which case the topics are ranked by importance, which can be defined by various criteria, including, but not limited to, the number of articles comprising them. Here is a typical pair of queries: What is happening at location X ?; and Where is topic T or article Y happening?

Their execution is facilitated by building an index on the spatial data,³⁶ preferably all at once through bulk loading, as in Hjaltason and Samet.¹² An index is relatively easy to construct when the spatial data is specified geometrically and numerically. However, data is not specified this way in NewsStand, as all data is unstructured. In particular, location and feature data are both just collections of words, some of which, in the case of spatial data, can be (but are not required to be) interpreted as the names of locations. That is, spatial data is specified using

text (called “toponyms”) rather than geometry, meaning some ambiguity is involved. This ambiguity has advantages and disadvantages. The advantage is that, from a geometric standpoint, the textual specification captures both the point and spatial extent interpretations of the data, analogous to a polymorphic type in parameter transmission serving as the cornerstone of inheritance in object-oriented programming languages. For example, a city can be geometrically specified by either a point (such as its centroid) or a region corresponding to its boundary, the choice of which depends on the level of zoom with which the query interface is activated. The disadvantage is we are not always sure if a term is a geographic location. For example, does “Jordan” refer to a country, a river, or a surname, as in “Michael Jordan”? The process of answering is called “toponym recognition.”¹⁸ Moreover, if it is a geographic location, then which, if any, of the possibly many instances of geographic locations with the same name is meant. For example, does “London” refer to an instance in the U.K., Ontario, Canada, or one of many others? The process of answering is called “toponym resolution.”¹⁹ Resolving these ambiguities with no errors (or almost none) is one of the main technical challenges we have faced in deploying NewsStand and related systems.

NewsStand User Interface

NewsStand’s goal is to offer an alternative to the news-reading process and, more important, experience. Users query NewsStand by choosing a region of interest and finding relevant associated topics and articles (experience the NewsStand interface at <http://news-stand.umiacs.umd.edu>). The topics and articles displayed are determined by the location and level of zoom that together dictate the spatial scope of the query, or region of interest. The two ways of interpreting the notion of “region of interest” are in terms of content and of news sources. In the simplest way, there are no predetermined boundaries on the locations of the news sources for the articles being displayed for the region of interest. In the second way, the sources can be limited to a subset of available sources by specifying them explicitly (such as *New York Times* and *Washington Post*), by language, by spatial region that can be specified textually (such as restrict sources to Ireland), or by drawing the region of interest on the NewsStand map (such as a box overlapping Ireland and the U.K.). Users can also constrain the spatial region and news sources; they need not be the same. This is a useful feature, as it enables users to see how one part of the world views events in another part of the world. For example, users may want to see how the Eng-

Figure 2. NewsStand Top Stories Mode: (a) Example screenshot for “Where is topic T or article Y happening on March 26, 2014?”; and (b) subset of images associated with vthe Obama/Putin relationship topic with duplicates and near-duplicates grayed over.



lish press views and interprets developments in the Middle East. The result is analogous to sentiment analysis. Other applications include monitoring hot spots for investors, national security, and keeping up with the spread of diseases, as in Lieberman et al.²⁴

Figure 1a is a screenshot of NewsStand’s output for “What is happening at location X on March 26, 2014?” This is NewsStand’s “Map Mode.” X is Africa, Europe, and part of the Americas. The figure includes an excerpt from an article about the Obama/Putin relationship that mentions Moscow. Each icon, or symbol on the map, we call a “marker,” represents a set of articles on the same and/or different topics where the main property shared by all the articles is that they mention the corresponding map location. The type of symbol conveys information about the news category (such as general news, business, science and technology, entertainment, health, and sports) spanning most of the article topics associated with the location. The user can select one or more of these categories by toggling the appropriate buttons at the top of the screen.

Figure 1b is an info bubble containing the headline from a representative article on the dominant topic associated with Moscow, or the Obama/Putin relationship. NewsStand obtains these topics by applying a clustering process to all the articles. The info bubble is generated by the user hovering the mouse cursor over Moscow. The hovering action also causes the markers at all other locations on the map associated with this representative article to be replaced by orange balls. In this example, these locations correspond to, in part, the countries involved in, or affected by, the Obama/Putin relationship. Some locations might lie outside the geographic span of the map (such as in North America and the Far East) currently visible in the screenshot.

Including areas of interest beyond the map is achieved through a minimap generated when the user hovers over a marker, along with the headline (not shown here). The action displays the geographic span of the representative article with orange balls at the appropriate locations. The utility of the minimap involves permitting users to see the selected article’s geographic fo-



NewsStand gathers its data by crawling the Web. Its primary sources are thousands of individual news sources worldwide in the form of RSS feeds.



cus, without having to leave their area of interest on the main map, and is independent of the current level of zoom.

Blue balls on both the main map and the minimap indicate other locations with the same name as the one over which the user is currently hovering—here Moscow. Allowing the minimap to include all other locations with the same name may cause the geographic span of the minimap to exceed that of the orange balls. The blue balls enable detecting toponym resolution errors.

A black ball on the minimap marks the location over which the user is currently hovering, or Moscow. Up and down arrows on the minimap allow the user to scroll through the orange and blue balls and output the corresponding location names. Scrolling through the blue balls enables ranking the interpretations of the location name. Green and red balls on the minimap correspond to the current blue and orange balls in the scrolling process. Hovering over an orange ball in the minimap yields the name of the location, while hovering over a blue ball yields both the name of the location and its containing location on the minimap (such as “Moscow, ID, United States”), as all blue balls have the same name.

Figure 1c is an info bubble showing headlines of representative articles for each topic associated with Moscow, the location over which the user hovered most recently. It results from clicking the > symbol in the headline info bubble associated with this location. Clicking on one of the headlines yields the summary info bubble (see Figure 1a), as well as the adjacent corresponding minimap, which is also generated when hovering over a marker. Note that orange balls (but not the blue balls) in the minimap differ as a user scrolls through the headlines of the topics. The summary info bubble also contains links to related images, videos, and other articles. Clicking on the headline in this summary info bubble causes the full text of the article to be displayed and, if it is in a language other than English, an option is available to translate it and/or the headline into English through a translation package (such as Google Translate and Microsoft Translator).

The domain of news sources for the articles from which the representative article is drawn can be restricted by language, geographic region, or country, as well as by specific newspaper. This is done by setting up an appropriate filter using the “settings” button (at the lower-right-hand corner of the screen in Figure 1a) and selecting the appropriate ones, as in the lower grayed half of Figure 1a. Note that users are also able to do a search by location or keyword(s), as well as vary the number of markers to be displayed through a display slider.

Figure 2a is a screenshot of NewsStand’s output for “Where is topic T or article Y happening on March 26, 2014? This is NewsStand’s “Top Stories Mode.” T is one of the topics whose representative headlines are shown in the bottom-left pane ranked using an importance measure. Importance is defined in terms of significance, age, and frequency, though velocity/acceleration of arrival should also be taken into account, as it is a better measure since topics eventually lose their timeliness. The headline displayed is the one that was clicked. It is highlighted (by being grayed) as a result of the user hovering over it, corresponding here to the Obama/Putin relationship topic. Clicking on the headline causes more details (such as an expanded description and the number of related documents, images, and videos) to appear about it, as shown in the top left pane of Figure 2a, along with the means to access them via a subsequent mouse click.

The hovering click in the bottom-left pane of Figure 2a also causes appropriate markers (category symbols) to appear on the map (right pane) at the principal geographic locations associated with the topic. In this example, these locations correspond to, in part, some of the countries involved in, or affected by, the Obama/Putin relationship, including the U.S. and Russia. Hovering the mouse cursor on the map in the right pane causes info bubbles and the associated minimap with the same semantics to appear, as in the “What is happening at X ?” query in Figure 1. In particular, the orange balls enable the user to differentiate between locations in close proximity (such as London and Wimbledon in the U.K. for a tennis cluster), while the blue balls

capture other instances of geographic locations with the same name (such as “Moscow, PA, United States”).

Users in Map and Top Stories modes can obtain the collection of images and videos associated with each cluster. For images, NewsStand detects duplicates or near duplicates and hides them from view. This is a powerful property, as it uses the words associated with the articles, or their semantics, as the first step in finding similar images, while the duplicates among the similar images can be detected through classical image-similarity methods, including hierarchical color histograms⁵ and the Scale-Invariant Feature Transform algorithm, or SIFT.²⁵ Figure 2b is an example of a subset of such images for the Obama/Putin relationship topic anchored in Moscow.

As outlined earlier, NewsStand’s ultimate goal is to make the map the medium of choice for presenting information with spatial relevance and is thus not restricted to news articles; that is, it can also be applied to search results, images, videos, and tweets. It also enables summarization of news, further exploration, and even knowledge acquisition through discovery of patterns in the news, a direct result of the association of topics or categories with the locations mentioned in constituent articles. For example, queries can be

chained in the sense that an interesting topic might be associated with Paris, France, and the same topic might also be associated with London, U.K., as found through the orange balls. At this point, the user would move the pointing device to London and click to find other related topics mentioning London, as well as other locations to which the user can transition by moving via the map-query interface. This unlimited chaining is possible only in Map Mode, as the queries are location-based, while the queries in Top Stories Mode are topic-based, and the markers on the map are restricted to the locations corresponding to the highest-ranked topics, unless the user does a keyword search.

NewsStand can also compute a cluster disease focus, or the most common term in the cluster corresponding to the name of a disease (such as “Europe on March 26, 2014” in Figure 3). Alternatively, a user can apply the same idea and find the most common term in the cluster corresponding to the name of a person or brand. Finding such a term is achieved by setting the “layers” parameter to “disease,” “people,” or “brand,” respectively.

Related Work

Comparing NewsStand with existing newsreaders is difficult, as reading the

Figure 3. NewsStand screenshot showing clusters that mention a disease name for Europe on March 26, 2014; the user is hovering over Valencia, Spain, and the disease is breast cancer. Orange balls in the minimap show all other locations in the world where the relevant cluster mentions breast cancer.



news with a map is a feature not found in any popular news reader (such as Pulse). News-reading systems (such as Microsoft Bing News, Google News, and Yahoo News) present the news in classical linear fashion with aggregation of different sources for each topic. These providers all include some aspect of locality in terms of aggregation of articles and topics relevant to a user's locality. Aggregation is usually done according to a ZIP or postal code or city-state specification. For example, for ZIP code 20742, topics could mention "College Park, MD." For Google News, this feature seems to be implemented, at least as far as we can tell, by applying Google search with location names as search keys. For example, after determining the user is in ZIP code 20742 (such as by virtue of the user's IP address, absent an alternative specification of the local area), Google News would return the topics mentioning "College Park, MD" or "University of Maryland," as they are known to be associated with this ZIP code. In addition, the resulting list of topics also appears to be based primarily on the location of the news source (usually a newspaper) where the articles comprising the topics are contained, rather than on story content. In these examples, the number of topics displayed is limited, though there is no particular reason for this limitation save for the absence of topics relevant to the user's locality. Note also that in these examples there is no notion of article importance in determining what is shown to the user.

Interestingly, none of the popular news readers use a map to present the articles, though they could all do so with a mashup on their mapping platforms. HealthMap¹⁰ does use a map to present disease outbreaks, where locations are obtained from the dateline of a disease report or metadata from ProMed reports. This use of a map to present disease reports is similar to the "disease layer" in NewsStand (see Figure 3), except that in NewsStand the locations are obtained from the article's actual text. It is also similar to an implementation of our Spatio-Textual Extraction on the Web Aiding the Retrieval of Documents, or STEWARD,²³ system with ProMed reports that can also show disease propagation over time.¹⁶ Note although the mapping

platforms supporting the mashups are able to zoom in, with the exception of NewsStand, none couple zoom with the ability to obtain more articles.

In the past, a number of systems tried to understand geographical locations in news articles and display them, though most are no longer available or accessible. For example, Reuters's NewsMap, the *Washington Post's* TimeSpace, the BBC's LiveStats, and the AP's Mobile News Network tried to associate news articles with a coarse geography based on the wire-service location where the article was filed. An article submitted to the Miami news wire would therefore be listed for all ZIP codes in Miami. Unlike NewsStand, there appears to be no attempt in the AP Mobile News Network to analyze individual articles to determine the main associated location, or geographic focus, or other important locations mentioned in the articles.

It is also useful to compare NewsStand with commercial services for Web search and recommender systems (such as review sites like Yelp and TripAdvisor). The difference is that in these systems, awareness of spatial entities is a result of the explicit population of their databases with spatial information in the form of addresses or GPS, or lat-long, values; hence they can support the exploration of the spatial information. NewsStand has a dual role: discover the spatial information in its input data that is specified textually and usually ambiguously (requiring incorporation of other information, some external to its input data); and exploratory, where the capabilities are similar to those in recommender systems, though there is less emphasis on a map-query interface in the recommender systems.

NewsStand Architecture

The key elements to understanding news were perhaps best captured in 1902 by Rudyard Kipling in his *Just So Stories*: "I keep six honest serving-men (They taught me all I knew); Their names are What and Where and When and How and Why and Who." NewsStand focuses on the "what" and "where" and to a lesser extent on "when," where "when" is recent. Here, we focus on "what" and later on "where."

NewsStand gathers its data by crawl-

ing the Web. Its primary sources are thousands of individual news sources worldwide in the form of really simple syndication (RSS) feeds; RSS is a widely used XML protocol for online publication, ideal for NewsStand, as it requires only a title, short description, and Web link for each published news item. RSS 2.0 also allows an optional publication date, helping determine the age, or "freshness," of an article. NewsStand currently indexes 10,000 news sources and processes approximately 50,000 news articles per day. It determines the geographic locations mentioned in the article, a process known as geotagging, and tries to determine an article's geographic focus or foci that are the key locations mentioned in it.

NewsStand also aggregates news articles by topic based on content similarity (termed "clustering") so articles concerning the same event are grouped in the same cluster. The main goal of clustering is to automatically group news articles that describe the same news event into sets of news articles, termed "article clusters" (also referred to earlier as "topics" and as "clusters"), such that each cluster contains only the articles encountered in the input seen so far pertaining to a specific topic. As news articles enter this stage, NewsStand assigns them to news clusters, essentially a one-shot process meaning once an article is added to a cluster it remains there forever. NewsStand will never revisit or recluster the article, which is desirable, as articles come into NewsStand at a high-throughput rate, and NewsStand needs a document-clustering system that can process them quickly while still managing to deliver good-quality clustering output. Such a version of the clustering algorithm is characterized as being "online."

Given these requirements, NewsStand uses the leader-follower clustering⁷ algorithm that permits online clustering in both the term-vector space using the term frequency-inverse document frequency, or TF-IDF, metric³⁵ and the temporal dimension. For each cluster, NewsStand maintains a term centroid and time centroid corresponding to the means of all term-feature vectors and publication times of articles in the cluster, respectively. To cluster a new article a ,

NewsStand checks whether a cluster exists where the distance from its term and time centroids to a is less than a fixed cutoff distance ϵ . If one or more candidate clusters exists, a is added to the closest such cluster, and the cluster's centroids are updated; otherwise, NewsStand creates a new cluster containing only a .

NewsStand's online clustering algorithm ranks the clusters based on its notion of "importance," as determined by several factors:

Number of articles. The number of articles in the cluster;

Number of unique news sources in a cluster. For example, an event in Irvine, CA, is important if carried by multiple news sources, especially if some are geographically distant from Los Angeles (approximately 50 miles from Irvine);

The cluster's rate of propagation. Articles about important events are picked up by multiple news sources within a short period of time; and

Time of addition. The time at which the most recent addition to the cluster took place, an option exercised by the NewsStand user, precluding consideration of the first three factors.

When clusters are ranked using the first three factors, NewsStand must choose the cluster's representative article, a form of secondary ranking. The nature of this article can be varied by the NewsStand user to be either the most recent article, thereby disregarding the corresponding cluster's importance (the fourth factor), or according

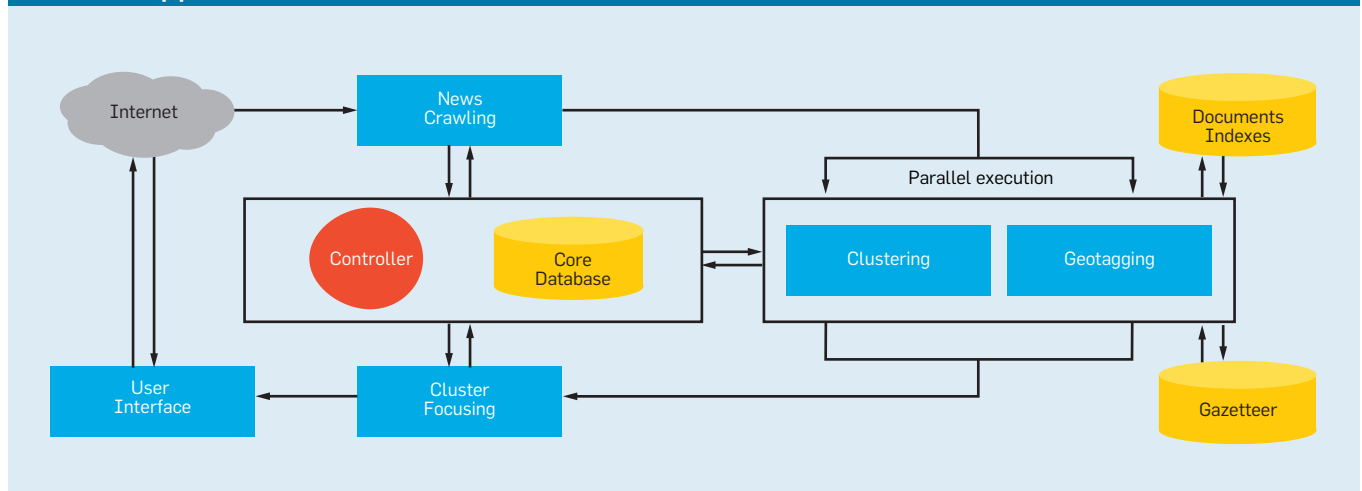
to the cluster's importance, where the choice is between the article from the most reputable source or from the source with the freshest article. Though it is important for NewsStand to show the clusters with the most significant topics in the current viewing window when in Map Mode, simply displaying the highest-ranked topics on the map may not produce a useful display for a wide audience, as these topics tend to be clustered in particular geographic areas. This situation reflects the uneven news coverage of major newspapers, as they tend to focus on these geographic areas. In NewsStand, topic selection is a trade-off between significance and spread. To achieve a balance, NewsStand subdivides the viewing window into a regular grid and requires each grid square contain no more than a maximum number of topics. The topics displayed are selected in decreasing order of significance and age, an approach that ensures a good spread of top topics across the entire map.

NewsStand also determines the geographic focus or foci associated with the cluster, a determination facilitated through the clustering process vis-à-vis the location feature. NewsStand displays each cluster at the positions of its geographic foci, provided it is one of the most important clusters or its geographic focus is also the focus of one of the most important clusters, where the number of locations is set by manipulating a slider in the upper-right corner of the map. The locations associated

with the most important clusters are thus the ones for which the map contains data. This display is usually done with the aid of symbols corresponding to their news category, as in Figure 1. However, rather than display the category symbol associated with the cluster, NewsStand can also display the text corresponding to the most prevalent term in the cluster we call the "keyword" by having the user set the appropriate "layers" parameter in Figure 1a. Alternatively, users can also display the actual name of the location that serves as the geographic focus by setting the layers parameter to "location."

Scalability and fast processing of individual articles were the most important criteria in designing NewsStand's architecture²⁰ (see Figure 4). Additional goals include presenting the latest news as quickly as possible, within minutes of its online publication, and being robust to failure. The NewsStand architecture fulfills these criteria by subdividing its collection and processing into several modules, each able to run independently on separate computing nodes in a distributed computing cluster. The figure outlines how the articles are processed by a sequence of these modules in a computing pipeline. Because each module might execute on a different node, a given article might be processed by several different computing nodes in the system. We also designed the modules in a way that allows for multiple instances of any module to run simultaneously on one or more nodes. News-

Figure 4. High-level overview of NewsStand's architecture. We designed the system as a pipeline, with individual processing modules working independently. A central control module orchestrates article processing by delegating work to the other modules and tracking articles in the pipeline.



Stand is thus able to execute as many instances of modules as required to handle the volume of news it receives. Each module receives input and sends output to a PostgreSQL database system that serves as a synchronization point. User actions (such as zoom, pan, and select) in the NewsStand interface are automatically converted into SQL queries that are answered by the PostgreSQL database.

Geotagging

NewsStand extracts geographic locations from news articles (termed “geotagging”) and is related to work in geographic information retrieval. Much of the existing work in this area deals with finding the geographic scope of websites and individual documents. In the context of news articles, NewsStand distinguishes among three types of geographic scope:²⁶

Provider. The publisher’s geographic location;

Content. The article or topic content’s geography; and

Serving. Based on the reader’s location.

NewsStand relies on article content to determine an article’s geographic scope and also tries to use provider scope, which it knows, and serving scope, which it attempts to learn.

NewsStand extends our earlier work on geotagging in STEWARD²³

to support spatio-textual queries on documents on the hidden Web. While STEWARD’s technology is applicable for an arbitrary set of documents, NewsStand contains additional modules and features designed specifically for more effective processing of news articles. STEWARD processes each document independent of all other documents, while NewsStand takes advantage of multiple versions and instances of articles about a topic by grouping them, most often from different news sources, into topic clusters that allow for improved geotagging and lets users retrieve related articles easily.

Geotagging consists of two processes: toponym recognition and toponym resolution. Toponym recognition involves geo/non-geo ambiguity, where a given phrase might refer to a geographic location or some other kind of entity (such as deciding whether a mention of “Washington” refers to a location or another entity, like a person’s name). Aliasing is a secondary issue, where multiple names refer to the same geographic location (such as “Los Angeles” and “LA”). Toponym resolution, also known as “geographic name ambiguity,” or polysemy, involves geo/geo ambiguity, where a given name could refer to any of several geographic locations. For example, “Springfield” is the name of many cities in the U.S., including in

Massachusetts and Illinois, where it is the state capital.

Toponym recognition. Many different approaches to toponym recognition have been undertaken, though all share certain characteristics. The idea is to extract the “interesting” phrases, or the ones most likely to be references to geographic locations and other entities, given the surrounding context. These phrases are collectively called the article’s “entity feature vector,” or EFV. The easiest way to identify the EFV is to look for phrases in the document that exist in a gazetteer or database of geographic names and locations. This approach is utilized by many researchers as their primary search strategy.² In particular, Web-a-Where,² a system for associating geography with Web pages, uses a small, well-curated gazetteer of approximately 40,000 locations, created by collecting the names of countries and cities with populations greater than 5,000. This size imposes a serious limitation on Web-a-Where’s practical geotagging capabilities, as it precludes it from being able to recognize the lightly populated, usually local, places commonplace in articles from local news sources. Moreover, a small gazetteer means Web-a-Where is more prone to making toponym-recognition errors because it misses out on being aware of geo/non-geo ambiguity afforded by the use of larger gazetteers.

To deal with the geo/non-geo ambiguity inherent in larger gazetteers, researchers, including Martins et al.,²⁷ Rauch et al.,³³ and Stokes et al.,⁴⁵ have proposed a variety of heuristics for filtering potentially erroneous toponyms. MetaCarta³³ recognizes spatial cue words (such as “city of”), as well as certain forms of postal addresses and textual representations of geographic coordinates. However, this strategy causes serious problems when geotagging newspaper articles, as the address of a newspaper’s home office is often included in each article. Given MetaCarta’s primary focus on larger prominent locations, these properly formatted address strings play too large a role in its geotagging process, resulting in many geotagging errors.

Other approaches to toponym recognition are rooted in solutions to related problems in natural language process-

Figure 5. Illustrative local lexicon for readers living in the vicinity of Columbus, OH; note the many local places that share names with more prominent locations elsewhere.



ing. For example, Named-Entity Recognition (NER)⁴⁷ focuses on nouns and noun phrases, aiming to identify noun phrases from an article that correspond to various entity classes (such as PERSON, ORGANIZATION, and LOCATION). Phrases tagged as LOCATION are most likely to be locations and stored as geographic features of the entity feature vector, while ORGANIZATION and PERSON phrases are stored as non-geographic features. NER approaches can be classified roughly as rule-based^{18,31} or statistical.¹⁷

Rule-based solutions feature catalogs of rules listing possible contexts in which toponyms may appear. On the other hand, statistical solutions rely on annotated corpora of documents to train language models using constructs like hidden Markov models (HMMs)⁴⁷ and conditional random fields (CRFs).¹⁵ HMMs and CRFs are used widely when annotated corpora are available. NewsStand's toponym-recognition procedure uses the NER tagger of the LingPipe toolkit⁴ that was trained on news data from the Message Understanding Conference, or MUC-6, and the well-known Brown corpus.⁹

Note that NER tagging does not preclude use of a gazetteer. Instead, these tagging methods serve as filters or pruning devices for controlling the number of lookups made to the gazetteer. The downside is that if an entity is not identified as a potential location, it will be missed, which happens. NewsStand uses GeoNames (<http://geonames.org/>), an open gazetteer originally assembled from more than 100 gazetteers, including the GEOnet Names Server and Geographic Names Information System. It is maintained by volunteers worldwide and currently contains the names of approximately 8.5 million different geographic locations, of which approximately 5.5 million are unique, with the difference accounting for the need to perform toponym resolution or resolve geo/geo ambiguity. The NewsStand gazetteer contains almost 16.3 million entries due to its need to keep track of the names of each location in multiple languages.

Our experience with the eight million articles most recently processed by NewsStand encountered only approximately 60,000 distinct locations,

though more than 40,000 were subject to geo/geo ambiguity, making toponym resolution critical. The gazetteer also stores the population of populated places or regions, as well as hierarchical information, including the country and administrative subdivisions containing the location, which is useful for recognizing highly local toponyms. Gazetteer lookup is applied to every geographic feature $f \in EFV$ and the matching locations to form the set $L(f)$, where there are as many sets as there are features, or $|EFV|$.

Toponym resolution. When a toponym is recognized, NewsStand applies a toponym-resolution procedure to resolve the geo/geo ambiguity. The problem of geo/geo ambiguity resolution is related to the more general problem of associating canonical entities with each noun phrase mentioned in a document, which is referred to as “named-entity disambiguation,” or NED. To disambiguate noun phrases, NED resorts to matching noun phrases to a knowledge repository (such as Wikipedia, DBpedia, and Yago). At a high level, noun phrases mentioned in a document are first matched to multiple candidate entities, then disambiguated based on the relatedness of these entities in the knowledge repository. For instance, Milne and Witten²⁹ used a supervised learning approach using a relatedness measure, where the relatedness between two Wikipedia articles is based on the number of overlapping incoming links. Similarly, Hoffart et al.¹³ used “coherence” among the various candidate entities to disambiguate all noun phrases. Some recent efforts have sought to combine NER and NED modules into a Named-Entity Recognition and Disambiguation, or NERD, module³⁴ that scans a document and outputs entities mentioned in it.

The simplest toponym-resolution strategy is to assign a default sense to each recognized toponym using some prominence measure (such as population). Many researchers, including Amitay et al.,² Martins,²⁷ Purves et al.,³¹ Rauch et al.,³³ and Stokes et al.,⁴⁵ have done so in combination with other methods. For example, MetaCarta³³ assigns “default senses” in the form of probabilities based on how often each interpretation of a given toponym appears in a pre-collected corpus of geo-

tagged documents. It then alters these probabilities based on other heuristics (such as cue words and occurrence with nearby toponyms). The Spatially aware Information Retrieval on the Internet, or SPIRIT, project³¹ uses techniques similar to those in MetaCarta by searching for sentence cues, falling back to a “default sense” for a given geographic reference in the absence of stronger evidence.

Note that using default senses and probabilities based on corpora makes it nearly impossible for the relatively unknown location references in articles (such as any of the more than 2,000 lesser-known instances of “London” around the world) in articles in local newspapers to be selected as correct interpretations, since these smaller places will have appeared in few pre-created corpora of news articles. In contrast, NewsStand uses a concept we call a “local lexicon”^{22,32} that is associated with a news source and contains the set of locations in the source's geographic scope. For example, the local lexicon of readers living in “Columbus, OH” includes “Dublin,” “Amsterdam,” “London,” “Delaware,” “Africa,” “Alexandria,” “Baltimore,” and “Bremen” (see Figure 5). Readers outside the Columbus area, lacking these place names in their local lexicons, would likely think first of the more prominent same-name places.

Using the local lexicon is analogous to using a combination of the provider- and serving-scopes interpretation of the geographic scope described earlier. In particular, NewsStand learns its serving scope by forming a corpus of articles for each news source and collecting the geographic locations mentioned in the corpus that are local to it. This approach is based on observing that news articles are written with an assumption of where their reader is located. For example, when the location “Springfield, IL” is mentioned in a newspaper article in Illinois (such as Chicago), the qualifier “Illinois” or “IL” is most likely not used due to the expectation that its readers will make the correct interpretation automatically. On the other hand, an article in the *New York Times* would retain the “Illinois” qualifier when discussing “Springfield” to avoid any possible misunderstanding. Local lexicons are

particularly useful when users zoom in on the map, thereby focusing on relatively small areas where the articles are more local in nature. In this case, knowledge of the provider scope is extremely valuable in overcoming the geo/geo ambiguity.

The local lexicon can also be seen as a “resolving context” for toponym resolution. A related popular strategy^{2,27,31,45} for toponym resolution places the resolving context within a hierarchical geographic ontology that involves finding a geographic region in which many of the document’s toponyms can be resolved. For example, Web-a-Where² pursues such an approach by searching for several forms of hierarchical evidence in documents, including minimal resolving contexts and containment of adjacent toponyms (such as “College Park, MD”). It identifies a document’s geographic focus through a simple scoring algorithm that takes into account the gazetteer hierarchy, as well as a confidence score, for each location. Ding et al.⁶ used a similar approach. MetaCarta³³ and Google Book Search have no notion of a computed geographic focus, and thus require users to determine a focus for themselves. Instead of using content location, Mehler et al.²⁸ associated documents with provider location, which, at times, is equivalent to using the dateline. Note the central assumption behind finding a minimal resolving context is that the document under consideration has a single geographic focus, useful for resolving toponyms in that focus, but not for resolving distant toponyms mentioned in passing.

Note, too, the local lexicon is just one of many techniques NewsStand uses for toponym resolution, its need manifested by the fact that some features are associated with multiple records, or $|L(f)| > 1$. In particular, NewsStand resolves such ambiguous references through heuristic filters that select the most likely set of assignments for each reference, based on how a human would read an article. These filters rely on NewsStand’s initial assumption that locations mentioned in the article give evidence to each other, in terms of geographic distance, document distance,¹⁹ and hierarchical containment. The “object container filter” is one

such filter. It searches for pairs of geographic features $f_1, f_2 \in EFV$ separated in the article by containment keywords or punctuation symbols (such as “ f_1 in f_2 ” or “ f_1, f_2 ”). If it finds a pair of locations (l_1, l_2) , so $l_1 \in L(f_1)$, $l_2 \in L(f_2)$, and l_1 is contained in l_2 , then f_1 and f_2 are disambiguated as l_1 and l_2 , respectively. For example, suppose $f_1 = \text{“Brooklyn”}$ and $f_2 = \text{“NYC.”}$ Also, let $L(f_1) = \{\text{“Brooklyn, New York City,” “Brooklyn, Shelby County”}\}$ and $L(f_2) = \{\text{“New York City, New York County,” “North Yorkshire County, U.K.”}\}$. We now disambiguate f_1 as $l_1 = \text{“Brooklyn, New York City”}$ and f_2 as $l_2 = \text{“New York City, New York County.”}$ This disambiguation is justified by NewsStand’s observation that a pair of features that are textually close in the article, close geographically, and exhibit a hierarchy relationship are unlikely to occur by chance. Another example of this strategy is when a query involves lists of locations, in which case NewsStand tries to use proximity, sibling, and prominence clues to resolve the ambiguity.^{1,21}

Evaluation. To see how well NewsStand’s geotagging performs, rather than display a news category icon at a location, NewsStand can display the actual name of the location by setting the “layers” parameter to “location” instead of to “icon.” In this way, it can detect wrong geo/geo interpretations (such as placing “Los Angeles” in “Chile” instead of in “California”), as well as wrong classifications of non-geo as geo (such as “George” in “South Africa” instead of “George Anthony” from the 2012 Casey Anthony baby murder trial in “Orlando, FL”) but not vice versa.

Moreover, hovering over the name n of a location l (both in the “location” and “icon” layers) causes NewsStand to generate a minimap, as well as markers in the form of blue balls at all other locations k with the same name n on both the map and the minimap, such that at least one article cluster is associated with k . This minimap enables NewsStand to quickly find geotagging errors. Research is under way to use this information to learn better classifiers. The blue balls enable NewsStand to overcome possible toponym resolution errors by providing access to all articles it determines mention a particular location name n for any

interpretation k of n as long as at least one article is associated with interpretation k , even though k may not be the correct interpretation, thereby giving the user the final say. By examining all mentions of n for the correct interpretation subject to NewsStand’s stipulation that at least one article is associated with the interpretation (assuming 100% recall for toponym recognition with lower precision), the result is that NewsStand achieves 100% recall for toponym resolution for the interpretations of a location that are in its gazetteer, with lower precision, though it does not miss any. Note that in some sense NewsStand is ranking its responses, where the highest-ranked response is associated with the queried location on the main map and the lower-ranked responses are associated with the minimap.

Results of Lieberman’s and Samet’s experiments¹⁸ with handcrafted corpuses of articles showed that NewsStand’s toponym recognition¹⁸ and toponym resolution¹⁹ processes outperformed Reuters’s OpenCalais and Yahoo’s Placemaker, which are closed-source commercial products providing public Web APIs that allow for automated geotagging of documents. At one time, the MetaCarta system³³ provided a similar capability by recognizing spatial cue words (such as “city of”), as well as certain forms of postal addresses and textual representations of geographic coordinates in text documents.

Lessons Learned

Building NewsStand has taught us that the geotagging tasks of toponym recognition and resolution are much more complex than we originally envisioned. For example, NewsStand’s geotagger could use more semantic hints from a document to aid correct geotagging (such as landmarks and rivers). Moreover, geography can be used to improve the clustering of news articles by modifying the TF-IDF framework so terms that are spatial synonyms are merged into one term instead of being treated as separate terms. A primary difficulty involves evaluating NewsStand’s performance on these tasks. Comparing NewsStand with other systems means having to use standardized datasets known as “corpuses.” We performed

this comparison for both components of the geotagging task, with emphasis on recall rather than precision, achieving superior results.^{18,19} Nevertheless, this evaluation method involves two shortcomings: the datasets are far too small, and “corpuses are like corpses” in that news and language are constantly changing. The news data can be characterized as streaming data. The evaluation should be conducted more in a spirit of sampling, as in inspection/quality control tasks, something we intend to do in the future.

In a Web browser, NewsStand works well with the mapping API provided by Google Maps to display topics. It has also been adapted to work with Bing Maps and the Google Earth plugin, though the plugin leads to a number of display problems due to the limited number of supported platforms. NewsStand has also been ported to work on devices with a gesturing touchscreen interface (such as smartphones and tablets) for use with Web browsers,⁴² albeit with a slightly different user interface, and as an app³⁸ for the iPhone, Android, and Windows Phone platforms (see Figure 6). NewsStand does not have a “public” API, though much of its functionality and ability to handle different smartphone platforms makes use of its “private” API.

Differences between the browser-

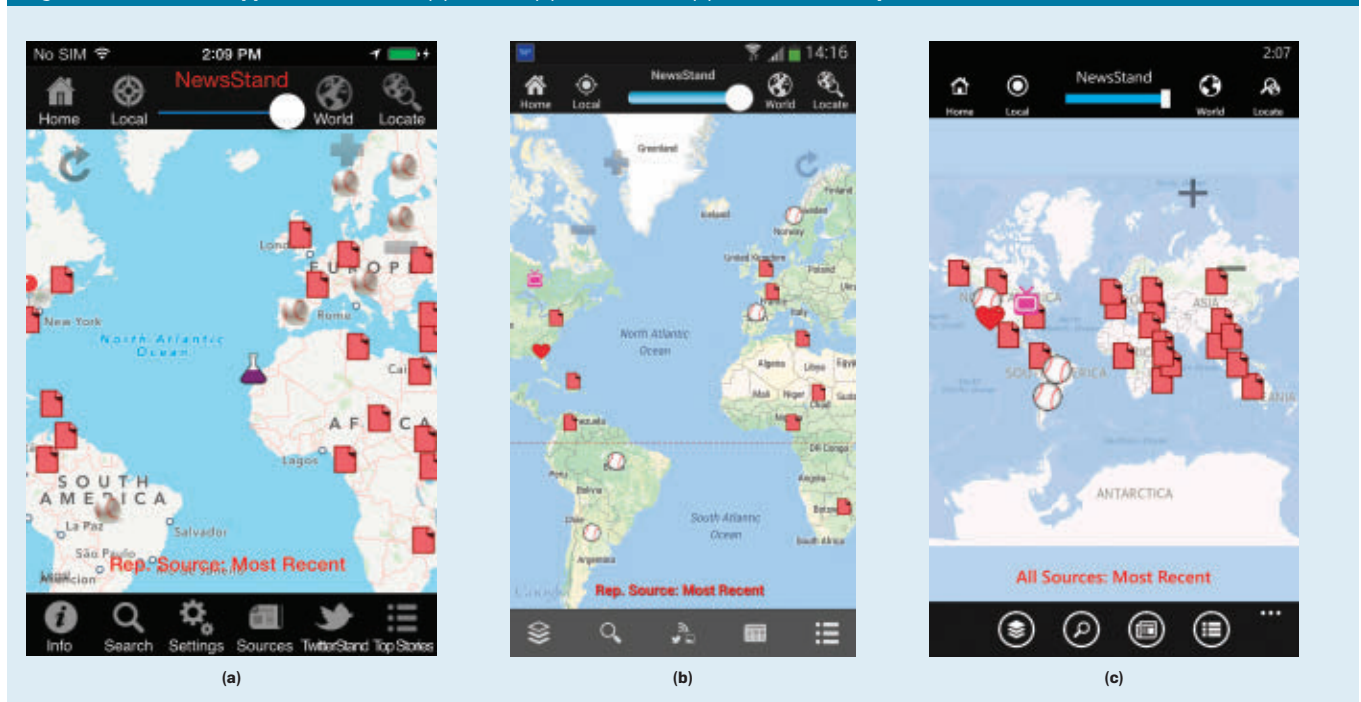
based Web environment and native app environment for mobile devices require changes in user behavior or habits. For example, map-centric applications on the Web function best as single-page applications, meaning external links (such as to news articles in NewsStand) are opened in separate browser tabs to preserve the NewsStand App and its state, which would not occur if the news articles would be opened in the same tab. A concrete example of the undesired ramification of opening the external link in a separate browser tab is that users cannot make use of the “back” button to return to the app and its prior state. Instead, they must explicitly close the newly opened tab, in which case the invoking tab and its state are implicitly restored. Such problems do not arise in the native app environment, which can coordinate fluid transitions among many windows, thereby providing more user-friendly interaction, with the trade-off, in our example, that only one external link to a news article can be opened at a time.

Porting NewsStand to a variety of mobile/smartphone platforms revealed a lack of adherence to classical cartographic principles in the implementations of the underlying mapping APIs. As a result, consistency issues surfaced for some operations (such as zooming

and panning). For example, once the name of a location appears in the map, that name should continue to be present as long as the location remains in the window as the user zooms in further or pans.⁴⁰ Curiously, some mapping apps on mobile and smartphone platforms do not enable zooming out so the entire world can be seen on the screen (such as in the Google Maps and Apple Maps mobile/smartphone mapping APIs), thereby requiring further panning to see the rest of the world, though it is present in the “here” Maps API.⁴⁰ This phenomenon is especially annoying in NewsStand where users want to see what is happening in the whole world.⁴⁰ Minimaps alleviate the problem via, in part, the orange balls showing all other locations mentioned in a particular article highlighted with a headline info bubble.

We had to account for not being able to hover in devices that make use of a gesturing interface when designing the user interface, as it means some features would have to be implemented differently on gesturing-enabled platforms. In particular, hovering enables the user to observe the spatial variability of phenomena being displayed, or expanded, as the pointing device passes over the location. The gesturing interface requires a tap or

Figure 6. NewsStand App screenshots for (a) iPhone, (b) Android, and (c) Windows Phone platforms.




click for such a display action to take place since a continuous motion of a finger over an area is interpreted as a single tap or click, and it is thus difficult to observe spatial variability. On the other hand, the absence of hovering means a transition from map location l to another map location b can be made by tapping on b . In contrast, the hovering needed to transition from l to b may lead to certain actions being taken that would destroy the current state of the system.

The design challenge of speedy map labeling involves placement of the minimap in close proximity to the headline bubble and associated info box. It also arises in dynamic display of labels (such as disease names in Figure 3 and keywords and names of people and brands). Our goal is to do so at interactive speeds under panning and zooming, achieving it through techniques developed for dynamic map labeling³⁰ and incorporated in the PhotoStand system.³⁷


Conclusion

We reviewed the design goals and functionality of the NewsStand system for using a map to read news on the Web, harnessing the power of spatial synonyms. NewsStand demonstrates that extracting geographic content from news articles taps a previously unseen dimension of information that can aid understanding news events across space and time. NEWS can indeed be described as an acronym of north, east, west, south. The increasing popularity of geotagged content on the Internet will enable compelling applications for systems like NewsStand in other knowledge domains. For example, sentiment/content analysis can reveal how the same news story can be interpreted by people in different countries or in different languages and hot-spot analysis based on news, tweets, or other sources of data feeds. Moreover, NewsStand represents a contribution to the emerging field of computational journalism.⁸

Future work includes using a map-query interface to access other media through representative images (such as PhotoStand³⁷), videos, and audio clips. We are also working on incorporating other sources of news and information. For example, we have in-




NewsStand currently indexes 10,000 news sources and processes approximately 50,000 news articles per day.



corporated Twitter tweets into NewsStand, resulting in the TwitterStand system⁴⁴ where the idea is to tap the large volume of news articles to serve as a kind of clustering corpus so very short and information-sparse tweets can be clustered using existing news clusters. An interesting aspect of this method is that the tweets, due to their short length, usually have little or no geographic content but, when clustered, inherit the geographic information associated with the geographic focus of the cluster with which they are associated. The novel result is the focus is now on the geographic regions about which a user is tweeting rather than on the geographic regions from which the user is tweeting (easy to find when the tweeting device has GPS capability). This focus is useful when tweeting about future events,¹⁴ but one must be careful in choosing whose tweets to follow.¹¹

Acknowledgments

This article is based on an earlier paper by Teitler et al.⁴⁶ This work was supported in part by the National Science Foundation under Grants IIS-07-13501, IIS-08-12377, CCF-08-30618, IIS-10-18475, IIS-12-19023, and IIS-13-20791, as well as by the Office of Policy Development & Research of the U.S. Department of Housing and Urban Development, Microsoft Research, Google Research, Nvidia, the E.T.S. Walton Visitor Award of the Science Foundation of Ireland, and the National Center for Geocomputation at the National University of Ireland at Maynooth. We also thank Larry Brandt, Jim Gray, Keith Marzullo, and Maria Zemankova for championing it. 

References

1. Adelfio, M.D. and Samet, H. Structured toponym resolution using combined hierarchical place categories. In *Proceedings of the Seventh ACM SIGSPATIAL Workshop on Geographic Information Retrieval* (Orlando, FL, Nov. 5). ACM Press, New York, 2013, 49–56.
2. Amitay, E., Har'El, N., Sivan, R., and Soffer, A. Web-a-Where: Geotagging Web content. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Sheffield, U.K., July 25–29). ACM Press, New York, 2004, 273–280.
3. Aref, W.G. and Samet, H. Efficient processing of window queries in the pyramid data structure. In *Proceedings of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems* (Nashville, TN, Apr. 2–4). ACM Press, New York, 1990, 265–272.
4. Baldwin, B. and Carpenter, B. *Lingpipe*; <http://alias-i.com/lingpipe/>
5. Chum, O., Philbin, J., Isard, M., and Zisserman, A.

- Scalable near-identical image and shot detection. In *Proceedings of the Sixth ACM International Conference on Image and Video Retrieval* (Amsterdam, The Netherlands, July 9–11). ACM Press, New York, 2007, 549–556.
6. Ding, J., Gravano, L., and Shivakumar, N. Computing geographical scopes of Web resources. In *Proceedings of the 26th International Conference on Very Large Data Bases* (Cairo, Egypt, Sept. 10–14). Morgan Kaufmann, San Francisco, 2000, 545–556.
 7. Duda, R.O., Hart, P.E., and Stork, D.G. *Pattern Classification, Second Edition*. Wiley Interscience, New York, 2000.
 8. Essa, I. *Computation + Journalism: A Study of Computation and Journalism and How They Impact Each Other*; <http://www.computation-and-journalism.com/>
 9. Francis, W.N. A standard corpus of edited present-day American English. *College English* 26, 4 (Jan. 1965), 267–273.
 10. Freifeld, C.C., Mandl, K.D., Reis, B.Y., and Brownstein, J.S. HealthMap: Global infectious disease monitoring through automated classification and visualization of Internet media reports. *Journal of the American Medical Informatics Association* 15, 2 (Mar. 2008), 150–157.
 11. Gramsly, N. and Samet, H. Seeder finder: Identifying additional needles in the Twitter haystack. In *Proceedings of the Fifth ACM SIGSPATIAL International Workshop on Location-Based Social Networks* (Orlando, FL, Nov. 5). ACM Press, New York, 2013, 44–53.
 12. Hjaltason, G.R. and Samet, H. Speeding up construction of PMR quadtree-based spatial indexes. *Very Large Data Bases Journal* 11, 2 (Oct. 2002), 109–137.
 13. Hoffart, J., Yosef, M.A., Bordino, I., Fürstenauf, H., Pinkal, M., Spaniol, M., Taneva, B., Thater, S., and Weikum, G. Robust disambiguation of named entities in text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (Edinburgh, Scotland, July 27–31). Association for Computational Linguistics, Stroudsburg, PA, 2011, 782–792.
 14. Jackoway, A., Samet, H., and Sankaranarayanan, J. Identification of live news events using Twitter. In *Proceedings of the Third ACM SIGSPATIAL International Workshop on Location-Based Social Networks* (Chicago, Nov. 1). ACM Press, New York, 2011, 25–32.
 15. Lafferty, J.D., McCallum, A., and Peireira, F.C.N. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning* (Williamstown, MA, June 28–July 1). Morgan Kaufman, San Francisco, 2001, 282–289.
 16. Lan, R., Lieberman, M.D., and Samet, H. The picture of health: Map-based, collaborative spatio-temporal disease tracking. In *Proceedings of the First ACM SIGSPATIAL International Workshop on the Use of GIS in Public Health* (Redondo Beach, CA, Nov. 6). ACM Press, New York, 2012, 27–35.
 17. Leidner, J.L. *Toponym Resolution in Text: Annotation, Evaluation and Applications of Spatial Grounding of Place Names*. Ph.D. thesis, University of Edinburgh, Edinburgh, Scotland, U.K., Oct. 2006; <https://www.era.lib.ed.ac.uk/bitstream/1842/1849/1/leidner-2007-phd.pdf>
 18. Lieberman, M.D. and Samet, H. Multifaceted toponym recognition for streaming news. In *Proceedings of the 34th International Conference on Research and Development in Information Retrieval* (Beijing, July 24–28). ACM Press, New York, 2011, 843–852.
 19. Lieberman, M.D. and Samet, H. Adaptive context features for toponym resolution in streaming news. In *Proceedings of the 35th International Conference on Research and Development in Information Retrieval* (Portland, OR, Aug. 12–16). ACM Press, New York, 2012, 731–740.
 20. Lieberman, M.D. and Samet, H. Supporting rapid processing and interactive map-based exploration of streaming news. In *Proceedings of the 20th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (Redondo Beach, CA, Nov. 7–9). ACM Press, New York, 2012, 179–188.
 21. Lieberman, M.D., Samet, H., and Sankaranarayanan, J. Geotagging: Using proximity, sibling, and prominence clues to understand comma groups. In *Proceedings of the Sixth Workshop on Geographic Information Retrieval* (Zürich, Switzerland, Feb. 18–19). ACM Press, New York, 2010.
 22. Lieberman, M.D., Samet, H., and Sankaranarayanan, J. Geotagging with local lexicons to build indexes for textually specified spatial data. In *Proceedings of the 26th IEEE International Conference on Data Engineering* (Long Beach, CA, Mar. 1–6). IEEE Press, 2010, 201–212.
 23. Lieberman, M.D., Samet, H., Sankaranarayanan, J., and Sperling, J. STEWARD: Architecture of a spatio-textual search engine. In *Proceedings of 15th ACM International Symposium on Advances in Geographic Information Systems* (Seattle, Nov. 7–9). ACM Press, New York, 2007, 186–193.
 24. Lieberman, M.D., Sankaranarayanan, J., Samet, H., and Sperling, J. Augmenting spatio-textual search with an infectious disease ontology. In *Proceedings of the Workshop on Information Integration Methods, Architectures, and Systems* (Cancun, Mexico, Apr. 11–12). IEEE Computer Society, 2008, 266–269.
 25. Lowe, D.G. Object recognition from local scale-invariant features. In *Proceedings of the Seventh International Conference on Computer Vision* (Corfu, Greece, Sept. 20–25). IEEE Computer Society, 1999, 1150–1157.
 26. Markowitz, A., Brinkhoff, T., and Seeger, B. Exploiting the Internet as a geospatial database. In *Proceedings of the Workshop on Next Generation Geospatial Information* (Cambridge, MA, Oct. 19–21, 2003).
 27. Martins, B., Manguinhas, H., Borbinha, J., and Siabato, W. A geo-temporal information extraction system for processing descriptive metadata in digital libraries. *e-Perimtron* 4, 1 (2009), 25–37.
 28. Mehler, A., Bao, Y., Li, X., Wang, Y., and Skiena, S. Spatial analysis of news sources. *IEEE Transactions on Visualization and Computer Graphics* 12, 5 (Sept.–Oct. 2006), 765–772.
 29. Milne, D. and Witten, I.H. Learning to link with Wikipedia. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management* (Napa Valley, CA, Oct. 26–30). ACM Press, New York, 2008, 509–518.
 30. Nutanong, S., Adelfio, M.D., and Samet, H. Multiresolution select-distinct queries on large geographic point sets. In *Proceedings of the 20th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (Redondo Beach, CA, Nov. 7–9). ACM Press, New York, 2012, 159–168.
 31. Purves, R.S., Clough, P., Jones, C.B., Arampatzis, A., Bucher, B., Finch, D., Fu, G., Joho, H., Syed, A.K., Vaid, S., and Yang, B. The design and implementation of SPIRIT: A spatially aware search engine for information retrieval on the Internet. *International Journal of Geographical Information Systems* 21, 7 (2007), 717–745.
 32. Quercini, G., Samet, H., Sankaranarayanan, J., and Lieberman, M.D. Determining the spatial reader scopes of news sources using local lexicons. In *Proceedings of the 18th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (San Jose, CA Nov. 3–5). ACM Press, New York, 2010, 43–52.
 33. Rauch, E., Bukatin, M., and Baker, K. A confidence-based framework for disambiguating geographic terms. In *Proceedings of the HLT-NAACL Workshop on Analysis of Geographic References* (Edmonton, Canada). Association for Computational Linguistics, Stroudsburg, PA, 2003, 50–54.
 34. Rizzo, G. and Troncy, R. NERD: A framework for unifying named entity recognition and disambiguation extraction tools. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics* (Avignon, France, Apr. 23–27). Association for Computational Linguistics, Stroudsburg, PA, 2012, 73–76.
 35. Salton, G. and Buckley, C. Term-weighting approaches in automatic text retrieval. *Information Processing & Management* 24, 5 (1988), 513–523.
 36. Samet, H. *Foundations of Multidimensional and Metric Data Structures*. Morgan Kaufmann, San Francisco, 2006.
 37. Samet, H., Adelfio, M.D., Fruin, B.C., Lieberman, M.D., and Sankaranarayanan, J. PhotoStand: A map query interface for a database of news photos. *Proceedings of the VLDB Endowment* 6, 12 (Aug. 2013), 1350–1353.
 38. Samet, H., Adelfio, M.D., Fruin, B.C., Lieberman, M.D., and Teitler, B.E. Porting a Web-based mapping application to a smartphone app. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (Chicago, Nov. 2–4). ACM Press, New York, 2011, 525–528.
 39. Samet, H., Alborzi, H., Brabec, F., Esperança, C., Hjaltason, G.R., Morgan, F., and Tanin, E. Use of the SAND spatial browser for digital government applications. *Commun. ACM* 46, 1 (Jan. 2003), 63–66.
 40. Samet, H., Fruin, B.C., and Nutanong, S. Daking it out at the smartphone mobile app mapping API corral: Apple, Google, and the competition. In *Proceedings of the First ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems* (Redondo Beach, CA, Nov. 6). ACM Press, New York, 2012, 41–48.
 41. Samet, H., Rosenfeld, A., Shaffer, C.A., and Webber, R.E. A geographic information system using quadtrees. *Pattern Recognition* 17, 6 (Nov./Dec. 1984), 647–656.
 42. Samet, H., Teitler, B.E., Adelfio, M.D., and Lieberman, M.D. Adapting a map query interface for a gesturing touchscreen interface. In *Proceedings of the 20th International World Wide Web Conference* (Hyderabad, India, Mar. 28–Apr. 1). ACM Press, New York, 2011, 257–260.
 43. Sankaranarayanan, J., Samet, H., Teitler, B., Lieberman, M.D., and Sperling, J. TwitterStand: News in tweets. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (Seattle, Nov. 4–6). ACM Press, New York, 2009, 42–51.
 44. Sarma, A.D., Lee, H., Gonzales, H., Madhavan, J., and Halevy, A. Efficient spatial sampling of large geographical tables. In *Proceedings of the ACM SIGMOD Conference* (Scottsdale, AZ, May 20–24). ACM Press, New York, 2012, 193–204.
 45. Stokes, N., Li, Y., Moffat, A., and Rong, J. An empirical study of the effects of NLP components on geographic IR performance. *International Journal of Geographical Information Systems* 22, 3 (Mar. 2008), 247–264.
 46. Teitler, B., Lieberman, M.D., Panozzo, D., Sankaranarayanan, J., Samet, H., and Sperling, J. NewsStand: A new view on news. In *Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (Irvine, CA, Nov. 5–7). ACM Press, New York, 2008, 144–153.
 47. Zhou, G. and Su, J. Named entity recognition using an HMM-based chunk tagger. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics* (Philadelphia, PA, July 6–12). Association for Computational Linguistics, Stroudsburg, PA, 2002, 473–480.
-
- Hanan Samet** (hjs@cs.umd.edu) is a distinguished university professor in the Computer Science Department, Center for Automation Research, and Institute for Advanced Computer Studies at the University of Maryland, College Park, MD.
- Jagan Sankaranarayanan** (sjagan@gmail.com) is a member of the research staff at NEC Labs, Cupertino, CA; his research for this article was conducted while he was an assistant research scientist at the Institute for Advanced Computer Studies at the University of Maryland, College Park, MD.
- Michael D. Lieberman** (mike.d.lieberman@gmail.com) is a research scientist at the Johns Hopkins University Applied Physics Laboratory, Laurel, MD; his research for this article was part of his Ph.D. in computer science at the University of Maryland, College Park, MD.
- Marco D. Adelfio** (marco@cs.umd.edu) is a Ph.D. candidate in computer science at the University of Maryland, College Park, MD.
- Brendan C. Fruin** ([bcfruin@gmail.com](mailto:bcfuin@gmail.com)) is a software engineer at Zillow, Seattle, WA; his research for this article was part of his master's in computer science at the University of Maryland, College Park, MD.
- Jack M. Lotkowski** (JackLotkowski@gmail.com) is an undergraduate student at the University of Maryland, College Park, MD.
- Daniele Panozzo** (daniele.panozzo@gmail.com) is a senior researcher at ETH, Zürich, Switzerland; his research for this article was conducted while he was a visiting student at the Institute for Advanced Computer Studies at the University of Maryland, College Park, MD.
- Jon Sperling** (jonxsperling@gmail.com) is a senior researcher in the Office of Policy Development and Research of the U.S. Department of Housing and Urban Development, Washington, D.C.
- Benjamin E. Teitler** (bteitler@cs.umd.edu) did research for this article as part of his master's in computer science at the University of Maryland, College Park, MD.

DOI:10.1145/2629489

This collaboratively edited knowledgebase provides a common source of data for Wikipedia, and everyone else.

BY DENNY VRANDEČIĆ AND MARKUS KRÖTZSCH

Wikidata: A Free Collaborative Knowledgebase

UNNOTICED BY MOST of its readers, Wikipedia continues to undergo dramatic changes, as its sister project Wikidata introduces a new multilingual “Wikipedia for data” (<http://www.wikidata.org>) to manage the factual information of the popular online encyclopedia. With Wikipedia’s data becoming cleaned and integrated in a single location, opportunities arise for many new applications.

Originally conceived in 2001 as a mainly text-based resource, Wikipedia¹ has collected increasing amounts of structured data, including numbers, dates, coordinates, and many types of relationships, from family trees to the taxonomy of species. It has become a resource of enormous value, with potential applications across all areas of science, technology, and culture. This development is hardly surprising, given that Wikipedia is committed to “a world in which every single human being can freely share in the sum of all knowledge,”

according to its vision statement (<https://wikimediafoundation.org/wiki/Vision>). There is no question this must include data that can be searched, analyzed, and reused.

It may be surprising that Wikipedia does not provide direct access to most of it, through either query services or downloadable data exports. Actual use of the data is rare and often restricted to specific pieces of information (such as geo-tags of Wikipedia articles used in Google Maps). The reason for this striking gap between vision and reality is that Wikipedia’s data is buried in 30 million Wikipedia articles in 287 languages from which extraction is inherently very difficult.

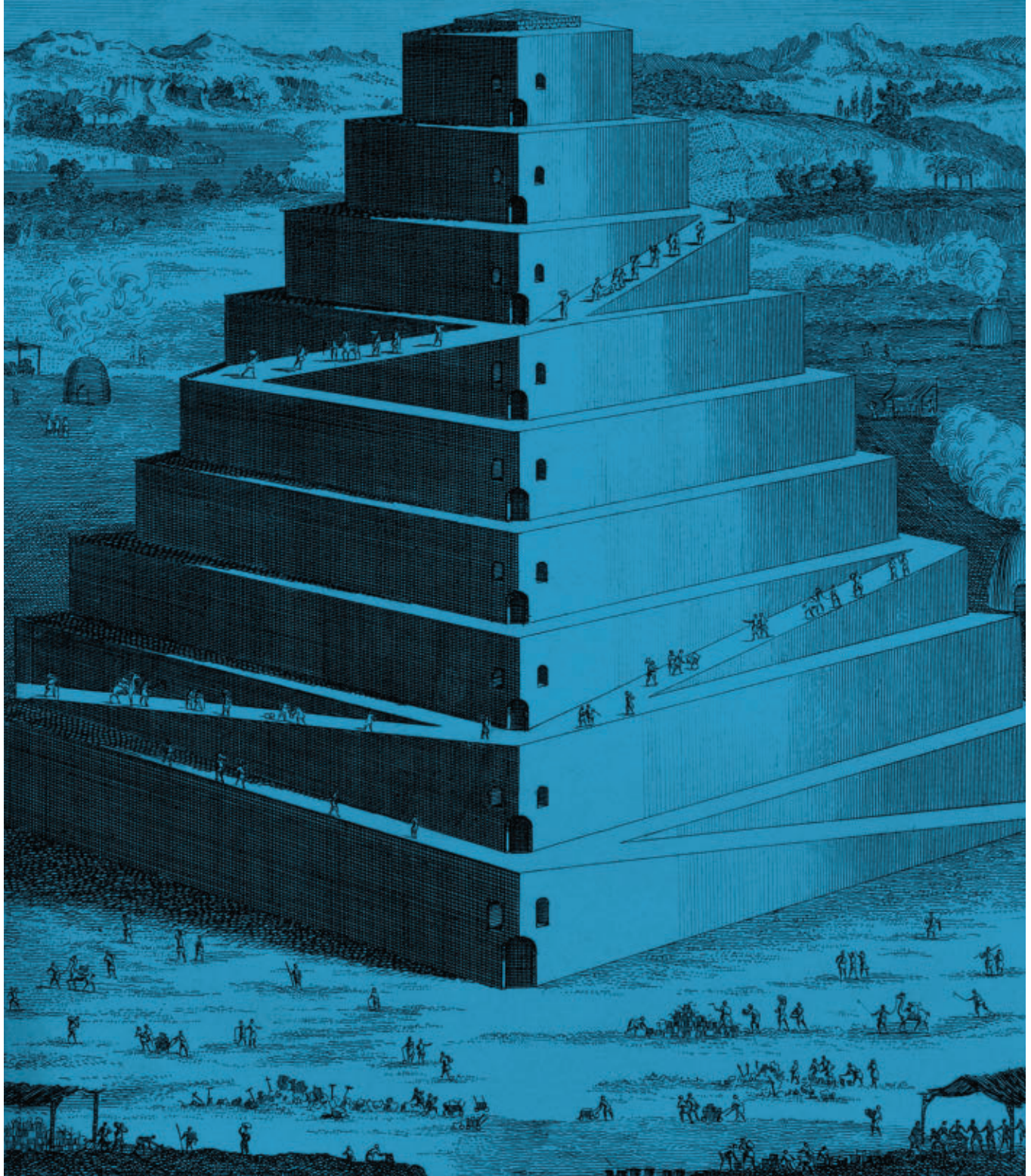
This situation is unfortunate for anyone wanting to use the data but is also an increasing threat to Wikipedia’s main goal of providing up-to-date, accurate, encyclopedic knowledge. The same information often appears in articles in many languages and in many articles within a single language. Population numbers for Rome, for example, can be found in English and Italian articles about Rome but also in the English article “Cities in Italy.” The numbers are all different.

Wikidata aims to overcome such inconsistencies by creating new ways for Wikipedia to manage its data on a global scale; see the result at <http://www.wikidata.org>. The following essential design decisions characterize the Wikidata approach.

Open editing. As in Wikipedia, Wikidata allows every user to extend and edit the stored information, even without creating an account. A form-based interface makes editing easy.

» key insights

- Wikidata provides a free collaborative knowledgebase all can share.
- Wikidata has quickly become one of the most active Wikimedia projects.
- Wikipedia, as well as an increasing number of other sites, taps content from Wikidata in every pageview, magnifying the data’s visibility and usefulness.



ENGRAVING BY ISAAC BASIRE, 1733

Community control. Not only is the actual data controlled by the contributor community, so, too, is the schema of the data. Contributors edit the population number of Rome but also decide whether there is such a number in the first place.

Plurality. It would be naive to expect global agreement on the “true” data, since many facts are disputed or simply uncertain. Wikidata allows conflicting data to coexist and provides mechanisms to organize this plurality.

Secondary data. Wikidata gathers facts published in primary sources, together with references to these sources; for example, there is no “true population of Rome” but rather a “population of Rome as published by the city of Rome in 2011.”

Multilingual data. Most data is not tied to a single language; numbers, dates, and coordinates have universal meaning, so labels like “Rome” and “population” are translated into many different languages. Wikidata is mul-

tilingual by design. While Wikipedia has independent editions for each language, there is only one Wikidata site.

Easy access. Wikidata’s goal is to allow data to be used both in Wikipedia and in external applications. Data is exported through Web services in several formats, including JavaScript Object Notation, or JSON, and Resource Description Framework, or RDF. Data is published under legal terms that allow the widest possible reuse.

Continuous evolution. In the best

tradition of Wikipedia, Wikidata grows with its community of editors and developers and the tasks they give it. Rather than develop a perfect system to be presented to the world in a couple of years, new features are deployed incrementally and as early as possible.

These properties characterize Wikidata as a specific kind of curated database.⁸

Data in Wikipedia

The value of Wikipedia's data has long been obvious, with many efforts to use it. The Wikidata approach is to crowdsource data acquisition, allowing a global community to edit the data. This extends the traditional wiki approach of allowing users to edit a website. Wiki is a Hawaiian word for fast; Ward Cunningham, who created the first wiki in 1995, used it to emphasize that his website could be changed quickly.¹⁷

The most popular such system is Semantic MediaWiki, or SMW,¹⁵ which extends MediaWiki, the software used to run Wikipedia,² with data-management capabilities. SMW was originally proposed for Wikipedia but was quickly used on hundreds of other websites as well. Unlike Wikidata, SMW manages data as part of its textual content, thus hindering creation of a multilingual, single knowledgebase supporting all Wikimedia projects. Moreover, the data model of Wikidata is more elaborate than that of SMW, allowing users to capture more complex information. In spite of these differences, SMW has had a great influence on Wikidata, and the two projects share code for common tasks.

Other examples of free knowledgebase projects are OpenCyc and Freebase. OpenCyc is the free part of Cyc,¹⁶ which aims for a much more comprehensive and expressive representation of knowledge than Wikidata. OpenCyc is released under a free license and available to the public, but, unlike Wikidata, is not editable by the public. Freebase, acquired by Google in 2010, is an online platform that allows communities to manage structured data.⁷ Objects in Freebase are classified by types that prescribe what kind of data an object can have; for example, Freebase classifies Einstein as a "musical artist" since it would otherwise not be possible to refer to recordings of his speeches. Wikidata supports the use



Wikipedia's data is buried in 30 million Wikipedia articles in 287 languages from which extraction is inherently very difficult.



of arbitrary properties on all objects. Other differences from Wikidata are related to multi-language support, source information, and the proprietary software used to run the site. The latter is critical for Wikipedia, which is committed to running on a fully open source software stack to allow all to fork, or copy and create one's own version of the project.

Other approaches to creating knowledgebases from Wikipedia have aimed at extracting data from Wikipedia, most notably DBpedia⁶ and Yago,¹³ that extract information from Wikipedia categories and from infoboxes, the tabular summaries in the upper-right area of many Wikipedia articles. Additional mechanisms help improve extraction quality. Yago includes temporal and spatial context information, but neither DBpedia nor Yago extract source information.

Wikipedia data, obtained from these projects or through custom extraction methods, has been used to improve object search in Google's Knowledge Graph (based on Freebase) and Facebook's Open Graph and in answering engines, including Wolfram Alpha,²⁴ Evi,²¹ and IBM's Watson.¹⁰ Wikipedia's geo-tags are also used by Google Maps. All these applications would benefit from up-to-date, machine-readable data exports (such as the way Google Maps shows India's Chennai district in the polar Kara Sea, next to Ushakov Island). Among these applications, Freebase and Evi are the only ones that also allow users to edit or to at least extend the data.

A Short History of Wikidata

Wikimedia launched Wikidata in October 2012. Initially, features were limited, with editors only able to create items and connect them to Wikipedia articles. In January 2013, three Wikipedias—first Hungarian, then Hebrew and Italian—began to connect to Wikidata. Meanwhile, the Wikidata community had already created more than three million items. The English Wikipedia followed in February, and all Wikipedias were connected to Wikidata in March.

Wikidata received input from more than 40,000 contributors as of February 2014. Since May 2013, it has continuously had more than 3,500 active contributors, those making at least

five edits per month. These numbers make it one of the most active Wikimedia projects.

In March 2013, Wikimedia introduced Lua as a scripting language for automatically creating and enriching parts of articles (such as the infoboxes mentioned earlier). Lua scripts can access Wikidata, allowing Wikipedia editors to retrieve, process, and display data.

Many other features were introduced in 2013, and development is planned to continue for the foreseeable future.

Out of Many, One

The first challenge for the Wikidata community was to reconcile the 287 language editions of Wikipedia; for example, for Wikidata to be truly multilingual, the object representing “Rome” must be one and the same across all languages. Fortunately, Wikipedia already has a closely related mechanism: language links, displayed on the left side of each article, connecting articles in different languages. These links were created from user-edited text entries at the bottom of each article, leading to a quadratic number of links; for example, each of the 207 articles on Rome included a list of 206 links to all other articles on Rome—a total of 42,642 lines of text. By the end of 2012, 66 of the 287 language editions of Wikipedia included more text for language links than for actual article content.

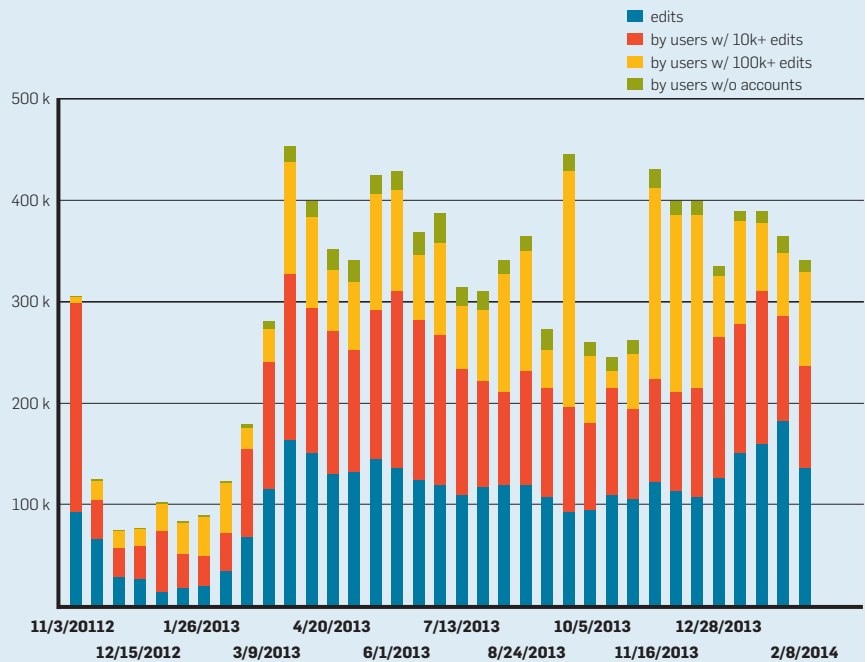
It would clearly be better to store and manage language links in a single location, and so became Wikidata’s first task. For every Wikipedia article, a page has now been created on Wikidata for managing links to related Wikipedia articles in all languages; these pages are called “items.” Initially, only a limited amount of data could be stored for each item: a list of language links, a label, a list of aliases, and a one-line description. Labels, aliases, and descriptions can now be specified individually for up to 358 languages.

The Wikidata community has created bots to move language links from Wikipedia to Wikidata, and more than 240 million links were removed from Wikipedia. Most language links displayed on Wikipedia are served from Wikidata. It is still possible

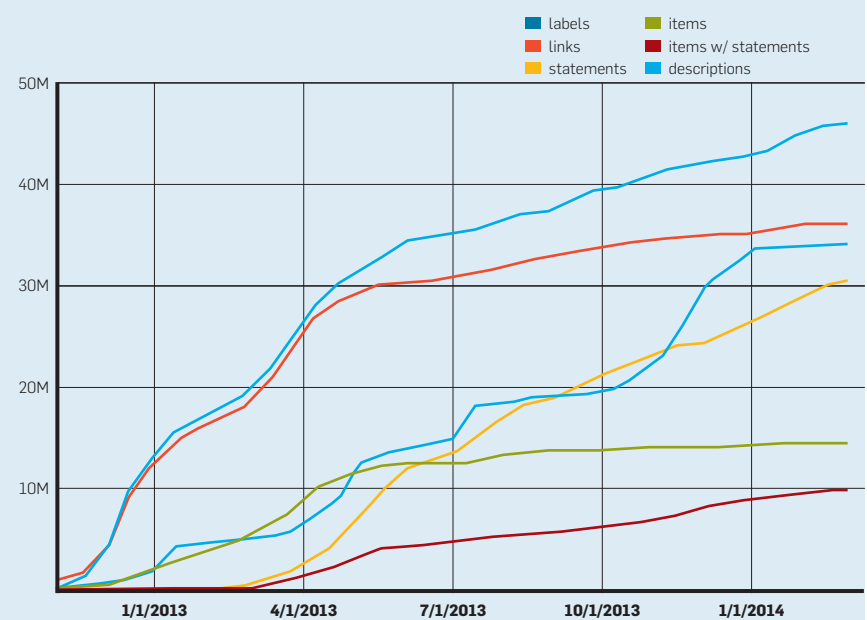
Figure 1. Screenshot of a complex statement in Wikidata.



Figure 2. Growth of Wikidata: (a) bi-weekly number of edits for different editor groups and (b) size of knowledgebase.



(a)



(b)

to add custom links in an article, as needed in the rare cases where links are not bi-directional; some articles refer to more general articles in other languages, while Wikidata deliberately connects pages that cover the same subject. By importing language links, Wikidata gained a huge set of initial items “grounded” in actual Wikipedia pages.

Simple Data (Properties and Values)

To store structured data beyond text labels and language links, Wikidata uses a simple data model. Data is basically described through property-value pairs; for example, the item for “Rome” might have a property “population” with value “2,777,979.” Properties are objects and have their own Wikidata pages with labels, aliases, and descriptions. Unlike items, however, these pages are not linked to Wikipedia articles.

On the other hand, property pages always specify a datatype that defines which type of values the property can have. “Population” is a number; “has father” relates to another Wikidata item; and “postal code” is a string. This information is important for providing adequate user interfaces and ensuring the validity of inputs. There are only a small number of datatypes, mainly quantity, item, string, date and time, geographic coordinates, and URL. Data is international, though its display may be language-dependent; for example, the number 1,003.5 is written “1.003,5” in German and “1 003.5” in French.

Not-So-Simple Data

Property-value pairs are too simple in many cases; for example, Wikipedia says the population of Rome was 2,651,040 “as of 2010” based on “estimations” published by the National Institute for Statistics, or Istat, in Italy

(<http://www.istat.it/>); see Figure 1 for how Rome statistics can be represented in Wikidata. Even leaving source information aside, the information cannot be expressed easily in property-value pairs. One could use a property “estimated population in 2010” or create an item “Rome” in 2010 to specify a value for its “estimated population.” However, either solution is clumsy and impractical. As suggested by Figure 1, we would like the data to contain a property “as of” with value “2010” and a property “method” with value “estimation.” These property-value pairs do not refer to Rome but to the assertion that Rome has a population of 2,651,040. We thus arrive at a model where the property-value pairs assigned to items can have additional subordinate property-value pairs we call “qualifiers.”

Qualifiers can be used to state contextual information (such as the validity time for an assertion). They can also be used to encode ternary relations that elude the property-value model; for example, to say Meryl Streep played Margaret Thatcher in the movie *The Iron Lady*, one could add to the item of the movie a property “cast member” with value “Meryl Streep” and an additional qualifier “role = Margaret Thatcher.”

Such qualifiers illustrate why we adopted an extensible set of qualifiers instead of restricting ourselves to the most common qualifiers (such as for temporal information). Qualifiers in their current form are indeed an almost direct representation of data found in Wikipedia infoboxes. This solution resembles known approaches to representing context information.^{11,18} It should not, however, be misunderstood as a workaround to represent relations of higher arity in graph-based data models, since Wikidata statements do not have a fixed (or even bounded) arity in this sense.²⁰

Wikidata also allows for two special types of statements: First, it is possible to specify that the value of a property is unknown; for example, one can say Ambrose Bierce’s day of death is unknown rather than not say anything about it, clarifying he is certainly not among the living. Second, one can say a property has no value at all (such as in asserting Australia has no countries sharing its borders). It is important to distinguish this situ-

Basic statistics about Wikidata (August 2014).

Supported languages	358
Labels	52,811,608
Descriptions	37,636,220
Aliases	8,765,542
Items	15,792,256
Items with statements	11,986,708
Items with ≥5 statements	3,017,227
Item with most statements: Rio Grande do Sul	513
Statements	43,189,145
Statements with reference	23,242,779
Properties	1,176
<i>Most-used properties</i>	
– instance of	10,892,599
– country	2,236,846
– sex or gender	2,203,270
Registered contributors	54,670
– with 5+ edits in June 2014	4,989
Edits	157,531,945
Use of datatypes	
– Wikidata items	29,199,563
– Strings	9,233,241
– Points in time	2,287,271
– Geocoordinates	1,620,508
– Media files	673,769
– URLs	97,949
– Numbers (new in 2014)	73,627

ation from the common case that information is simply incomplete. It would be wrong to consider these two cases as special values, becoming clear when considering queries that ask for items sharing the same value for a property; otherwise, one would have to conclude Australia and Iceland have a common neighbor.

Further details on the Wikidata data model and its expression in Web Ontology Language in Resource Description Framework, or OWL/RDF, can be found in Erxleben et al.⁹

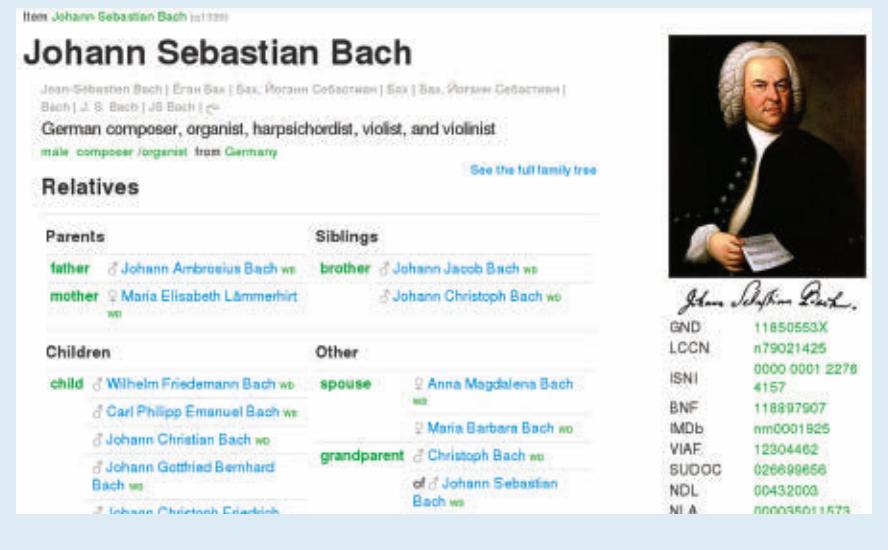
Citation Needed

Property assertions, possibly with qualifiers, provide a rich structure for expressing arbitrary claims. In Wikidata, every such claim can include a list of references to sources that support the claim. Including references agrees with Wikipedia’s goal of being a secondary (or tertiary) source that does not publish its own research but rather gathers information published in other primary (or secondary) sources.

There are many ways to specify a reference, depending on whether it is a book, a curated database, a website, or something else entirely. Moreover, some sources may be represented by Wikidata items, while others are not. In this light, a reference is simply a list of property-value pairs, leaving the details of reference modeling to the community. Note Wikidata does not automatically record provenance¹⁹ but does provide for the structural representation of references.

Sources are also important as context information. Different sources often make contradictory claims, yet Wikidata intends to represent all views rather than choose one “true” claim. Combined with the context information provided by qualifiers (such as for temporal context), many statements could be stored about a single property (such as population). To help manage this plurality, Wikidata allows contributors to optionally mark statements as “preferred” (for the most relevant, current statements) or “deprecated” (for irrelevant or unverified statements). Deprecated statements may be useful to Wikidata editors, to record erroneous claims of certain sources or to keep statements that still need to be improved or verified. As

Figure 3. Wikidata in external applications: the “Reasonator” data browser (<http://tools.wmflabs.org/reasonator/>)



with all Wikidata content, these classifications are subject to community-governed editorial processes, similar to those of Wikipedia.¹

Wikidata by the Numbers

Wikidata has grown significantly since its launch in October 2012; see the table here for key facts about its current content. It has also become the most edited Wikimedia project, with 150–500 edits per minute, or a half million per day, about three times as many as the English Wikipedia. Approximately 90% of these edits are made by bots contributors create for automating tasks, yet almost one million edits per month are still made by humans. Figure 2a shows the number of human edits during 14-day intervals. We highlight contributions of power users with more than 10 or even 100,000 edits, respectively, as of February 2014, as they account for most of the variation. The increase in March 2013 marked the official announcement of the site.

Figure 2b shows the growth of Wikidata from its launch until February 2014. There were approximately 14.5 million items and 36 million language links. Essentially, every Wikipedia article is connected to a Wikidata item today, so these numbers grow slowly. In contrast, the number of labels, 45.6 million, as of February 2014, continues to grow; there are more labels than Wikipedia articles. Almost 10 million items have statements, and more

than 30 million statements were created using more than 900 different properties. As expected, property use is skewed; the most frequent property is “instance of” P31 (5.6 million uses) for classifying items; one of the least-frequent properties is P485 (133 uses), which connects a topic (such as Johann Sebastian Bach) with the institution that archives the topic (such as the Bach-Archiv in Leipzig).

The Web of Data

One promising development in Wikidata is the volunteer community’s reuse and integration of external identifiers from existing databases and authority controls, including the International Standard Name Identifier, or ISNI, China Academic Library and Information System, or CALIS, International Air Transport Association, or IATA, MusicBrainz for albums and performers, and North Atlantic Basin’s Hurricane Database, or HURDAT. These external IDs allow applications to integrate Wikidata with data from other sources that remain under the control of the original publisher.

Wikidata is not the first project to reconcile identifiers and authority files from different sources. Others include the Virtual International Authority File, or VIAF, in the bibliographic domain,³ GeoNames in the geographical domain,²² and Freebase.⁷ Wikidata is linked to many of these projects yet also differs in terms

of scope, scale, editorial processes, and author community.

The collected data is exposed in various ways; for example, current per-item exports are available in JSON, XML, RDF, and several other formats. Full database dumps are created at intervals and supplemented by daily diffs. All data is licensed under a Creative Commons CC0 license, thus putting the data in the public domain.

Every Wikidata entity is identified by a unique URI (such as <http://www.wikidata.org/entity/Q42> for item Q42, Douglas Adams). By resolving this URI, tools are able to obtain item data in the requested format (through content negotiation). This follows Linked Data standards for data publication,⁵ making Wikidata part of the Semantic Web^{4,9} while supporting integration of other semantic Web data sources with Wikidata.


Wikidata Applications

The data in Wikidata lends itself to manifold applications on very different levels of data integration.


Language labels and descriptions.

Wikidata provides labels and descriptions for many terms in different languages, possibly using them to present information to international audiences. Unlike common dictionaries, Wikidata covers many named entities (such as for places, chemicals, plants, and specialist terms) that may be very difficult to translate. Many data-centric views can be translated trivially term by term—think maps, shopping lists, and ingredients of dishes on a menu—assuming all items are associated with suitable Wikidata IDs. The open source JavaScript library `qLabel` (<http://google-knowledge.github.io/qLabel/>) provides this functionality for any website.

Identifier reuse. Item IDs can be used as language-independent identifiers to facilitate data exchange and integration across application boundaries. Referring to Wikidata items, applications can provide unambiguous definitions for the terms they use that are also the entry points to a wealth of related information. Wikidata IDs thus resemble digital object identifiers, or DOIs, but emphasize (meta)data beyond online document locations and use another social infrastructure for ID assignment. Wikidata IDs are stable:



Wikidata allows conflicting data to coexist and provides mechanisms to organize this plurality.



IDs do not depend on language labels; items can be deleted, though IDs are never reused; and the links to other datasets and sites further increase stability. Besides providing a large collection of IDs, Wikidata also provides the means to support contributors in selecting the right ID by displaying labels and descriptions; external applications can use the same functionality through the same API.

Accessing Wikidata. The information collected by Wikidata is interesting in its own right, and many applications can be built to access it more conveniently and effectively. Applications created as of early 2014 included generic data browsers like the one in Figure 3 and special-purpose tools, including two genealogy viewers, a tree of life, a table of the elements, and various mapping tools. Applications can use the Wikidata API to browse, query, and even edit data. If simple queries are not enough, a dedicated copy of the data is needed; that copy can be obtained from regular dumps and possibly be updated in real time by mirroring edits on Wikidata. The Wikidata Toolkit, an open source Java library (https://www.mediawiki.org/wiki/Wikidata_Toolkit), provides convenient access to the dumps.

Enriching applications. Many applications can be enriched by embedding information from Wikidata directly into their interfaces; for example, a music player might want to fetch the portrait of the artist just being played in the audio file. Unlike earlier uses of Wikipedia data (such as in Google Maps), application developers need not extract and maintain the data themselves. Such lightweight data access is particularly attractive for mobile apps. In other cases, application developers preprocess data to integrate it into their applications; for example, it would be easy to extract a file of all German cities, together with their regions and post-code ranges, that could then be used in an application. Such derived data can be used and redistributed online or in software under any license, even in commercial contexts.

Advanced analytics. Information in Wikidata can be further analyzed to derive new insights beyond what is already revealed on the surface. An important approach in this regard is logical reasoning, where information

about general relationships is used to derive additional facts; for example, Wikidata's property "grandparent" is obsolete, since its value can be inferred from values of properties "father" and "mother." If an application developer is generally interested in ancestors, then a transitive closure must be computed. Such a closure is relevant for many hierarchical, spatial, and paronomical relations. Other types of advanced analytics include statistical evaluations of both the data and the incidental metadata collected in the system; for example, a researcher can readily analyze article coverage by language,¹² as well as the gender balance of persons described in Wikipedia articles.¹⁴ As in Wikipedia, Wikidata provides plenty of material for researchers to study.

These are only the most obvious approaches to exploiting the data, and many as-yet unforeseen uses should be expected. Wikidata is young, and its data is far from complete. We look forward to new and innovative applications due to Wikidata and its development as a knowledgebase.²³

Prospects

Features still missing include support for complex queries, which is now under development. However, in trying to predict the future of Wikidata, the development team's plans are probably less important than one would expect; for example, the biggest open questions concern the evolution and interplay of the many Wikimedia communities. Will Wikidata earn their trust? How will each of them, with its own language and culture, access, share, and co-evolve the way Wikidata is structured? And how will Wikidata respond to the demands of the communities beyond Wikipedia?


The influence of the volunteer community extends to technical development of the website and its underlying software. Wikidata is based on an open development process that invites contributions, while the site itself provides many extension points for user-created add-ons. The community has designed and developed features (such as article badges for featured articles, image embedding and multi-language editing). The community has also developed ways to enrich the semantics of properties by encoding (soft) con-

straints, as reflected in the guideline "Items should have no more than one birthplace." External tools gather this information, analyze the dataset for constraint violations, and publish the list of violations on Wikidata to allow editors to check if they are valid exceptions or errors.

These aspects of the Wikidata development process illustrate the close relationships among technical infrastructure, editorial processes, and content and the pivotal role the community plays in shaping Wikidata. However, the community is as dynamic as Wikidata itself, based not on status or membership but on the common goal of turning Wikidata into the most accurate, useful, and informative resource possible. This goal promises stability and continuity, even as it allows anyone to take part in defining the future of Wikidata.

Wikipedia is by all accounts one of the most important websites today, a legacy Wikidata must live up to. In only two years, Wikidata is already an important platform for integrating information from many sources. In addition, it also aggregates large amounts of incidental metadata about its own evolution and contribution to Wikipedia. Wikidata thus has the potential to be a major resource for both research and development of new and improved applications. Wikidata, the free knowledgebase anyone can edit, may thus bring us all one step closer to a world that freely shares in the sum of all knowledge.

Acknowledgments

The development team's work on Wikidata is funded through donations by the Allen Institute of Artificial Intelligence, Google, the Gordon and Betty Moore Foundation, and Yandex. Markus Krötzsch's research is supported by the German Research Foundation through the Data Integration and Access by Merging Ontologies and Databases, or DIAMOND, project (Emmy Noether grant KR 4381/1-1). 

References

1. Ayers, P., Matthews, C., and Yates, B. *How Wikipedia Works: And How You Can Be a Part of It*. No Starch Press, San Francisco, CA, 2008.
2. Barrett, D.J. *MediaWiki*. O'Reilly Media, Inc., Sebastopol, CA, 2008.
3. Bennett, R., Hengel-Dittrich, C., O'Neill, E.T., and Tillett, B.B. VIAF (Virtual International Authority File): Linking Die Deutsche Bibliothek and Library of

4. Congress name authority files. In *Proceedings of the World Library and Information Congress 72nd General Conference and Council* (Seoul, South Korea, Aug. 20–24). IFLA, Den Haag, The Netherlands, 2006.
5. Berners-Lee, T., Hendler, J., and Lassila, O. The Semantic Web. *Scientific American* (May 2001), 96–101.
6. Bizer, C., Heath, T., and Berners-Lee, T. Linked data: The story so far. *International Journal on Semantic Web and Information Systems* 5, 3 (2009), 1–22.
7. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., and Hellmann, S. DBpedia: A crystallization point for the Web of Data. *Journal of Web Semantics* 7, 3 (Sept. 2009), 154–165.
8. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the ACM SIGMOD International Conference on Management of Data* (Vancouver, BC, Canada, June 9–12). ACM Press, New York, 2008, 1247–1250.
9. Buneman, P., Cheney, J., Tan, W.-C., and Vansummeren, S. Curated databases. In *Proceedings of the 27th Symposium on Principles of Database Systems*, M. Lenzerini and D. Lembo, Eds. (Vancouver, BC, Canada, June 9–12). ACM Press, New York, 2008, 1–12.
10. Erxleben, F., Günther, M., Krötzsch, M., Mendez, J., and Vrandečić, D. Introducing Wikidata to the Linked Data Web. In *Proceedings of the 13th International Semantic Web Conference* (Trentino, Italy, Oct. 19–23). Springer, Berlin, 2014.
11. Ferrucci, D.A., Brown, E.W., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A., Lally, A., Murdock, J.W., Nyberg, E., Prager, J.M., Schlaefel, N., and Welty, C.A. Building Watson: An overview of the DeepQA project. *AI Magazine* 31, 3 (Fall 2010), 59–79.
12. Guha, R.V., McCool, R., and Fikes, R. Contexts for the Semantic Web. In *Proceedings of the Third International Semantic Web Conference, Vol. 3298 of LNCS*, S.A. McIlraith, D. Plexousakis, and F. van Harmelen, Eds. (Hiroshima, Japan, Nov. 7–11). Springer, Berlin, 2004, 32–46.
13. Hale, S.A. *Multilinguals and Wikipedia Editing*. arXiv:1312.0976 [cs.CY], 2013; <http://arxiv.org/abs/1312.0976>
14. Hoffart, J., Suchanek, F.M., Berberich, K., and Weikum, G. YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. *Artificial Intelligence (Special Issue on Artificial Intelligence, Wikipedia, and Semi-Structured Resources)* 194 (Jan. 2013), 28–61.
15. Klein, M. and Kyriakos, A. VIAFbot and the integration of library data on Wikipedia. *code4lib Journal* 22 (Oct. 2013); <http://journal.code4lib.org/articles/8964>
16. Krötzsch, M., Vrandečić, D., Völkel, M., Haller, H., and Studer, R. Semantic Wikipedia. *Journal of Web Semantics* 5, 4 (Dec. 2007), 251–261.
17. Lenat, D.B. and Guha, R.V. *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project*. Addison-Wesley, Boston, MA, 1989.
18. Leuf, B. and Cunningham, W. *The Wiki Way: Quick Collaboration on the Web*. Addison-Wesley Professional, Boston, MA, 2001.
19. MacGregor, R.M. Representing reified relations in Loom. *Journal of Experimental and Theoretical Artificial Intelligence* 5, 2–3 (1993), 179–183.
20. Moreau, L. The foundations for provenance on the Web. *Foundations and Trends in Web Science* 2, 2–3 (Oct. 2010), 99–241.
21. Noy, N. and Rector, A., Eds. *Defining N-ary Relations on the Semantic Web*. W3C Working Group Note, Apr. 12, 2006; <http://www.w3.org/TR/swbp-n-aryRelations/>
22. Tunstall-Pedoe, W. True Knowledge: Open-domain question answering using structured knowledge and inference. *AI Magazine* 31, 3 (Fall 2010), 80–92.
23. Unxos GmbH. GeoNames (launched 2005); <http://www.geonames.org>
24. Vrandečić, D. The rise of Wikidata. *IEEE Intelligent Systems* 28, 4 (July/Aug. 2013), 90–95.
25. Wolfram Research. Wolfram Alpha (launched 2009); <https://www.wolframalpha.com>

Denny Vrandečić (vrandecic@google.com) is an ontologist at Google, San Francisco, and was project director of Wikidata at Wikimedia Deutschland, Berlin, until September 2013.

Markus Krötzsch (markus.kroetzsch@tu-dresden.de) is lead of the Wikidata data model specification and research group leader at Technische Universität Dresden, Dresden, Germany.

Copyright held by owners/author(s).

New abstractions are critical for achieving SDN goals.

BY MARTIN CASADO, NATE FOSTER, AND ARJUN GUHA

Abstractions for Software-Defined Networks

SOFTWARE-DEFINED NETWORKING (SDN) has received a lot of attention in recent years as a means of addressing some of the long-standing challenges in networking. SDN starts from two simple ideas: generalize network hardware so it provides a standard collection of packet-processing functions instead of a fixed set of narrow features, and decouple the software that controls the network from the devices that implement it. This design makes it possible to evolve the network without having to change the underlying hardware and enables expressing network algorithms in terms of appropriate abstractions for particular applications.

Figure 1 contrasts the architectures of traditional networks and SDN. In SDN, one or more controller machines execute a general-purpose program that responds to events such as changes in network topology, connections initiated by end hosts, shifts in traffic load, or messages from other controllers, by computing

a collection of packet-forwarding rules. The controllers then push these rules to the switches, which implement the required functionality efficiently using specialized hardware.

Because SDN does not specify how controllers are implemented, it can be used to implement a variety of network algorithms, including simple ones such as shortest-path routing, and more sophisticated ones such as traffic engineering.

Many novel applications have been implemented with SDN including policy-based access control, adaptive traffic monitoring, wide-area traffic engineering, network virtualization, and others.^{6,9,16,18-20,44} In principle, it would be possible to implement any of these applications in a traditional network, but it would not be easy: the programmer would have to design new distributed protocols and also address practical issues because traditional switches cannot be easily controlled by third-party programs.

Early SDN controller platforms exposed a rudimentary programming interface that provided little more than a thin wrapper around the features of the underlying hardware. Where there were higher-level abstractions, they reflected structures already found in traditional networks such as topology or link-state information. However, there is now a growing body of work exploring how SDN can change not only which control algorithms can be

» key insights

- **SDN is a new network architecture that decouples the software that controls a network from the devices that implement it.**
- **By providing global visibility into network state, SDN can dramatically simplify the way that many network algorithms are expressed.**
- **SDN also makes it possible to evolve the functionality of a network without having to change the underlying hardware.**
- **SDN is enabling the development of new network programming models, systems abstractions, and verification tools.**



thateasily expressed, but how they can best be written. Just as modern operating systems provide rich abstractions for managing hardware-level resources, we believe that similar abstractions will be needed for networks to fully realize the vision of SDN.

These abstractions are the topic of this article. We review recent and ongoing work on improving SDN programming models and abstractions, focusing on the following areas:

Network-wide structures: SDN controllers are built using relatively small collections of tightly-coupled servers, which makes them amenable to distributed algorithms that maintain consistent versions of network-wide structures such as topology, traffic statistics, and others.

Distributed updates: SDN controllers manage the entire network, so they must often change rules on multiple switches. Update mechanisms that provide consistency guarantees during periods of transition can simplify the development of dynamic programs.

Modular composition: Many network programs naturally decompose into several modules. Controllers that provide compositional programming interfaces make it easy to specify orthogonal aspects of network behavior

in terms of modular components.

Virtualization: Decoupling application logic from the physical topology simplifies programs, ensures isolation, and provides portability. Virtual network abstractions can also provide enhanced scalability and fault tolerance.

Formal verification: To help programmers write correct programs, some controllers provide tools for automatically checking formal properties and diagnosing problems when unexpected errors occur.

Here, we explore these abstractions in further detail. To provide a common basis for discussion, we begin by introducing OpenFlow as a concrete instance of SDN.

OpenFlow

The OpenFlow specification defines a standard collection of features switches must provide, as well as an interface controllers can use to communicate with switches: instructions for installing and deleting forwarding rules, and notifications about flows, topology, and traffic statistics.³¹

An OpenFlow switch maintains a *forwarding table* that contains a list of prioritized *rules*. Each rule has a *pattern* that describes a set of packets and *actions* that describe transfor-

mations on packets. When a packet arrives at a switch, the switch finds a rule whose pattern matches the packet headers and applies the associated actions. If multiple rules match, the switch applies the actions of the highest priority rule, while if no rules match, the switch encapsulates the packet in an OpenFlow message and sends it to the controllers. The controllers can either process the packet directly, or send messages back to the switch instructing it to install or delete rules in its forwarding table. The maximum size of a table is determined by hardware constraints, but most switches have space for at least several thousand rules.

To support traffic monitoring, every rule has associated counters that keep track of basic statistics such as the number and total size of all packets processed with that rule. Controllers can read these counters using OpenFlow messages. They can also configure the physical ports on a switch by creating queues that rate-limit traffic or provide minimum bandwidth guarantees—features that are useful for implementing traffic engineering applications.

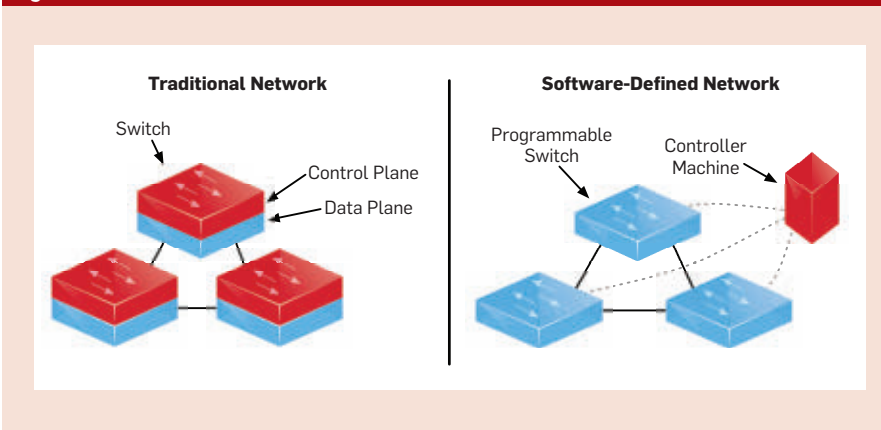
As an example, consider the accompanying table: Read from top to bottom, these rules block all SSH traffic, forward non-SSH traffic destined for hosts 10.0.0.1 and 10.0.0.2 out ports 1 and 2 respectively, and divert all other traffic to the controller for further processing.

Network-wide Structures

A major advantage of SDN is that the controllers can compute network-wide structures that give global visibility into network state, using distributed algorithms that provide strong guarantees about the consistency of these structures across controllers. It would be practically infeasible to maintain these network-wide structures in a traditional network where control is distributed across a larger number of devices, but by using them the logic of many applications can become much simpler. For example, shortest path routing can be implemented by evaluating Dijkstra’s algorithm over the structure representing the topology.⁴¹

Example. To illustrate, consider the task of maintaining a spanning

Figure 1. Traditional and software-defined architectures.



Example of OpenFlow forwarding table.

Priority	Pattern	Action	Counters
30	TcpDstPort = 22	Drop	(7156, 124)
20	IpDstAddr = 10.0.0.1	Forward 1	(2648, 38)
10	IpDstAddr = 10.0.0.2	Forward 2	(14184, 246)
0	*	Controller	(1686, 14)

tree that connects the switches in the network. Such a tree could be used to forward broadcast traffic without any danger of forwarding loops. Designing a distributed algorithm to construct and maintain a spanning tree is surprisingly difficult because it must work correctly in arbitrary topologies and rapidly reconverge to a new tree when events such as unexpected device or link failures occur.

Traditional solution. The classic way to build a spanning tree is to use the *spanning tree protocol*³⁶—a fully distributed protocol, in which the switches periodically exchange information with their neighbors using pairwise announcements. The switches agree on a root node by running a distributed leader election protocol, and then construct the spanning tree incrementally from that node, enabling and disabling links to select the shortest path to the root, and breaking ties using switch identifiers. Note that an implementation of the spanning tree protocol requires neighbor discovery, leader election, as well as the actual tree construction algorithm, but because these components are specific to the protocol, their logic cannot be easily reused by other protocols that require similar functionality. Moreover, when the topology changes, the time to calculate a new tree scales with the size of the longest loop-free path.

SDN solution. Most SDN controllers provide a suite of common functions that arise in many applications such as topology discovery and link fault detection and also maintain structures that keep track of information about the state of the network such as host locations, link capacities, the traffic matrix etc. The database that stores this information is often called a Network Information Base (NIB).²⁵ Using a NIB, an SDN implementation of spanning tree can be dramatically simpler than its distributed counterpart: whenever the topology changes, it simply computes a spanning tree from the topology using Prim’s algorithm, and installs rules on switches that forward along the tree.

Richer applications. By providing programmers with information about the state of the entire network, the NIB also makes it easy to implement richer applications such as traffic engineering that would be difficult to realize in traditional networks.¹¹ For example,



SDN can make many network programs vastly simpler by providing network-wide structures and allowing common distributed programming abstractions to be implemented once and reused across many applications.



the B4 and SWAN systems use SDN to balance load across the wide-area links between datacenters, achieving much higher utilization than was possible with traditional approaches.^{18,19} These applications require distributed controllers that automatically manage data replicated across many controllers through the NIB.²⁶

Using multiple controllers addresses important issues such as scalability and fault tolerance—for example, one controller can take over for another if its load becomes high, or if its links with the switches fail. However, because the number of controllers is typically small, these controllers can use algorithms such as Paxos—something that would not scale in fully distributed settings. Hence, although controllers do use distributed algorithms, they are simpler and often converge faster than traditional protocols since there are fewer controllers than switches.

Discussion. SDN can make many network programs vastly simpler by providing network-wide structures and allowing common distributed programming abstractions to be implemented once and reused across many applications. Such reuse is effectively impossible in traditional networks, where forwarding and control are tightly coupled on each device, implementations of functions such as leader election are tied to specific protocols, and devices have varying CPU, memory, and storage capabilities.

Distributed Updates

In traditional networks, it is often acceptable for configuration updates to be merely eventually consistent. For example, if the network configuration is recalculated due to a link failure, a packet may traverse a switch once in the original state and a second time in the updated state. This can lead to behaviors such as forwarding loops or dropping packets, but since most networks only provide best-effort delivery, as long as the network eventually converges to the new state, transient errors during the transition may be acceptable. However, eventually consistent updates do not always suffice in SDN. For example, an SDN controller might manage filtering rules in addition to forwarding rules, and these rules may be critical for ensuring invariants such


as access control or isolation between the traffic of tenants sharing the network. If configuration updates are propagated to switches in a merely eventually consistent manner, these invariants can easily be violated during periods of transition.

Programmers can sometimes work around these problems by carefully ordering updates so that packets only traverse paths whose configurations have been fully propagated into the network. For example, a programmer might update the ingress switches first, and check that all partially updated paths in the interior of the network are otherwise unreachable during the transition. But calculating orderings manually is complicated and makes updates slow to roll out. Recent work has investigated abstractions that provide general mechanisms for handling distributed updates as well as guarantees ensuring packets never “see” a partially updated path. The idea is to attach versions to configurations and carefully design update protocols that ensure every packet (or set of related packets) is processed by a single consistent version.


Examples. The need for configuration updates that provide strong consistency is a significant departure from traditional networks. To demonstrate they are not only of academic interest, consider the following scenarios:

► **Shortest-path routing:** Initially the network is configured to forward along shortest paths. Then the operator decides to take several switches down for maintenance. The controller generates a new network-wide configuration that forwards along a different set of paths. At all times, the network is expected to provide connectivity and be free of forwarding loops.

► **Distributed access control:** Initially the network is configured to filter a set of “forbidden” packets and otherwise forward along shortest paths. Because the filtering rules are too large to fit into a single forwarding table, the rules are distributed across several switches in the network. The configuration is carefully constructed to ensure each packet traverses the appropriate switches containing the necessary filtering rules. Later, the operator decides to rearrange the rules, maintaining the same policy but placing filtering rules



Beyond the basic abstraction of versioning, the state update subsystem of the controller can expose multiple consistency models to the application.



on different switches. At all times, the network is expected to filter forbidden packets and forward other packets to their destinations.

► **Server load balancing:** Initially the network is configured to redirect incoming requests to several back-end server replicas. At some point, more servers are brought online. The controller then generates a new configuration that balances the load among the new set of servers. At all times, the network is expected to forward incoming traffic to one of the back-end servers while ensuring connection affinity—all packets in a connection should be sent to the same server.

In each of these scenarios, computing the initial and final configurations is straightforward, but transitioning between them while preserving the desired invariants is not. In particular, because the controller lacks the ability to update the state of the entire network atomically, packets traversing the network will necessarily be processed by old, new, or even intermediate configurations containing a mixture of forwarding rules from both configurations.

Update abstractions. Consistent update abstractions allow a controller to update the forwarding state of the entire network while ensuring a packet will never traverse a path that is in transition between two states. The abstractions themselves are straightforward to describe: the controller program specifies the version of the state being pushed into the network and the update subsystem guarantees that each packet traversing the network only “sees” a consistent version of the state. Beyond the basic abstraction of versioning, the state update subsystem of the controller can expose multiple consistency models to the application.

One possible model is per-packet consistency: each packet is processed using a single version of the forwarding state.³⁹ That is, every packet is either processed with the old network-wide configuration, or the new configuration, but not a mixture of the two. Another model is per-flow consistency: every set of related packets is processed using a single configuration version.³⁹ Other extensions consider bandwidth and attempt to avoid creating additional congestion during the transition.^{18,27}

Update mechanisms. A general mechanism for implementing consistent updates is to use a two-phase update. As its name suggests, a two-phase update proceeds in two steps: the controller modifies the new configuration by instrumenting the forwarding rules so they only match packets stamped with a tag corresponding to the new version, and installs it on every switch; the controller updates the rules at the perimeter of the network to stamp packets with the new version tag, and uninstalls the old configuration from every switch. Although the network contains a mixture of rules from the old and new configurations during the transition, these rules have the property that any given packet will be processed according to a single version. Similar mechanisms can be used to implement per-flow consistency.³⁹

In many situations, optimized mechanisms can be used in place of two-phase update. For example, if the update only adds paths, then only rules that impinge on those paths need to be updated. Likewise, if the update only affects a subset of the switches (and the policy has the property that it never forwards traffic across those switches more than once) then the other switches do not need to be updated at all. These optimized mechanisms generate fewer messages, use less rule space on switches, or complete the transition more rapidly than full two-phase update. Consistent updates can also be implemented incrementally²¹ or by diverting some packets to the controller.³¹

Discussion. Updates are a fundamental abstraction for any SDN controller. But despite some promising initial results, many open questions remain. An obvious concern is efficiency: the mechanisms just described require substantial space for rules and a large number of control messages to implement transitions. In large networks, the costs of these mechanisms would be prohibitive. The optimizations discussed here are a good start, but a more comprehensive investigation is needed. Another important issue is the responsiveness of updates. The abstractions described in this section make no guarantees about how long an update will take to complete. For planned changes, this may be acceptable, but when reacting to

failures, a fast response is essential.³⁸ It would be interesting to explore abstractions that trade off weaker guarantees for more responsive update mechanisms. For example, an abstraction that only guarantees packets ultimately reach their final destination and do not traverse loops seems natural, and would admit more efficient implementations. Finally, it may be useful to synthesize updates from application-specific invariants.^{28,35}

Modular Composition

In operating systems, processes allow multiple users to share the available hardware resources on a single machine. Each process is associated with a thread of execution, along with system resources such as memory, locks, file descriptors, and sockets. The operating system requires all interactions between processes take place over well-specified interfaces. For example, memory allocated to one process cannot be tampered with by another, unless it has explicitly been shared by the first process. Although SDN controllers have been compared to “network operating systems,” current controllers lack abstractions analogous to processes.¹³ Instead, most controllers give applications unfettered access to the forwarding tables on every switch in the network, which makes it difficult to write programs in a modular way.

This is unfortunate, because network programming should lend itself naturally to modularization. SDN applications are commonly built out of standard building blocks such as routing, broadcast, monitoring, and access control. However, the lack of modularity in most SDN controllers forces programmers to reimplement these fundamental services from scratch in each new application instead of simply obtaining them from libraries.

Examples. The following scenarios illustrate why modularity can be difficult to achieve in current SDN controllers.

Forwarding and monitoring. The network implements forwarding and traffic monitoring. Because switch tables implement both features, the rules must be carefully crafted to forward and monitor certain packets but only forward or monitor others. If the programmer executes standard forwarding and monitoring programs side-by-side, the

programs may install overlapping rules and the overall behavior of the system will be unpredictable.

Forwarding with isolation. The network is partitioned into two sets of hosts. Each set is isolated from the other, but the network forwards traffic between pairs of hosts in the same set. As with the previous example, the program decomposes into two orthogonal functions: isolation and forwarding. However, the programmer must consider both functions at once as rules generated by one module could easily forward traffic to hosts in the other set, violating the intended policy.

Low-latency video and bulk data transfer. The network provides low-latency service to a videoconferencing application and allows a backup application to forward traffic along several different paths, as long as there is sufficient bandwidth. The programmer must consider both functions simultaneously, to ensure the service-level requirements of each application are met.

Although these examples involve different applications, the problems share a common cause: allowing programs to manipulate low-level network state directly makes it effectively impossible to develop SDN applications in a modular way.

Programming language abstractions. One way to make SDN applications more modular is to change the programming interface they use. Rather than explicitly managing low-level forwarding rules on switches, SDN programmers could use a high-level language that compiles to OpenFlow. Such a language should allow programmers to develop and test modules independently without worrying about unintended interactions. A programmer could even replace a module with another that provides the same functionality.

The NetKAT² language (and its predecessor NetCore^{14,32,33}) provides a collection of high-level programming constructs including operators for composing independent programs. In the first example, the forwarding and monitoring modules could be composed using its union operator, which would yield a module that both forwards and monitors, as desired. The NetKAT compiler takes this policy and generates equivalent forward-

ing rules that can be installed on the switches by its runtime system. The Maple controller⁴³ allows programmers to write modules as packet-processing functions in Java or Haskell and thus use the modularity mechanisms those languages provide. Maple uses a form of runtime tracing to record program decisions and create optimized OpenFlow rules.

Isolated slices. In certain situations, programmers need to ensure the programs being combined will not interfere with each other. For example, in the traffic isolation scenario, the two forwarding modules must be non-interfering.

Combining them using union would be incorrect—the modules might interact by sending packets to each other. One way to guarantee isolation is by using an abstraction that allows multiple programs to execute side-by-side while restricting each to its own isolated “slice” of the network. FlowVisor interposes a hypervisor between the controller and the switches, inspecting each event and control message to ensure the program and its traffic is confined to its own segment of the network.⁴² The FortNOX controller also provides strong isolation between applications, using a framework based on role-based authentication.³⁷ A recent extension to NetKAT also provides a programming construct analogous to slices.^{2,15}

Participatory networking. Combining behaviors from multiple modules sometimes leads to conflicts. For example, if one module reserves all the bandwidth available on a link, other modules will not be able to use that link. The PANE controller¹⁰ allows network administrators to specify module-specific quotas and access control policies on network resources. PANE leverages this mechanism to provide an API that allows end-host applications to request network resources. For example, a videoconferencing application can easily be modified to use the PANE API to reserve bandwidth for a high-quality video call. PANE ensures its bandwidth request does not exceed limits set by the administrator and does not starve other applications of resources.

Discussion. Abstractions for decomposing complex applications

into simple modules are critical technology for SDN. Without them, programmers have to write programs in a monolithic style, developing, testing, and reasoning about the potential interactions between each piece of the program simultaneously. The abstractions provided by high-level languages such as NetKAT and Maple, hypervisors such as FlowVisor and FortNOX, and controllers such as PANE, make it possible to build applications in a modular way. But although these abstractions are a promising first step, much more work is needed. For example, developers need intuitive reasoning principles for establishing properties of programs built out of separate modules—for example, whether one module can be replaced by another without affecting the behavior of the overall program. They also need better ways of expressing and resolving conflicts, especially for properties involving security and resource constraints.

Virtualization

SDN decouples the software that controls the network from the underlying forwarding elements. But it does not decouple the forwarding logic from the underlying physical network topology. This means a program that implements shortest-path routing must maintain a complete representation of the topology and it must recompute paths whenever the topology changes. To address this issue, some SDN controllers now provide primitives for writing applications in terms of virtual network elements. Decoupling programs from topology also creates opportunities for making SDN applications more scalable and fault tolerant.

Examples. As motivation for virtualization, consider these scenarios:

Access control: Access control is typically implemented by encoding information such as MAC or IP addresses into configurations. Unfortunately, this means topology changes such as a host moving from one location to another can undermine security. If access control lists are instead configured in terms of a virtual switch that is connected to each host, then the policy remains stable even if the topology changes.

Multi-tenant datacenter: In datacenters, one often wants to allow multiple tenants to impose different policies on devices in a shared physical network. However, overlapping addresses and services (Ethernet vs. IP) lead to complicated forwarding tables, and it is difficult to guarantee that traffic generated by one tenant will be isolated from other tenants. Using virtual switches, each tenant can be provided with a virtual network they can configure however they like without interfering with other tenants.

Scale-out router: In large networks, it can be necessary to make a collection of physical switches behave like a single logical switch. For example, a large set of low-cost commodity switches could be assembled into a single carrier-grade router. Besides simplifying the forwarding logic for individual applications, this approach can also be used to obtain scalability—because such a router only exists at the logical level, it can be dynamically augmented with additional physical switches as needed.

As these examples show, virtualization can make applications more portable and scalable, by decoupling their forwarding logic from specific physical topologies.

Virtualization abstractions. The most prominent example of a virtual network abstraction for SDN is VMware’s Network Virtualization platform (NSX).^{7,9} The Pyretic controller supports similar abstractions.³³ These controllers expose the same fundamental structure to programmers at the virtual and physical levels—a graph representing the network topology—which allows programs written for the physical network to be used at the virtual level, and vice versa.

To define a virtual network, the programmer specifies a mapping between the elements in the logical network and the elements in the physical network. For example, to create a single “big switch” out of an arbitrary topology, they would map all of the switches in the physical network onto the single virtual switch and hide all internal links.^{7,33}

Virtualization mechanisms. Virtualization abstractions are easy to describe, but their implementations are far from simple. Platforms such as

NSX are based on a controller hypervisor that maps events and control messages at the logical down to the physical level, and vice versa. To streamline the bookkeeping needed to implement virtualization, most platforms stamp incoming packets with a tag (for example, a VLAN tag or MPLS label) that explicitly associates it with one or more virtual networks.

Packet processing in these systems proceeds in several steps. First, the system identifies the logical context of the packet—that is, its location in the virtual network consisting of a switch and a port. Second, it processes the packet according to the policy for its logical context, which relocates the packet into a different logical context (and possibly generates additional packets). Finally, it maps the packet down to the physical level. The hypervisor typically generates physical-level forwarding rules that implement all three steps simultaneously. One challenge concerns the rule space available on physical switches. Depending on the number of virtual networks and the size of their policies, the hypervisor may not be able to accommodate the complete set of rules needed to realize these policies on the switches. Hence, just as in memory management in an ordinary operating system, the hypervisor typically implements a form of “paging,” moving rules onto and off of physical switches dynamically.

Discussion. Virtualization abstractions are an important component of modern SDN controllers. Decoupling programs from the physical topology simplifies applications and also enables sharing the network among several different programs without interference. However, although several production controllers already support virtualization, many open questions remain. One issue concerns the level of detail that should be exposed at the logical level. Current implementations of SDN virtualization provide the same programming interface at the logical and physical levels, eliding resources such as link capacities, queues, and local switch capacity. Another question is how to combine virtualization with other abstractions such as consistent updates. Doing this combination directly is not



Decoupling programs from the physical topology simplifies applications and also enables sharing the network among several different programs without interference.



always possible as both abstractions are commonly implemented using tagging schemes. Finally, current platforms do not support efficient nested virtualization. Semantically there are no deep issues, but there are practical ramifications of implementing nested virtualization using hypervisors.

Formal Verification

Today’s network operators typically work with low-level network configurations by hand. Unsurprisingly, this leads to configuration errors that make many networks unreliable and insecure. By standardizing the interface to network hardware, SDN offers a tremendous opportunity to develop methods and tools that make it much easier to build and operate reliable networks. There are many critical invariants that arise in networks, several of which are described here. These properties can be checked automatically using static or dynamic tools that formally model the state of the network and controller.

Examples. Many network properties are topology-specific, so they can only be stated and verified given a model of the structure of the network.

Connectivity: Packets emitted by any host in the network are eventually delivered to their intended destinations, except possibly due to congestion or failures.

Loop freedom: No packet is ever forwarded along a loop back to a location in the network where it was previously processed with the same headers and contents.

Waypointing: Packets emitted by untrustworthy hosts traverse a middlebox that scans for malicious traffic before being forwarded to their intended destinations.

Bandwidth: The network provides the minimum bandwidth specified in service-level agreements with tenants.

Other properties are either entirely topology-agnostic or hold for large classes of topologies. These properties capture general correctness criteria for applications that are intended to be executed on many different networks:

Access control: The network blocks all traffic emitted by unauthorized hosts, as specified by an access control list.

Host-learning: The controller eventually learns the location of all hosts

and the network forwards packets directly to their intended destinations.

Spanning tree: The network forwards broadcast traffic along a tree that contains every switch (if the network is connected).

Both types of properties have been difficult to establish in traditional networks, as they require reasoning about complex state distributed across many heterogeneous devices. Building on the uniform interfaces provided by SDN, several recent tools have made it possible to verify many network properties automatically.

Verifying configurations. Verifying properties such as loop freedom and connectivity, among others, requires modeling both the topology and switch configurations. Header Space Analysis²³ models switches and the topology as functions in an n -dimensional space, where points represent the vector of packet headers. This model can be used to generate test packets that provide coverage for each rule in the overall configuration⁴⁶ and extensions can check configurations incrementally.²² FlowChecker is based on similar ideas, but encodes policies as binary-decision diagrams.¹ Anteater²⁹ encodes switch configurations as Boolean SAT instances, building on an encoding originally developed by Xie et al.⁴⁵ VeriFlow²⁴ develops domain-specific representations and algorithms for checking properties in real time, which is important because the forwarding behavior of an SDN can rapidly evolve, especially if the controller is reacting to changing network conditions. Finally, NetKAT² includes a sound, complete, and decidable equational reasoning system for proving equivalences between network programs.

Verifying controllers. In addition to tools that can verify properties of configurations, some recent efforts have focused on tools that can verify control programs themselves, often focusing on topology-independent properties. NICE⁵ uses a combination of symbolic execution and model checking to verify several important properties, including the absence of race conditions and bugs akin to switch memory leaks. Another tool developed by Scott et al. checks whether abstractions provided by SDN controllers are correctly realized



There is a tremendous need for tools that can provide rigorous guarantees about the behavior, performance, reliability, and security of networked systems.



in switch-level configurations.¹⁷ Guha et al. describe a framework for establishing controller correctness using a proof assistant, as well as a machine-verified implementation of the NetCore language against a detailed operational model of OpenFlow.¹⁴ VeriCon shows that Hoare-style verification is possible for controllers written as simple imperative programs³ and has been applied successfully to a number of examples adapted from the SDN literature (for example, firewalls, routing algorithms, and so on). Nelson, et al. present a Datalog-based SDN programming language, called Flowlog, that they also use to write and verify several canonical properties.³⁴ Because Flowlog is designed to be finite-state, it is amenable to automatic verification without the need for complex programmer-supplied assertions.

Discussion. There is a tremendous need for tools that can provide rigorous guarantees about the behavior, performance, reliability, and security of networked systems. By standardizing the interfaces for controlling networks, SDN makes it feasible to build tools for verifying configurations and controllers against precise formal models. Some possible next steps in this area include developing custom logics and decision procedures for expressing and checking properties, enriching models with additional features such as latency and bandwidth, and better integrating property checking and debugging tools into SDN controller platforms.

Related Work

Enormous momentum has gathered behind SDN in recent years, but the ideas behind SDN build on many previous efforts. Tempest,⁴⁰ an architecture developed at Cambridge in the mid-1990s, was an early attempt to decouple forwarding and control in the context of ATM networks. Several features from Tempest can be found in SDN today including an emphasis on open interfaces and support for virtualization. Similarly, the IETF ForCES working group defined a standard protocol that a controller could use to manage multiple heterogeneous devices in a single network.⁸ The Soft-Router project


explored the benefits of separating forwarding and control in terms of extensibility, scalability, reliability, security, and cost.²⁶

The Routing Control Platform,⁴ developed at AT&T, demonstrated that logical centralization could be used to dramatically simplify routing algorithms while still providing good performance. These ideas were later expanded in the 4D platform,¹² which introduced the distinction between management and control planes. The benefits of expressing algorithms using network-wide data structures instead of using distributed algorithms in SDN can also be seen in this work.

The most immediate predecessor of SDN was Ethane,⁶ a system aimed at providing fine-grained in-network access control. Ethane provided a high-level language for defining security policies, and a controller program that implemented those policies by installing and uninstalling custom forwarding rules in programmable network switches. The NOX controller was based on Ethane,¹³ and the protocol used by the Ethane controller to communicate with switches later evolved into the first version of the OpenFlow standard.³¹

Conclusion

Many of the initial efforts around SDN have focused on architectural concerns—making it possible to evolve the network and develop rich applications. But the growth of this new software ecosystem has also led to the development of fundamental new abstractions that exploit the ability to write network control software on standard servers with a less constrained state distribution model. We believe these abstractions are critical for achieving the goals of SDN and may prove to be some of its most lasting legacies.

Acknowledgments. The authors wish to thank Shrutarshi Basu, Andrew Ferguson, Anil Madhavapeddy, mark Reitblatt, Jennifer Rexford, Mooly Sagiv, Steffen Smolka, Robert Soulé, David Walker, and *Communications'* reviewers for helpful comments. Our work is supported by NSF grant CNS-1111698, ONR award N00014-12-1-0757, a Sloan Research Fellowship, and a Google Research Award. 

References

- Al-Shaer, E. and Al-Haj, S. FlowChecker: Configuration analysis and verification of federated OpenFlow infrastructures. In *Proceedings of SafeConfig*, 2010.
- Anderson, C.J., Foster, N. Guha, A., Jeannin, J-B, Kozen, D., Schlesinger, C. and Walker, D. NetKAT: Semantic foundations for networks. In *Proceedings of POPL*, 2014.
- Ball, T., Bjørner, N., Gember, A., Itzhaky, S., Karbyshev, A., Sagiv, M., Schapira, M. and Valadarsky, A. VeriCon: Towards verifying controller programs in software-defined networks. In *Proceedings of PLDI*, 2014.
- Caesar, M., Caldwell, D.F., Feamster, N., Rexford, J., Shaikh, A. and van der Merwe, J.E. Design and implementation of a routing control platform. In *Proceedings of NSDI*, 2005.
- Canini, M., Venzano, D., Perešini, P., Kostić, D. and Rexford, J. A NICE way to test OpenFlow applications. In *Proceedings of NSDI*, 2012.
- Casado, M., Freedman, M.J., Pettit, J., Luo, J., McKeown, N. and Shenker, S. Ethane: Taking control of the enterprise. In *Proceedings of SIGCOMM*, 2007.
- Casado, M., Koponen, T., Ramanathan, R. and Shenker, S. Virtualizing the network forwarding plane. In *Proceedings of PRESTO*, 2010.
- Doria, A., Hadi Salim, J., Haas, R., Khosravi, H., Wang, W. Dong, L., Gopal, R. and Halpern, J. Forwarding and control element separation (ForCES). 2010. IETF RFC 5810.
- Koponen, T. et al. Network virtualization in multi-tenant datacenters. In *Proceedings of NSDI*, 2014.
- Ferguson, A.D., Guha, A., Liang, C., Fonseca, R. and Krishnamurthi, S. Participatory Networking: An API for application control of SDNs. In *Proceedings of SIGCOMM*, 2013.
- Fortz, B., Rexford, J. and Thorup, M. Traffic engineering with traditional IP routing protocols. *IEEE Commun.* (Oct. 2002).
- Greenberg, A.G., Hjálmtýsson, G., Maltz, D.A., Myers, A., Rexford, J., Xie, G.G., Yan, H., Zhan, J. and Zhang, H. A clean slate 4D approach to network control and management. *SIGCOMM CCR* 35, 5 (2005).
- Gude, N., Koponen, T., Pettit, J., Pfaff, B., Casado, M., McKeown, N. and Shenker, S. NOX: Towards an operating system for networks. *ACM SIGCOMM CCR* 38, 3 (2008).
- Guha, A., Reitblatt, M. and Foster, N. Machine-verified network controllers. In *Proceedings of PLDI*, 2013.
- Gutz, S., Story, A., Schlesinger, C. and Foster, N. Splendid isolation: A slice abstraction for software-defined networks. In *Proceedings of HotSDN*, 2012.
- Heller, B., Seetharaman, S., Mahadevan, P., Yakoumīs, Y., Sharma, P., Banerjee, S. and McKeown, N. ElasticTree: Saving energy in datacenter networks. In *Proceedings of NSDI*, 2010.
- Heller, B. et al. Leveraging SDN layering to systematically troubleshoot networks. In *Proceedings of HotSDN*, 2013.
- Hong, C-Y, Kandula, S., Mahajan, R., Zhang, M., Gill, V., Nanduri, M. and Wattenhofer, R. Achieving high utilization with software-driven WAN. In *Proceedings of SIGCOMM*, 2013.
- Jain, S. et al. B4: Experience with a globally deployed software defined WAN. In *Proceedings of SIGCOMM*, 2013.
- Jose, L., Yu, M. and Rexford, J. Online measurement of large traffic aggregates on commodity switches. In *Proceedings of HotICE*, 2011.
- Katta, N.P., Rexford, J., and Walker, D. Incremental consistent updates. In *Proceedings of HotSDN*, 2013.
- Kazemian, P., Chang, M., Zeng, H., Varghese, G., McKeown, N. and Whyte, S. Real-time network policy checking using Header Space Analysis. In *Proceedings of NSDI*, 2013.
- Kazemian, P., Varghese, G. and McKeown, N. Header space analysis: Static checking for networks. In *Proceedings of NSDI*, 2012.
- Khurshid, A., Zhou, W., Caesar, M. and Godfrey, B. VeriFlow: Verifying network-wide invariants in real time. In *Proceedings of NSDI*, 2013.
- Koponen, T., Casado, M., Gude, N., Stribling, J., Poutievski, L., Zhu, M., Ramanathan, R., Iwata, Y., Inoue, H., Hama, T. and Shenker, S. Onix: A distributed control platform for large-scale production networks. In *Proceedings of OSDI*, 2010.
- Lakshman, T.V., Nandagopal, T., Ramjee, R., Sabnani, K. and Woo, T. The SoftRouter architecture. In *Proceedings of HotNets*, 2004.
- Liu, H.H., Wu, X., Zhang, M., Yuan, L., Wattenhofer, R. and Maltz, D. zUpdate: Updating datacenter networks with zero loss. In *Proceedings of SIGCOMM*, 2013.
- Mahajan, R. and Wattenhofer, R. On consistent updates in software-defined networks. In *Proceedings of HotNets*, 2013.
- Mai, H., Khurshid, A., Agarwal, R., Caesar, M., Godfrey, B. and King, S.T. Debugging the data plane with Anteater. In *Proceedings of SIGCOMM*, 2011.
- McGeer, R. A safe, efficient update protocol for OpenFlow networks. In *Proceedings of HotSDN*, 2012.
- McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S. and Turner, J. OpenFlow: Enabling innovation in campus networks. *SIGCOMM CCR* 38, 2 (2008).
- Monsanto, C., Foster, N., Harrison, R. and Walker, D. A compiler and run-time system for network programming languages. In *Proceedings of POPL*, 2012.
- Monsanto, C., Reich, J., Foster, N., Rexford, J. and Walker, D. Composing software defined networks. In *Proceedings of NSDI*, 2013.
- Nelson, T., Ferguson, A., Scheer, M., and Krishnamurthi, S. Tierless programming and reasoning for software-defined networks. In *Proceedings of NSDI*, 2014.
- Noyes, A., Warszawski, T., Cerny, P. and Foster, N. Toward synthesis of network updates. In *Proceedings of SYNT*, 2013.
- Perlman, R. An algorithm for distributed computation of a spanning tree in an extended LAN. *SIGCOMM CCR* 15, 4 (1985).
- Porras, P., Shin, S., Yegneswaran, V., Fong, M., Tyson, M. and Gu, G. A security enforcement kernel for OpenFlow networks. In *Proceedings of HotSDN*, 2012.
- Reitblatt, M., Canini, M., Foster, N. and Guha, A. Fattire: Declarative fault-tolerance for software-defined networks. In *Proceedings of HotSDN*, 2013.
- Reitblatt, M., Foster, N., Rexford, J., Schlesinger, C. and Walker, D. Abstractions for network update. In *Proceedings of SIGCOMM*, 2012.
- Rooney, S., van der Merwe, J.E., Crosby, S.A. and Leslie, I.M. The Tempest: A framework for safe, resource assured, programmable networks. *IEEE Commun.* 36, 10 (1998).
- Shenker, S., Casado, M., Koponen, T. and McKeown, N. The future of networking and the past of protocols. Invited talk at Open Networking Summit, Oct. 2011.
- Sherwood, R. et al. Carving research slices out of your production networks with OpenFlow. *SIGCOMM CCR* 40, 1 (2010).
- Voellmy, A., Wang, J., Yang, Y.R., Ford, B. and Hudak, P. Maple: Simplifying SDN programming using algorithmic policies. In *Proceedings of SIGCOMM*, 2013.
- Wang, R., Butnariu, D. and Rexford, J. OpenFlow-based server load balancing gone wild. In *Proceedings of HotICE*, 2011.
- Xie, G.G., Zhan, J., Maltz, D.A., Zhang, H., Greenberg, A.G., Hjálmtýsson, G. and Rexford, J. On static reachability analysis of IP networks. In *Proceedings of INFOCOM*, 2005.
- Zeng, H., Kazemian, P., Varghese, G. and McKeown, N. Automatic test packet generation. In *Proceedings of CoNext*, 2012.

Martin Casado (mcasado@vmware.com) is fellow, senior vice president and general manager of networking and security at VMware and was the co-founder and CTO of Nicira Networks, Palo Alto, CA.

Nate Foster (nfoster@cs.cornell.edu) is an assistant professor of computer science at Cornell University, Ithaca, NY.

Ajun Guha (arjun@cs.umass.edu) is an assistant professor of computer science at the University of Massachusetts, Amherst, MA.

research highlights

P. 97

**Technical
Perspective**
**Attacking a Problem
from the Middle**

By Bart Preneel

P. 98

**Dissection: A New Paradigm
for Solving Bicomposite
Search Problems**

By Itai Dinur, Orr Dunkelman, Nathan Keller, and Adi Shamir

Technical Perspective

Attacking a Problem from the Middle

By Bart Preneel

THE FOLLOWING PAPER presents an elegant new algorithm for solving a broad class of combinatorial optimization problems. The idea of the dissection technique came up by studying strengthened versions of the DES algorithm. DES is a cryptographic algorithm proposed by the National Bureau of Standards (now known as the National Institute for Standards and Technology, or NIST) in 1977 for the protection of sensitive but unclassified U.S. government data. For almost three decades, DES has been a worldwide de facto standard for encryption. IBM may have designed DES, but the U.S. National Security Agency (NSA) decided the key length should be restricted to 56 bits. This was a controversial move, as it would allow the NSA (and no one else) to break the cipher by searching the key space.

A straightforward way to increase the key length seems to encrypt twice, known as double-DES or $C = \text{DES}_{K_1}(\text{DES}_{K_2}(P))$, where P and C are the plaintext and ciphertext and K_1 and K_2 are two different k -bit keys. In 1977, Diffie and Hellman demonstrated this method does not help. Let us assume a few ciphertexts and corresponding plaintexts are known (this is a standard assumption in cryptography, one that is very often met in practice). The meet-in-the-middle attack works as follows: one tries all 2^k values of the first key K_1 , computes the values in the middle $A = \text{DES}_{K_1}(P)$ and stores the pairs $(K_1, A(K_1))$ in a table. In a second step, one computes for all keys K_2 the value in the middle as $A' = \text{DES}_{K_2}^{-1}(C)$ and searches for the value A in the table. If a match


is found, the candidate pair (K_1, K_2) is checked with an additional plaintext and ciphertext. This algorithm requires about 2^k encryptions and 2^k memory. This is the reason why the financial sector has upgraded single DES to triple-DES, with three encryptions. A meet-in-the-middle attack on triple-DES requires about 2^{2k} encryptions and 2^k memory.

The authors of this paper ask a natural question: Would one get even better security with quadruple-DES? At first sight one would conclude that breaking quadruple DES requires 2^{2k} encryptions and 2^{2k} memory. But the rather surprising answer is the cost can be reduced to that of triple-DES by applying recursion, that is, guessing first the value in the middle and subsequently performing two meet-in-the-middle attacks. This idea was proposed by Zhu and Gong for the block cipher KATAN and the authors were inspired by the work of Isobe on the block cipher GOST; however,

The authors of the following paper ask a natural question: Would one get even better security with quadruple-DES?

the authors push these ideas much further in their Crypto 2012 paper by optimizing and generalizing this algorithm for the case of n -fold encryption for any value of n . Moreover, they show that the dissection technique has many applications.

This paper describes dissection for two examples. The first is the solution of Rubik's cube. The dissection solution has the same complexity as an earlier algorithm, but without using group-theoretic properties. The second example is the combinatorial partition problem; a variant of this problem (the knapsack problem) has played a very important role in cryptography for the construction of public-key encryption and hash functions. For this example a more complex strategy is used that divides the problem into uneven parts.

The dissection technique is remarkable, as it starts from a very simple idea, the meet-in-the-middle attack. The authors show this idea can be generalized in an elegant way and that many clever optimizations can push the idea to its limits. A key contribution of this work is the realization that dissection is broadly applicable to combinatorial optimization problems. Finally, the proposed algorithms reduce memory requirements, which is essential to make the algorithms practical. 

Bart Preneel (bart.preneel@esat.kuleuven.be) is a professor in the COSIC research group of the Electrical Engineering Department at the Katholieke Universiteit Leuven, Belgium.

Copyright held by author.

Dissection: A New Paradigm for Solving Bicomposite Search Problems

By Itai Dinur, Orr Dunkelman, Nathan Keller, and Adi Shamir

Abstract

Combinatorial search problems are usually described by a collection of possible states, a list of possible actions which map each current state into some next state, and a pair of initial and final states. The algorithmic problem is to find a sequence of actions which maps the given initial state into the desired final state. In this paper, we introduce the new notion of *bicomposite search problems*, and show that they can be solved with improved combinations of time and space complexities by using a new algorithmic paradigm called *dissection*. To demonstrate the broad applicability of our new paradigm, we show how to use it in order to untangle Rubik's cube and to solve a typical NP-complete partition problem with algorithms which are better than any previously described algorithm for these problems.

1. INTRODUCTION

A central problem in the design of efficient algorithms is how to solve search problems, in which we are given a pair of states and a collection of possible actions, and we are asked to find how to get from the first state to the second state by performing some sequence of actions. In some cases, we only want to decide whether such a sequence exists at all, while in other cases it is clear that such sequences exist but we are asked to find the shortest possible sequence. Many search problems of this type have associated decision problems which are NP-complete, and thus we do not expect to find any polynomial time algorithms which can solve all their instances. However, what we hope to find are new exponential time algorithms whose exponents are smaller than in the best previously known algorithms. For example, the problem of breaking a cryptographic scheme whose key has $n = 100$ unknown bits cannot be solved in a practical amount of time via an exhaustive key search algorithm, since its time complexity of 2^n would be beyond reach even for the largest currently available data center. However, if we manage to find a better cryptanalytic attack whose running time is $2^{n/2}$, we can break the scheme with a modest effort in spite of the exponential nature of this complexity function. One trick which is often helpful in such situations is to find a tradeoff between the time and space complexities of the attack: Exhaustive search requires a lot of time but a negligible amount of memory, and thus a tradeoff which uses more memory (in the form of large tables of precomputed values) in order to reduce the time (by skipping many computational steps) will be very beneficial. For reasons which are

explained in the extended version of this paper (available in Dinur et al.²), we usually consider the product of the amount of time and the amount of space required by the algorithm as the appropriate complexity measure that we try to minimize. In the example above, breaking the cryptosystem with $T = 2^n$ time and an $S = 1$ space is infeasible, breaking it with $T = 2^{2n/3}$ time and $S = 2^{n/3}$ space (whose product $TS = 2^n$ is the same as before) is better but still barely feasible, and breaking it in $T = 2^{n/2}$ time and $S = 2^{n/4}$ space (whose product $TS = 2^{3n/4}$ has a smaller exponent) is completely feasible.

A typical search problem is defined by a condition F (e.g., in the form of a CNF Boolean formula which is a conjunction of clauses) and a candidate solution X (e.g., in the form of a 0/1 assignment to all the variables in F), and the goal is to find among all the possible X at least one that satisfies the condition that $F(X)$ is true. Such a representation has no internal structure in the sense that it uses the full description of F and the full value of X in order to decide whether $F(X)$ is satisfied. However, we can usually replace the all-or-nothing choice of X by a sequence of smaller decisions. For example, we can start with the assignment of 0 to all the variables, and at each stage we can decide to flip the current value of one of the Boolean variables. At any intermediate point in this process F is in a state in which some of its clauses are satisfied and some are not, and our goal is to reach a state in which all the clauses are simultaneously satisfied. More generally, we say that a search problem is *composite* if solving it can be described by a sequence of atomic actions, which change the system from some initial state through a series of intermediate states until it reaches some final desired state. Most search problems have such a composite structure, which can be represented by the *execution matrix* described in Figure 1. The rows of this matrix represent states S_0, S_1, \dots , and the solution X is represented by the sequence of actions a_1, a_2, \dots on the left side of the figure in which action a_i changes state S_{i-1} to state S_i . In many cases, the atomic actions are invertible operations over the states, which makes it possible to map S_{i-1} to S_i by using the action a_i , and to map S_i back to S_{i-1} by using the action a_i^{-1} . For example, in the case of Boolean formulas we are allowed to flip the current value of any one of the variables in X , but we can cancel its effect by flipping the same variable a second time,

The previous version of this paper was published in *CRYPTO*, vol. 7417 (2012) *Lecture Notes in Computer Science*. Springer, 719–740.

so in this case the action and its inverse happen to be the same. In this paper we consider only such invertible cases, which allow to split the search problem by applying some forward actions to the initial state, applying some inverse actions to the final state, and searching for a meet-in-the-middle (MITM) state which combines these parts. Our new dissection technique can be applied even when some of the actions are not invertible, but this makes its description more complicated and its complexity slightly higher, as discussed in Dinur et al.²

So far we have partitioned the execution matrix into multiple rows by dividing the solution process into a series of atomic actions. In order to apply our new dissection technique, we also have to partition the execution matrix into multiple columns by dividing each state S_i into smaller chunks $S_{i,j}$ which we call substates, as described in Figure 2. However, only partitions in which the substates can be manipulated independently of each other will be useful to us. We say that a search problem has a *bicomposite structure* if it can be described by an execution matrix in which the knowledge of the action a_i makes it possible to uniquely determine substate $S_{i,j}$ from $S_{i-1,j}$ and $S_{i-1,j}$ from $S_{i,j}$, even when we know nothing about the other substates in the matrix. This immediately implies that if we choose any rectangle of any dimensions within the execution matrix, knowledge of the substates $S_{i-1,j}, S_{i-1,j+1}, \dots, S_{i-1,k}$ along its top edge and knowledge of the actions $a_i, a_{i+1}, \dots, a_\ell$ to its left

suffices in order to compute the substates $S_{\ell,j}, S_{\ell,j+1}, \dots, S_{\ell,k}$ along its bottom edge, and vice versa.

Not every search problem has such a bicomposite representation, but as we demonstrate in this paper there are many well-known problems which can be represented in such a way. When a problem is bicomposite, we can solve it with improved efficiency by using our generic new technique. Our main observation is that in such cases, we can improve the standard MITM algorithms (which try to match the forward and backward directions at a full intermediate state) by considering algorithms which only partially match the two directions at a partially specified intermediate state. In addition, we can reverse the logic of MITM algorithms, and instead of trying to converge from both ends of the execution toward some intermediate state which happens to be the same, we can start by partially guessing this intermediate state in all possible ways, and for each guessed value we can break the problem into two independent subproblems by proceeding from this intermediate state toward the two ends (exploiting the fact that in bicomposite problems we can determine the effect of long sequences of actions on partially specified states). We can then solve each one of these subproblems recursively by partially guessing additional substates in the execution matrix. We call this approach a *dissection* algorithm since it resembles the process in which a surgeon makes a sequence of short cuts at various strategically chosen locations in the patient's body in order to carry out the operation. For example, in Section 4 we show how to solve the well-known combinatorial partition problem by first guessing only two-sevenths of the bits of the state which occur after performing three-sevenths of the actions, and then solving one of the resultant subproblems by guessing in addition one-seventh of the bits of the state which occurs after performing five-sevenths of the actions.

Our main purpose in this paper is to introduce the new ideas by applying them in the simplest possible way to several well-known search problems. We intentionally overlook several nuisance issues whose proper handling is not easy to explain, and ignore several possible optimizations which can further reduce the complexity of our algorithms. When we analyze the running time of our algorithms, we often assume for the sake of clarity that the instance we try to solve is randomly chosen, and that the intermediate states we try to guess are uniformly distributed.

The paper is organized as follows. In Section 2 we describe the problem of solving Rubik's cube with the smallest possible number of steps as a bicomposite search problem, and in Section 3 we show how to solve it with time complexity which is approximately the square root of the size of the search space, and a space complexity which is approximately the fourth root of the size of the search space by using the simplest version of the new dissection algorithm. In Section 4 we describe several improvements of the basic dissection technique, and show how to use them in order to solve a different search problem called the "combinatorial partition problem" with combinations of time and space complexities which could not be achieved by any previously published algorithm, and which are more suitable for large

Figure 1. An execution matrix of a composite search problem.

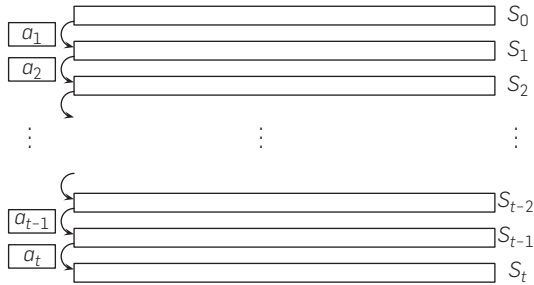
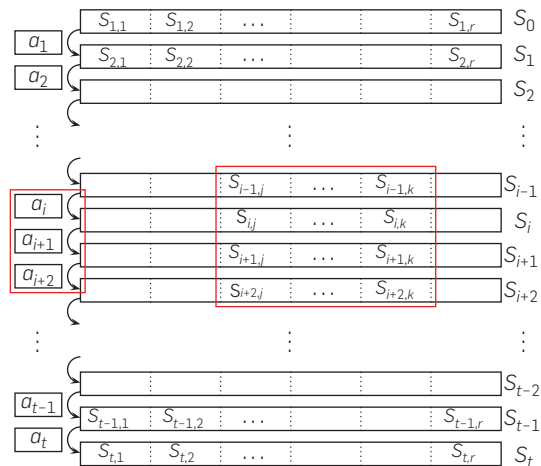


Figure 2. An execution matrix of a bicomposite search problem.



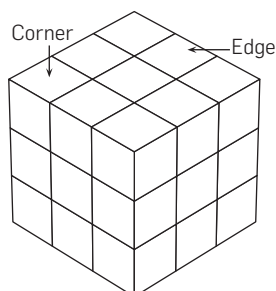
scale FPGA-based hardware. We conclude the paper with some remarks in Section 5.

2. REPRESENTING RUBIK'S CUBE AS A BICOMPOSITE SEARCH PROBLEM

In this section we show how to construct a bicomposite representation of the well-known problem of solving a standard $3 \times 3 \times 3$ Rubik's cube.⁶ We can assume that we always hold the cube in a fixed orientation, in which the white center color is at the top, the yellow center color is on the left, etc. One of the 27 subcubes is at the center of the cube, and we can ignore it since it is completely invisible. The six subcubes at the center of each face are not moved when we rotate that (or any other) face, and thus we can ignore them as well in our state representation. The actions we can take are to rotate each one of the six faces of the cube by 90, 180, or 270 degrees (we are not allowed to rotate a center slice since this will change the standard orientation of the cube defined above). Consequently, we have a repertoire of 18 atomic actions we can apply to each state of the cube. Note that all these actions are invertible mappings on the state of Rubik's cube in the sense that the inverse of a 90 degree rotation is a 270 degree rotation applied to the same face, and both of them are available as atomic actions.

Among the $27 - 6 - 1 = 20$ subcubes which we can move, 12 have two visible colors and are called edge subcubes, and 8 have three visible colors and are called corner subcubes (Figure 3). Each such subcube can be uniquely described by the combination of colors on it, such as a blue-white (BW) edge subcube or a green-orange-red (GOR) corner subcube. In addition, each location on the cube can be described by its relevant sides (i.e., a combination of top/bottom, left/right, front/back). We can thus describe any state of the cube by a vector of length 20, whose i th entry describes the current location of the i th subcube (e.g., the first entry in the vector will always refer to the blue-green edge subcube, and specify that it is currently located at the top-front position). To complete the specification, we also have to choose some standard orientation of the colors, and note that edge subcubes can be either in the standard (e.g., BW) or in an inverted (e.g., WB) state, and each corner subcube can be in one of three possible orientations (e.g., GOR, ORG, or RGO). Note that any possible action can only move edge subcubes to edge subcubes and corner subcubes to corner subcubes. If we use the first 12 positions in the state vector to describe the current locations of the 12 edge subcubes

Figure 3. Rubik's cube.



(in some fixed order), then each entry in these positions can be described by a number between 1 and 24 (specifying in which one of the 12 possible positions it is currently located and in which one of its 2 possible orientations). Similarly, when we use the last 8 positions in the state vector to describe the current locations of the 8 corner subcubes (in some fixed order), then each one of these entries can again contain a number between 1 and 24, this time specifying in which one of the 8 possible positions it is located and in which of its 3 possible orientations. We can thus describe any state of Rubik's cube by a vector of length 20 whose entries are numbers between 1 and 24. Any one of the 18 atomic actions will change 8 of the entries in this vector, by moving 4 edge subcubes and 4 corner subcubes to new positions and orientations, leaving the remaining 12 entries unchanged.

The problem of solving Rubik's cube can now be formalized as the following search problem: We are given a first vector of length 20 (representing the initial scrambled state of the cube) and a second vector of length 20 (representing the standard unscrambled state of the cube). We would like to find a sequence of atomic actions (in the form of face rotations) that will change the first vector into the second vector. Finding *some* sequence is easy, and there are many algorithms which are described in the recreational mathematics literature to achieve this (some of which are described in Slocum⁶). However, these algorithms are typically quite wasteful, using between 50 and 100 actions in order to move subcubes to their correct positions one at a time in some fixed order, ignoring the effect of these actions on other subcubes. Some algorithms use fewer actions, but then they are much harder to describe since they require a very detailed case analysis of the full state before choosing the first action. Recently, Davidson et al.¹ proved that 20 actions are necessary and sufficient if we want to solve *any* solvable state of Rubik's cube, but this was an existential result and the authors did not present an efficient way to actually find such a sequence. Note that the number of possible sequences of 20 atomic actions is $18^{20} = 12748236216396078174437376 \approx 2^{83}$, but we can slightly reduce the size of the search space to $18 \times 15^{19} = 399030807609558105468750 \approx 2^{78}$ by noticing that there is no point in rotating the same face twice in a row. However, even this reduced size cannot be exhaustively searched in a feasible amount of time.

To show that our representation of the search problem is bicomposite, assume that we know the current location and orientation of a particular subcube (namely, we know the value of $S_{i-1,j}$ in the execution matrix as a number between 1 and 24), and we apply to it some known face rotation action. We can then uniquely determine the new location and orientation of that particular subcube (namely, the value of $S_{i,j}$) even when we know nothing about the current location and orientation of any other subcube (namely, all the other $S_{k,\ell}$ values in the execution matrix). Notice that many other natural representations of the states of Rubik's cube do not have such a bicomposite structure. For example, if we associate the first entry in the state vector with a particular cube position (such as top-front) and use it to denote which edge subcube (such as BW) is currently

located in it and in which orientation, then knowledge of just this entry in the first state does not tell us anything about which edge subcube (such as GR) replaces it at the top-front position if we rotate the top face by 90 degrees. Such a representation requires knowledge of other columns in the execution matrix, depending on which action was applied to the state, and thus we cannot use it in our new dissection technique.

As shown in Section 3, we can use the bicomposite representation in order to find for any given initial state of Rubik's cube a sequence of up to 20 face rotations using a completely feasible combination of a time complexity which is the square root of the size of the search space (namely, in about 2^{39} steps, or a few minutes on a standard PC) and a space complexity which is about the fourth root of this number (namely, about $2^{19.5}$ memory locations, or a few megabytes). The resultant algorithm is completely generic, makes no use of the details of the problem besides its bicompositeness, and matches the complexities of the best previous algorithm for solving Rubik's cube (designed about 25 years ago, see Fiat et al.³) which was highly specific and depended on the group-theoretic properties of the set of permutations defined by Rubik's cube.

3. THE BASIC DISSECTION TECHNIQUES

We now assume that we are given an initial state vector S_0 and a final state vector S_ℓ of Rubik's cube, and our goal is to find a series of atomic actions a_1, a_2, \dots, a_ℓ that transform the initial state into the final state. As described in the previous section, we know that $\ell = 20$ suffices to find a solution, and hence our goal is to find a_1, a_2, \dots, a_{20} .

Our dissection algorithms are extensions of the classical MITM algorithm, which was first presented in 1974 by Horowitz and Sahni⁵ in order to solve the Knapsack problem. We can apply a MITM algorithm to almost any composite problem with invertible actions. When the size of the search space is 2^n , the MITM algorithm requires about $2^{n/2}$ time and $2^{n/2}$ space. For the sake of completeness, we describe below how to apply this algorithm in the context of Rubik's cube, whose search space has about 2^{78} states. In this case, a time complexity of $2^{78/2} = 2^{39}$ is feasible, but a space complexity of $2^{78/2} = 2^{39}$ random access memory locations is too expensive.

The complete details of the algorithm are given in Figure 4. The first step of the algorithm is to iterate over all possible $20/2 = 10$ action sequences a_1, \dots, a_{10} . We have $18 \times 15^9 \approx 2^{39}$ such sequences, and for each one, we apply its actions to S_0 and obtain a possible value for S_{10} . We store the 2^{39} values of a_1, \dots, a_{10} in a list next to the corresponding value of S_{10} , and sort⁶ the list according to S_{10} . Next, we iterate over all possible action vectors a_{11}, \dots, a_{20} . Again, there are about 2^{39} such action vectors, and for each one, we apply its inverted actions to S_{20} and obtain a possible value for S_{10} . We now search the sorted list for this value of S_{10} , and for each match, we obtain the corresponding value of a_1, \dots, a_{10} from the list and output a_1, a_2, \dots, a_{20} as a solution.

⁶ For the sake of simplicity, we ignore logarithmic factors in our complexity analysis, and thus we assume that sorting is a linear time operation.

Figure 4. Meet-in-the-middle procedure to solve the Rubik's cube.

Algorithm MITM-Rubik

```

Input: Initial state  $S_0$  and final state  $S_{20}$ 
for all  $a_1, a_2, \dots, a_{10}$  do
  Compute  $S_{10} = a_{10}(\dots(a_2(a_1(S_0))\dots))$ 
  Store  $(S_{10}, a_1, a_2, \dots, a_{10})$  in a list  $L$ 
Sort  $L$  according to the value of  $S_{10}$  in each entry (under
some lexicographical order)
for all  $a_{11}, a_{12}, \dots, a_{20}$  do
  Compute  $S_{10} = a_{11}^{-1}(\dots(a_{19}^{-1}(a_{20}^{-1}(S_{20}))\dots))$ 
  Search for  $S_{10}$  in  $L$ 
if  $S_{10}$  is found then
  return the associated  $a_1, a_2, \dots, a_{10}$  and  $a_{11}, a_{12}, \dots, a_{20}$ 
as a solution

```

The MITM algorithm requires about 2^{39} memory cells in order to store the sorted list, and its time complexity is about 2^{39} , which is the time required in order to iterate over each one of the action vectors a_1, \dots, a_{10} and a_{11}, \dots, a_{20} .

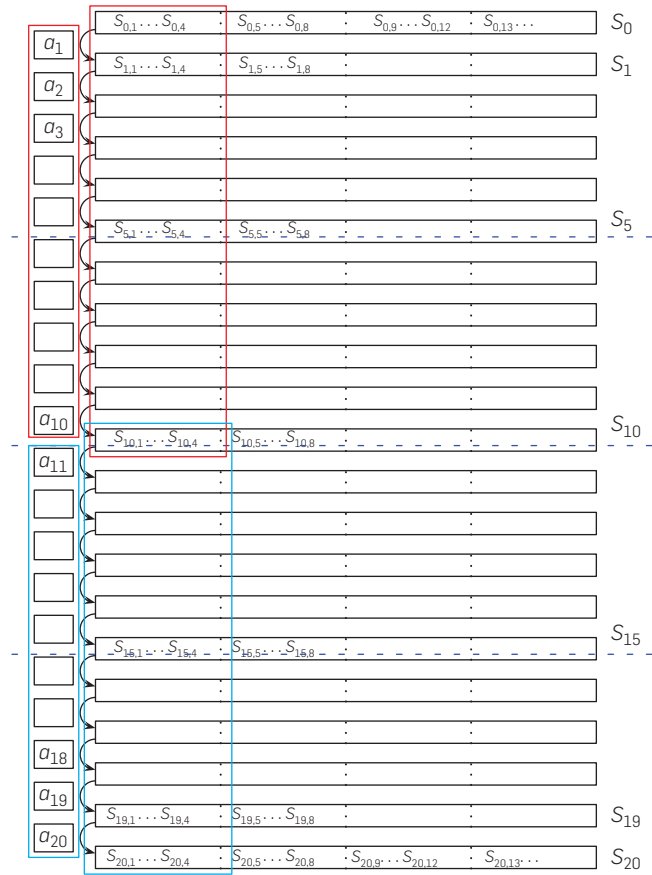
3.1. Improving the MITM algorithm using dissection

In this section, we show how to improve the classical MITM algorithm on Rubik's cube by using a basic version of our new dissection technique. The main idea here is to "dissect" the execution matrix in the middle by iterating over all the possible values of some part of the middle state S_{10} . The size of the partial S_{10} that we iterate on is chosen such that it contains about $2^{n/4} = 2^{19.5}$ partial states. Since S_{10} is represented as a 20-entry vector, where each entry can attain 24 values, we choose to iterate on its first 4 entries, which can assume about $24^4 \approx 2^{18.5}$ values. For each such partial value of S_{10} , we use the bicomposite structure of the problem in order to independently work on the two partial execution matrices shown in Figure 5 as red and blue rectangles, and finally join the partial solutions in order to obtain the full action vector.

The complete details of the algorithm are given in Figure 6. We have an outer loop which iterates over all the possible values of $S_{10,1}, S_{10,2}, S_{10,3}, S_{10,4}$. Assuming that this value is correctly guessed, we first concentrate on the upper part of the execution matrix and find all the partial action vectors a_1, \dots, a_{10} which transform $S_{0,1}, S_{0,2}, S_{0,3}, S_{0,4}$ into $S_{10,1}, S_{10,2}, S_{10,3}, S_{10,4}$. This is done using a simple MITM algorithm on this smaller execution matrix. For each solution a_1, \dots, a_{10} that we obtain using the MITM algorithm, we apply its actions to the full state S_0 , obtain candidate values of the full S_{10} state, and store it next to a_1, \dots, a_{10} in a list. After the MITM algorithm finishes populating the list, we sort it (e.g., in some lexicographic order) according to the value of S_{10} .

We now focus on the bottom execution matrix and find all the partial action vectors a_{11}, \dots, a_{20} which transform $S_{10,1}, S_{10,2}, S_{10,3}, S_{10,4}$ into $S_{20,1}, S_{20,2}, S_{20,3}, S_{20,4}$. We use the same idea that we used for the upper part, that is, we execute a MITM algorithm on the bottom execution matrix. For each solution a_{11}, \dots, a_{20} that we obtain, we apply its inverse actions to S_{20} and obtain a value for S_{10} . Then, we check for matches

Figure 5. Dissection of the Rubik's cube execution matrix.



for S_{10} in the sorted list, and for each match, we output a full solution a_1, a_2, \dots, a_{20} .

In order to analyze the algorithm, we fix a value of $S_{10,1}, S_{10,2}, S_{10,3}, S_{10,4}$ and estimate the average number of solutions that we expect for the upper (smaller) execution matrix. Namely, we calculate the expected number of action vectors a_1, \dots, a_{10} which transform $S_{0,1}, S_{0,2}, S_{0,3}, S_{0,4}$ into $S_{10,1}, S_{10,2}, S_{10,3}, S_{10,4}$. First, we notice that the number of possible action vectors a_1, \dots, a_{10} is about 2^{39} . Each such action vector transforms $S_{0,1}, S_{0,2}, S_{0,3}, S_{0,4}$ into an arbitrary partial state which matches $S_{10,1}, S_{10,2}, S_{10,3}, S_{10,4}$ with probability of about $1/(24^4) \approx 2^{-18.5}$ (which is inverse-proportional to the number of possible values of $S_{10,1}, S_{10,2}, S_{10,3}, S_{10,4}$). Thus, the expected number of solution (that we store in our sorted list) is $2^{39} \times 2^{-18.5} = 2^{20.5}$.

In general, the time complexity of the MITM algorithm is about square root of the search space, and thus its time complexity on the upper execution matrix is about $2^{39/2} = 2^{19.5}$. However, since in this case we could not split the problem into two parts of exactly equal sizes, we expect $2^{20.5}$ solutions (which we enumerate and store), and thus its time complexity is slightly increased to $2^{20.5}$. This is also the expected time complexity of the MITM algorithm on the bottom part (although here we do not store the solutions, but immediately check each one of them). Since we have an outer loop which we execute $24^4 \approx 2^{18.5}$ times, the

Figure 6. Solving the Rubik's cube using dissection into 4.

Algorithm Dissect4-Rubik

```

Input: An initial state  $S_0$  and a final state  $S_{20}$ 
for all  $S_{10,1}, S_{10,2}, S_{10,3}, S_{10,4}$  do
  Obtain all candidate  $(a_1, a_2, \dots, a_{10})$  satisfying  $S_{10,1}, S_{10,2}, S_{10,3}, S_{10,4} = a_{10}(\dots(a_2(a_1(S_{0,1}, S_{0,2}, S_{0,3}, S_{0,4})))\dots)$  by calling PartialMITM( $S_{0,1}, S_{0,2}, S_{0,3}, S_{0,4}, S_{10,1}, S_{10,2}, S_{10,3}, S_{10,4}$ )
  for all obtained  $a_1, a_2, \dots, a_{10}$  do
    Compute  $S_{10,5}, S_{10,6}, S_{10,7}, S_{10,8} = a_{10}(\dots(a_2(a_1(S_{0,5}, S_{0,6}, S_{0,7}, S_{0,8})))\dots)$ 
    Store  $(S_{10,5}, S_{10,6}, S_{10,7}, S_{10,8}, a_1, a_2, \dots, a_{10})$  in  $L_{10}$ 
    Sort  $L_{10}$  according to the values of  $S_{10,5}, S_{10,6}, S_{10,7}, S_{10,8}$  in each entry (under some lexicographical order)
    Obtain all candidates  $a_{11}, a_{12}, \dots, a_{20}$  satisfying  $S_{20,1}, S_{20,2}, S_{20,3}, S_{20,4} = a_{20}(\dots(a_{12}(a_{11}(S_{10,1}, S_{10,2}, S_{10,3}, S_{10,4})))\dots)$  by calling PartialMITM( $S_{10,1}, S_{10,2}, S_{10,3}, S_{10,4}, S_{20,1}, S_{20,2}, S_{20,3}, S_{20,4}$ )
    for all obtained  $a_{11}, a_{12}, \dots, a_{20}$  do
      Compute  $S_{10,5}, S_{10,6}, S_{10,7}, S_{10,8} = a_{11}^{-1}(\dots(a_{19}^{-1}(a_{20}^{-1}(S_{20,5}, S_{20,6}, S_{20,7}, S_{20,8})))\dots)$ 
      Search for  $S_{10,5}, S_{10,6}, S_{10,7}, S_{10,8}$  in  $L_{10}$ 
      if  $S_{10,5}, \dots, S_{10,8}$  are found then
        Obtain the associated  $a_1, a_2, \dots, a_{10}$  from  $L_{10}$ 
        if  $S_{20} = a_{20}(\dots(a_2(a_1(S_0)))\dots)$  then
          return  $a_1, a_2, \dots, a_{20}$  as the solution
  
```

Procedure PartialMITM

```

Input: An partial initial state  $S_{0,1}, S_{0,2}, S_{0,3}, S_{0,4}$  and a partial final state  $S_{10,1}, S_{10,2}, S_{10,3}, S_{10,4}$ 
for all  $a_1, a_2, \dots, a_5$  do
  Compute  $S_5 = a_5(\dots(a_2(a_1(S_{0,1}, S_{0,2}, S_{0,3}, S_{0,4})))\dots)$ 
  Store  $(S_{5,1}, S_{5,2}, S_{5,3}, S_{5,4}, a_1, a_2, \dots, a_5)$  in a list  $L_5$ 
  Sort  $L_5$  according to the values of  $S_{5,1}, S_{5,2}, S_{5,3}, S_{5,4}$  in each entry (under some lexicographical order)
  for all  $a_6, a_7, \dots, a_{10}$  do
    Compute  $S_{5,1}, S_{5,2}, S_{5,3}, S_{5,4} = a_6^{-1}(\dots(a_9^{-1}(a_{10}^{-1}(S_{10,1}, S_{10,2}, S_{10,3}, S_{10,4})))\dots)$ 
    Search for  $S_{5,1}, S_{5,2}, S_{5,3}, S_{5,4}$  in  $L_5$ 
    if  $S_{5,1}, \dots, S_{5,4}$  are found then
      Obtain the associated  $a_1, a_2, \dots, a_5$  from  $L_5$ 
      return  $a_1, a_2, \dots, a_{10}$  as a candidate solution
  
```

expected time complexity of the full algorithm is about $2^{18.5+20.5} = 2^{39}$. The expected memory complexity is $2^{20.5}$, required in order to store the solutions for the MITM on the upper part (note that we reuse this memory for each guess of $S_{0,1}, S_{0,2}, S_{0,3}, S_{0,4}$).

4. IMPROVED DISSECTION TECHNIQUES

A closer look at the algorithm presented in Section 3 reveals that the algorithm treats the top and bottom parts of the execution matrix differently. Indeed, while the suggestions from the top part are stored in a table (L_{10} in the example of Figure 6), the suggestions from the bottom part are checked on-the-fly against the table values. As a result, while the

number of suggestions in the top part is bounded from above by the size of the memory available for the algorithm, the number of suggestions from the bottom part can be arbitrarily large and generated on-the-fly in an arbitrary order.

This suggests that an asymmetric division of the execution matrix, in which the bottom part is significantly bigger than the top part, may lead to better performance of the algorithm.

In this section we show that this is indeed the case. As the algorithm for untangling the Rubik's cube presented in Section 3 is already practical on a PC so that there is no significant value in further improving it, we choose another classical search problem, known as the *combinatorial partition problem* to be our running example in this section.

The problem is defined as follows. We are given a set of n integers, $U = \{x_1, x_2, \dots, x_n\}$. Our goal is to partition U into two complementary subsets U_1, U_2 whose elements sum up to the same number, that is,

$$\sum_{x_i \in U_1} x_i = \sum_{x_j \in U_2} x_j. \quad (1)$$

The combinatorial partition problem is known to be NP-complete,⁴ and hence, one cannot expect a sub-exponential solution in general. Nevertheless, there are various techniques which allow to find a solution efficiently in various cases, especially when there exist many partitions (U_1, U_2) which satisfy Equation (1). We thus consider the “hardest” instance of the problem, in which each of the x_i s is of n digits in binary representation (i.e., $x_i \approx 2^n$). In this case, at most a few solutions (U_1, U_2) are expected to exist, and no sub-exponential algorithms for the problem are known. For the sake of simplicity, we focus on the *modular* variant of the problem, in which Equation (1) is slightly altered to

$$\sum_{x_i \in U_1} x_i \equiv \sum_{x_j \in U_2} x_j \pmod{2^n}. \quad (2)$$

As a specific numeric example, consider the case $n = 112$. In this case, checking all 2^{112} possible partitions is, of course, completely infeasible. The standard MITM algorithm allows to reduce the time complexity to 2^{56} , but it increases the space complexity to 2^{56} which is currently infeasible. As we show below, the problem can be represented as a bicomposite problem, and hence, the technique of Section 3 can be applied to obtain the better tradeoff of $T = 2^{56}$ and $S = 2^{28}$. While these numbers are almost practical, the relatively large amount of required memory disallows the use of FPGAs in the computation, which makes it barely feasible. We show below that by an *asymmetric* variant of the dissection algorithm, we are able to obtain the complexities $T = 2^{64}$ (which is only 2^8 times larger) and $S = 2^{16}$ (which is 2^{12} times smaller), which allow for a significantly faster computation using memory-constrained FPGAs. Note that the asymmetric dissection algorithm outperforms the symmetric one by a factor of 16 according to the complexity measure $S \times T$ (the complexities are 2^{80} vs. 2^{84}). Such a factor, while not extremely big, can make a difference in practical scenarios.

4.1. Representing combinatorial partition as a bicomposite search problem

In order to apply dissection algorithms to the combinatorial partition problem, we have to find a way to represent it as a bicomposite search problem.

First, we represent it as a composite problem. We treat the problem of choosing the partition (U_1, U_2) as a sequence of n atomic decisions, where the i th decision is whether to assign $x_i \in U_1$ or $x_i \in U_2$. We introduce a counter C which is initially set to zero, and then at the i th step, if the choice is $x_i \in U_1$ then C is replaced by $C + x_i \pmod{2^n}$, and if the choice is $x_i \in U_2$, C is replaced by $C - x_i \pmod{2^n}$. Note that the value of C after the n th step is $\sum_{x_i \in U_1} x_i - \sum_{x_j \in U_2} x_j \pmod{2^n}$, and hence, the sequence of choices leads to the desired solution if and only if the final value of C is zero.

In this representation, the partition problem has all the elements of a composite problem: an initial state ($C_{\text{initial}} = 0$), a final state ($C_{\text{final}} = 0$), and a sequence of n steps, such that in each step, we have to choose one of two possible atomic actions. Our goal is to find a sequence of choices which leads from the initial state to the final state. In terms of the execution matrix, we define S_i to be the value of C after the i th step (which is an n -bit binary number) and a_i to be the action transforming S_{i-1} to S_i , whose possible values are either $C \leftarrow C + x_i \pmod{2^n}$ or $C \leftarrow C - x_i \pmod{2^n}$.

The second step is to represent the problem as a bicomposite problem. The main observation we use here is the fact that for any two integers a, b , the m th least significant bit (LSB) of $a + b \pmod{2^n}$ depends only on the m LSBs of a and b (and not on their other digits). Hence, if we know the m LSBs of S_{i-1} and the action a_i , we can compute the m LSBs of S_i .

Using this observation, we define $S_{i,j}$ to be the j th LSB of S_i . This leads to an n -by- n execution matrix $S_{i,j}$ for $i, j \in 1, 2, \dots, n$ with the property that if we choose any rectangle within the execution matrix which includes the rightmost column of the matrix, knowledge of the substates $S_{i-1}^j, S_{i-1}^{j+1}, \dots, S_{i-1}^k$ along its top edge and knowledge of the actions a_i, a_{i+1}, \dots, a_i to its right suffices in order to compute the substates $S_i^j, S_i^{j+1}, \dots, S_i^k$ along its bottom edge.

Note that the condition satisfied by our execution matrix is weaker than the condition given in the definition of a bicomposite problem, since in our case, the “rectangle” property holds only for rectangles of a certain kind and not for all rectangles. However, as we show in the next subsection, even this weaker property is sufficient for applying all dissection algorithms we present.^b

4.2. Dissection algorithm for the combinatorial partition problem

The basic idea in the algorithm is to divide the state matrix into seven (!) parts of $n/7$ steps each, where three parts belong to the top part S^c and four parts belong to the bottom

^b For the sake of simplicity, we disregard the issue of carries. We note that in order to know all the carries required to execute our dissection algorithms, we guess the values of $S_{i,j}$ from the least significant bit to the most significant bit. For more details, refer to the extended version of this paper.²

part S^b . The partition is obtained by enumerating the $2n/7$ LSBs of the state $S_{3n/7}$.

For each value v of these bits, we perform a simple MITM algorithm in the top part, which yields about $2^{3n/7} \times 2^{-2n/7} = 2^{2n/7}$ possible combinations of actions $a_1, a_2, \dots, a_{3n/7}$ which lead to a state $S_{3n/7}$ whose $2n/7$ LSBs equal to the vector v . For each of these combinations, we compute the full value of the state $S_{3n/7}$. The resulting values of $S_{3n/7}$ are stored in a table, along with the corresponding combinations of $a_1, a_2, \dots, a_{3n/7}$.

Then we consider the bottom part, and apply to it the dissection algorithm described in Section 3 (thus, dividing it into four chunks of $n/7$ steps each). This results in $2^{4n/7} \times 2^{-2n/7} = 2^{2n/7}$ possible combinations of actions $a_{(3n/7)+1}, a_{(3n/7)+2}, \dots, a_n$ which lead (in the inverse direction) to a state $S_{3n/7}$ whose $2n/7$ LSBs equal to the vector v . For each of these combinations, we compute the full value $S_{3n/7}$ and compare it to the values in the table. If a match is found, this means that the corresponding sequences $\{a_1, a_2, \dots, a_{3n/7}\}$ and $a_{(3n/7)+1}, a_{(3n/7)+2}, \dots, a_n$ match to yield a solution of the problem. Note that if a solution exists, then our method must find it, since it actually goes all over all possible combinations of actions (though, in a sophisticated way). The pseudocode of the algorithm is given in Figure 7.

The memory complexity of the algorithm is $O(2^{2n/7})$, as both the standard MITM algorithm for the top part and the dissection algorithm for the bottom part have this complexity. (For the bottom algorithm, the complexity is $(2^{4n/7})^{1/4} = 2^{n/7}$.)

The time complexity is $2^{4n/7}$. Indeed, the enumeration in the state $S_{3n/7}$ is performed over $2^{2n/7}$ values, both the standard MITM algorithm for the top part and the dissection algorithm for the bottom part require $2^{2n/7}$ time, and the remaining $2^{2n/7}$ possible combinations of $a_{(3n/7)+1}, a_{(3n/7)+2}, \dots, a_n$ are checked instantly. This leads to time complexity of $2^{2n/7} \times 2^{2n/7} = 2^{4n/7}$.

In the special case of $n = 112$, each of the seven chunks consists of 16 steps, the enumeration is performed on the 28 LSBs of the state S_{32} , the memory complexity is $2^{n/7} = 2^{16}$, and the time complexity is $2^{4n/7} = 2^{64}$.

4.3. Advanced dissection algorithms

The algorithms presented in Section 3 and in this section are the two simplest dissection algorithms, which demonstrate the general idea behind the technique. In the extended version of the paper,² we present more advanced dissection algorithms, which include division of the matrix to “exotic” numbers of parts, such as 11 and 29, and show the optimality of such choices within our general framework.

So far we only considered search algorithms which are not allowed to fail (i.e., if there are any solutions to the problem then our algorithm will always find all of them, but its running time may be longer than expected if the instances are not randomly chosen or if the number of solutions is too large). In Dinur et al.,² we also consider algorithms which may fail to find a solution with a small probability, and show how to improve the efficiency of our algorithms

Figure 7. Solving the partitioning problem using dissection into 7.

Algorithm Dissect7-Partition

```

Input:  $U = \{x_1, x_2, \dots, x_n\}$ 
for all  $S_{3n/7,1}, S_{3n/7,2}, \dots, S_{3n/7,2n/7}$  ( $2n/7$  LSBs of  $S_{3n/7}$ ) do
  call PartialMITM( $S_{0,1}, S_{0,2}, \dots, S_{0,2n/7}, S_{3n/7,1}, S_{3n/7,2}, \dots, S_{3n/7,2n/7}$ )
  for all obtained  $a_1, a_2, \dots, a_{3n/7}$  do
    Compute the  $5n/7$  MSBs of  $S_{3n/7} = a_{3n/7}(\dots(a_2(a_1(S_0)))\dots)$ 
    Store  $(S_{3n/7}, a_1, a_2, \dots, a_{3n/7})$  in  $L_{3n/7}$ 
  Sort  $L_{3n/7}$  according to the values of  $S_{3n/7}$ 
  for all  $S_{5n/7,1}, S_{5n/7,2}, \dots, S_{5n/7,n/7}$  do
    call PartialMITM( $S_{3n/7,1}, S_{3n/7,2}, \dots, S_{3n/7,n/7}, S_{5n/7,1}, S_{5n/7,2}, \dots, S_{5n/7,n/7}$ )
    for all obtained  $a_{3n/7+1}, a_{3n/7+2}, \dots, a_{5n/7}$  do
      Compute the  $n/7$  bits  $S_{5n/7,n/7+1}, S_{5n/7,n/7+2}, \dots, S_{5n/7,2n/7} = a_{5n/7}$ 
       $(\dots(a_{3n/7+2}(a_{3n/7+1}(S_{3n/7,n/7+1}, S_{3n/7,n/7+2}, \dots, S_{3n/7,2n/7})))\dots)$ 
      Store  $(S_{5n/7}, a_{3n/7+1}, a_{3n/7+2}, \dots, a_{5n/7})$  in  $L_{5n/7}$ 
    Sort  $L_{5n/7}$  according to the values of  $S_{5n/7}$ 
    call PartialMITM( $S_{5n/7,1}, S_{5n/7,2}, \dots, S_{5n/7,n/7}, S_{7n/7,1}, S_{7n/7,2}, \dots, S_{7n/7,n/7}$ )
    for all obtained  $a_{5n/7+1}, a_{5n/7+2}, \dots, a_{7n/7}$  do
      Compute  $S_{5n/7,n/7+1}, S_{5n/7,n/7+2}, \dots, S_{5n/7,2n/7} = a_{5n/7+1}^{-1}(\dots(a_{7n/7-1}^{-1}(a_{7n/7}^{-1}(S_{5n/7})))\dots)$ 
      Search for  $S_{5n/7}$  in  $L_{5n/7}$ 
      if  $S_{5n/7}$  value is found then
        obtain  $a_{3n/7+1}, a_{3n/7+2}, \dots, a_{5n/7}$  from  $L_{5n/7}$ 
        for all obtained  $a_{3n/7+1}, a_{3n/7+2}, \dots, a_{5n/7}$  do
          Compute  $S_{3n/7,2n/7+1}, S_{3n/7,2n/7+2}, \dots, S_{3n/7,7n/7} = a_{3n/7+1}^{-1}(\dots(a_{7n/7-1}^{-1}(a_{7n/7}^{-1}(S_{7n/7})))\dots)$ 
          Search for  $S_{3n/7}$  in  $L_{3n/7}$ 
          if  $S_{3n/7}$  value is found then
            obtain  $a_1, a_2, \dots, a_{3n/7}$  from  $L_{3n/7}$ 
            return  $a_1, a_2, \dots, a_{7n/7}$  as a solution

```

Procedure PartialMITM

```

Input: A partial state  $S_{0,1}, S_{0,2}, \dots, S_{0,tn'}$ , a partial state  $S_{(t+1)n',1}, S_{(t+1)n',2}, \dots, S_{(t+1)n',tn'}$  and “distance”  $(t+1)n'$ 
for all  $a_1, a_2, \dots, a_{n'}$  do
  Compute  $S_{n',1}, S_{n',2}, \dots, S_{n',tn'} = a_{n'}(\dots(a_2(a_1(S_{0,1}, S_{0,2}, \dots, S_{0,tn'})))\dots)$ 
  Store  $(S_{n',1}, S_{n',2}, \dots, S_{n',tn'}, a_1, a_2, \dots, a_{n'})$  in a list  $L_{n'}$ 
  Sort  $L_{n'}$  according to the values of  $S_{n',1}, S_{n',2}, \dots, S_{n',tn'}$ 
  for all  $a_{n'+1}, a_{n'+2}, \dots, a_{(t+1)n'}$  do
    Compute  $S_{n',1}, S_{n',2}, \dots, S_{n',tn'} = a_{n'+1}^{-1}(\dots(a_{(t+1)n'-1}^{-1}(a_{(t+1)n'}^{-1}(S_{(t+1)n',1}, S_{(t+1)n',2}, \dots, S_{(t+1)n',tn'})))\dots)$ 
    Search for  $S_{n',1}, S_{n',2}, \dots, S_{n',tn'}$  in  $L_{n'}$ 
    if  $S_{n',1}, S_{n',2}, \dots, S_{n',tn'}$  are found then
      Obtain the associated  $a_1, a_2, \dots, a_{n'}$  from  $L_{n'}$ 
      return  $a_1, a_2, \dots, a_{(t+1)n'}$  as a candidate solution


```

in this case by combining them with a classical technique called *parallel collision search*, devised by Wiener and van Oorschot in 1996.⁷

5. CONCLUSION

In this paper we introduced the notion of bicomposite search problems, and developed new types of algorithmic techniques called dissection algorithms in order to solve them with improved time and space complexities. We demonstrated how to use these techniques by applying them to two standard types of problems (Rubik's cube and combinatorial partitions). However, some of the most exciting applications of these techniques are in cryptanalysis, which is beyond the scope of this paper. For example, many banks are still using a legacy cryptographic technique called *triple-DES*, which encrypts sensitive financial data by encrypting it three times with three independent keys. A natural question is whether using *quadruple-DES* (which encrypts the data four times with four independent keys) would offer significantly more security due to its longer key and more complicated encryption process. By using our new dissection techniques, we can show the surprising result that finding the full key of quadruple-DES could be achieved with essentially the same time and space complexities as finding the full key of the simpler triple-DES encryption scheme, and thus there is no significant security advantage in upgrading triple-DES to quadruple-DES.

Acknowledgments

The second author (O.D.) was supported in part by the Israel Science Foundation through grant No. 827/12 and in part by the German-Israeli Foundation for Scientific Research and Development through grant No. 2282-2222.6/2011. The third author (N.K.) was supported by the Alon Fellowship. 

References

1. Davidson, M., Dethridge, J., Kociemba, H., Rokicki, T. God's number is 20, 2010. <http://cube20.org>.
2. Dinur, I., Dunkelman, O., Keller, N., Shamir, A. Efficient dissection of composite problems, with applications to cryptanalysis, knapsacks, and combinatorial search problems. In *CRYPTO*, R. Safavi-Naini and R. Canetti, eds. Volume 7417 of *Lecture Notes in Computer Science* (2012). Springer, 719–740.
3. Fiat, A., Moses, S., Shamir, A., Shimshoni, I., Tardos, G. Planning and learning in permutation groups. In *FOCS*. IEEE Computer Society, 1989, 274–279.
4. Garey, M.R., Johnson, D.S. *Computers and Intractability: A Guide to the*
5. Horowitz, E., Sahni, S. Computing partitions with applications to the knapsack problem. *J. ACM* 21, 2 (1974), 277–292.
6. Slocum, J. *The Cube: The Ultimate Guide to the World's Bestselling Puzzle—Secrets, Stories, Solutions*. Black Dog & Leventhal Publishers, 2011.
7. van Oorschot, P.C., Wiener, M.J. Improving implementable meet-in-the-middle attacks by orders of magnitude. In *CRYPTO*, N. Kobitz, ed. Volume 1109 of *Lecture Notes in Computer Science* (1996). Springer, 229–236.

Itai Dinur and Adi Shamir ([itai.dinur, adi.shamir]@weizmann.ac.il), Computer Science Department, The Weizmann Institute, Rehovot, Israel.

Orr Dunkelman (orrd@cs.haifa.ac.il), Computer Science Department, University of Haifa, Israel.

Nathan Keller (nathan.keller@biu.ac.il), Department of Mathematics, Bar-Ilan University, Israel.

© 2014 ACM 0001-0782/14/10 \$15.00

World-Renowned Journals from ACM

ACM publishes over 50 magazines and journals that cover an array of established as well as emerging areas of the computing field. IT professionals worldwide depend on ACM's publications to keep them abreast of the latest technological developments and industry news in a timely, comprehensive manner of the highest quality and integrity. For a complete listing of ACM's leading magazines & journals, including our renowned Transaction Series, please visit the ACM publications homepage: www.acm.org/pubs.

ACM Transactions on Interactive Intelligent Systems



ACM Transactions on Interactive Intelligent Systems (TIIS). This quarterly journal publishes papers on research encompassing the design, realization, or evaluation of interactive systems incorporating some form of machine intelligence.

ACM Transactions on Computation Theory



ACM Transactions on Computation Theory (ToCT). This quarterly peer-reviewed journal has an emphasis on computational complexity, foundations of cryptography and other computation-based topics in theoretical computer science.

PLEASE CONTACT ACM MEMBER SERVICES TO PLACE AN ORDER
Phone: 1.800.342.6626 (U.S. and Canada)
+1.212.626.0500 (Global)
Fax: +1.212.944.1318
(Hours: 8:30am–4:30pm, Eastern Time)
Email: acmhelp@acm.org
Mail: ACM Member Services
General Post Office
PO Box 30777
New York, NY 10087-0777 USA



Association for
Computing Machinery

Advancing Computing as a Science & Profession

www.acm.org/pubs

CAREERS

California State Polytechnic University, Pomona

Computer Science Department
<http://www.csupomona.edu/~cs/>
Assistant Professor

The Computer Science Department invites applications for a tenure-track position at the rank of **Assistant Professor** to begin Fall 2015. We are particularly interested in candidates with specialization in Software Engineering, Cloud Computing, Data Mining, or Computer Graphics and Animation. Cal Poly Pomona is 30 miles east of L.A. and is one of 23 campuses in the California State University. The department offers an ABET-accredited B.S. program and an M.S. program. Qualifications: Possess, or complete by September 1, 2015, a Ph.D. in Computer Science or closely related area. Demonstrate strong communication skills, commitment to actively engage in the teaching, research, and curricular development activities of the department at both undergraduate and graduate levels, and ability to work with a diverse student body and multicultural constituencies. Ability to teach a broad range of courses, and to articulate complex subject matter to students at all educational levels. First consideration will be given to completed applications received no later than November 14, 2014. Contact: Faculty Search Committee, Computer Science Department, Cal Poly Pomona, Pomona, CA 91768. Email: cs@csupomona.edu. Cal Poly Pomona is an Equal Opportunity, Affirmative Action Employer. Position announcement available at: <http://academic.csupomona.edu/faculty/positions.aspx>. Lawful authorization to work in US required for hiring.

California State University, East Bay (Hayward, CA) Department of Mathematics and Computer Science Faculty Position in Computer Science

POSITION (OAA Position No. 15-16 MCS-DATA/CLOUD-TT): The Department invites applications for a tenure-track appointment as Assistant Professor in Computer Science (preference to emerging areas of computer science such as big data, cloud computing, mobile app development, data mining) starting Fall 2015.

Teaching includes day and evening courses at B.S. and M.S. levels, with a typical teaching responsibility of three classes/Quarter. Required: potential for excellent teaching, research and curriculum development; ability to teach, advise and mentor students from diverse backgrounds; potential to serve Department and University. Applicants must have a Ph.D. by September 2015.

Please submit an application letter and a current and complete vita through the following URL: https://my.csueastbay.edu/psp/pspdb1/EMPLOYEE/HRMS/c/HRS_HRAM.HRS_CE.GBL

Additionally, please email graduate transcripts, 3 letters of recommendation, 3 references, a statement of teaching philosophy, and evidence of teaching and research abilities to the Computer Science Search Committee at the email address below.

A detailed position announcement is available at: <http://www20.csueastbay.edu/about/career-opportunities/>

For questions, email: CSSearch@mcs.CSUEastBay.edu

APPLICATION DEADLINE The deadline for applications is October 31, 2014; review of applications will begin November 1, 2014. CSUEB, situated in the hills overlooking San Francisco Bay, is an EOE, committed to "educational excellence for a diverse society". If you are considered as a finalist for the position, you may be subject to a background check.

Drexel University College of Computing & Informatics Teaching Faculty Positions

Drexel University's College of Computing & Informatics (www.cci.drexel.edu) invites applications for multiple full-time teaching faculty positions. The department of Computing offers BS, BA, MS, and PhD degrees in computer science, as well as BS and MS degrees in software engineering.

Areas of relevant teaching expertise include computer security, computer game programming and design; C++, Java, Python programming; data structures; Unix scripting and program development; CS mathematical foundations; web and mobile app development; systems programming and architecture; software engineering fundamentals; and software specification, design and architecture.

Drexel is a private university committed to research with real-world applications. The university has over 25,000 students in 14 colleges and schools and offers about 200 degree programs. The College of Computing and Informatics has approximately 75 faculty and 2,300 students. Drexel has one of the largest and best-known cooperative education programs, with over 1,200 co-op employers. Drexel is located on Philadelphia's "Avenue of Technology" in the University City District and at the hub of the academic, cultural, and historical resources of the nation's sixth largest metropolitan region.

Review of applications begins immediately. Possession of a doctoral degree in computer science or related disciplines is preferred. To be considered, apply at www.drexeljobs.com/applicants/Central?quickFind=78885

Your application should consist of a cover letter, CV, a brief statement describing your teaching interests, and a list of references. Letters of

reference will be requested from the candidates who are invited for a campus interview. Electronic submissions in PDF format are required.

Drexel University is an Equal Opportunity/Affirmative Action Employer. The College of Computing & Informatics is especially interested in qualified candidates who can contribute to the diversity and excellence of the academic community. Background investigations are required for all new hires as a condition of employment, after the job offer is made. Employment will be contingent upon the University's acceptance of the results of the background investigation.

Grinnell College Assistant Professor, Computer Science

GRINNELL COLLEGE. Tenure-track Computer Science starting Fall 2015. Asst Prof (PhD) preferred; Instructor (ABD) or Assoc Prof possible. Area open. Details/instructions: <https://jobs.grinnell.edu>. Candidates will upload letter of application, cv, transcripts, teaching statement, description of scholarly activities, email addresses for 3 refs. Questions: Prof Sam Rebelsky, CSSearch@grinnell.edu, 641-269-3169. Deadline: Nov. 8, 2014. AA/EOE

Lehigh University Assistant or Associate Professor

Applications are invited for two tenure-track positions at the Assistant or Associate Professor level in the Computer Science and Engineering Department (<http://www.cse.lehigh.edu>) of Lehigh University to start in August 2015. Outstanding candidates in all areas of computer science will be considered.

The successful applicant will hold a Ph.D. in Computer Science, Computer Engineering, or a closely related field. The candidate must demonstrate a strong commitment to quality undergraduate and graduate education, and the potential to develop and conduct a high-impact research program with external support. Applicants should have an interest in teaching core courses in computer science as well as courses in their research area. The successful applicant will also be expected to contribute to interdisciplinary research programs.

The faculty of the Computer Science and Engineering department maintains an outstanding international reputation in a variety of research areas, and includes ACM and IEEE fellows as well as six NSF CAREER award winners. We offer B.A., B.S., M.S., and Ph.D. degrees in Computer Science and jointly oversee B.S., M.S., and Ph.D. degree programs in Computer Engineering with the department of Electrical and Computer Engineering. We also offer a B.S. in Computer Science and Business with the College of Business and Economics.

Lehigh University is a private, highly selective institution that is consistently ranked among the top 40 national research universities by U.S. News & World Report. Located in Bethlehem, Pennsylvania, Lehigh is 80 miles west of New York City and 50 miles north of Philadelphia, providing an accessible and convenient location that offers an appealing mix of urban and rural lifestyles.

Applications can be submitted online at <https://academicjobsonline.org/ajo/jobs/4239> and should include a cover letter, vita, both teaching and research statements, and contact information for at least three references. Review of applications will begin December 1, 2014 and will continue until the positions are filled.

Lehigh University is an affirmative action/equal opportunity employer and does not discriminate on the basis of age, color, disability, gender, gender identity, genetic information, marital status, national or ethnic origin, race, religion, sexual orientation, or veteran status. Lehigh University is a 2010 recipient of an NSF ADVANCE Institutional Transformation Grant for promoting the careers of women in academic science and engineering. Lehigh University provides comprehensive benefits including domestic partner benefits (see also <http://www.lehigh.edu/worklifebalance/>). Lehigh Valley Inter-regional Networking & Connecting (LINC) is a newly created regional network of diverse organizations designed to assist new hires with dual career, community and cultural transition needs. Please contact infdcap@lehigh.edu for more information. Questions concerning this search may be sent to faculty-search@cse.lehigh.edu.

Massachusetts Institute of Technology Faculty Positions

The Department of Electrical Engineering and Computer Science (EECS) seeks candidates for faculty positions starting in September 2015. Appointment will be at the assistant or untenured associate professor level. In special cases, a senior faculty appointment may be possible. Faculty duties include teaching at the undergraduate and graduate levels, research, and supervision of student research. Candidates should hold a Ph.D. in electrical engineering and computer science or a related field by the start of employment. We will consider candidates with research and teaching interests in any area of electrical engineering and computer science.

Candidates must register with the EECS search website at <https://eeecs-search.eecs.mit.edu>, and must submit application materials electronically to this website. Candidate applications should include a description of professional interests and goals in both teaching and research. Each application should include a curriculum vitae and the names and addresses of three or more individuals who will provide letters of recommendation. Letter writers should submit their letters directly to MIT, preferably on the website or by mailing to the address below. Please submit a complete application by December 1, 2014.

Send all materials not submitted on the website to:
Professor Anantha Chandrakasan
Department Head, Electrical Engineering
and Computer Science

Massachusetts Institute of Technology
Room 38-401
77 Massachusetts Avenue
Cambridge, MA 02139

M.I.T. is an equal opportunity/
affirmative action employer.

National University of Singapore Multiple Tenure-Track Faculty Positions

The Department of Computer Science, National University of Singapore (NUS), has openings for several tenure-track faculty positions. Our main focus is on candidates at the Assistant Professor level with research interests in the following areas:

- ▶ Cyber-physical systems
- ▶ Big data analytics
- ▶ Security
- ▶ Sensor data modelling and learning

These areas are to be viewed in a broad sense, and we are particularly interested in candidates whose research interests cut across these and related areas. We seek candidates demonstrating excellent research potential and a strong commitment to teaching. We will also seriously consider exceptional candidates in other areas of computer science. Further, we will consider candidates at senior ranks (Associate and Full Professor) who have an outstanding record of research accomplishments.

We are an internationally top-ranked depart-

Call for Assistant Professors and Professors



IST Austria invites applications for **Tenure-Track Assistant Professor** and **Tenured Professor** positions to lead independent research groups in all areas of

COMPUTER SCIENCE

IST Austria is a recently founded public institution dedicated to basic research and graduate education near Vienna. Currently active fields of research include biology, neuroscience, physics, mathematics, and computer science. IST Austria is committed to become a world-class research center with up to 1000 scientists and doctoral students by 2026. The institute has an interdisciplinary campus, an international faculty and student body, as well as state-of-the-art-facilities. The working language is English.

Successful candidates will be offered highly competitive research budgets and salaries. Faculty members are expected to apply for external research funds and participate in graduate teaching. Candidates for senior positions must be internationally accomplished scientists in their respective fields.

DEADLINES: Open call for Professor applications. For full consideration, Assistant Professor applications should arrive on or before November 15, 2014. Application material must be submitted online: www.ist.ac.at/professor-applications

IST Austria values diversity and is committed to equal opportunity. Female researchers are especially encouraged to apply.



Professor of Computer Science (Medical Informatics)

→ The Department of Computer Science (www.inf.ethz.ch) at ETH Zurich invites applications for a Professor of Computer Science in the area of Medical Informatics.

→ Applicants should have an excellent record of internationally recognized research, which demonstrates a strong link of core computer science (information systems & machine learning) and applications in medicine and life science. Examples for the computer science expertise of the successful candidate include, but are not limited to, medical data analytics, data management, cloud computing, computational medicine, clinical care systems, possibly complemented by computational science or medical imaging.

→ The professor is expected to establish and lead a research group in the Department of Computer Science with close links to the University Hospital Zurich. Active involvement in the network "Life Science Zurich" is also envisioned, especially with the Competence Center for Personalized Medicine. The professorship is embedded in a vibrant research and teaching community for interdisciplinary medical and life science research that benefits from the proximity of the University Hospital Zurich and ETH Zurich and ranges from biology, systems biology and biotechnology to life science, medicine, and health research in various engineering departments.

→ Candidates are expected to supervise graduate students, to teach undergraduate level courses (German or English) and graduate level courses (English) in his/her own field of research, and to participate in core courses of computer science. The expectation is to fill the position with a tenured full professor but excellent applications at the (tenure track) assistant professor level will also be considered.

→ Please apply online at www.facultyaffairs.ethz.ch

→ Applications should include a curriculum vitae, a list of publications, a statement of future research and teaching interests and the names of at least five references. The letter of application should be addressed to the **President of ETH Zurich, Prof. Dr. Ralph Eichler**. The closing date for applications is **15 December 2014**. ETH Zurich is an equal opportunity and family friendly employer and is further responsive to the needs of dual career couples. We specifically encourage women to apply.

ment with low teaching loads, excellent facilities, and intensive external collaborations. Significant funding opportunities abound for strong candidates. The research of the faculty covers all the major areas of computer science and is well represented at prestigious international conferences and journals. The department has a thriving PhD programme and it actively strives to attract the best students from the region and beyond. More information can be found at <http://www.comp.nus.edu.sg/>.

NUS offers highly competitive salaries and generous benefits, while Singapore provides a vibrant international environment with world-class health care, excellent infrastructure, a warm climate and very low taxes.

Interested candidates are invited to send, via electronic submission, the following materials to the Chair of the CS Search Committee, Prof. P.S. Thiagarajan, at csrec@comp.nus.edu.sg

- ▶ Cover letter
- ▶ Curriculum Vitae
- ▶ A teaching statement
- ▶ A research statement
- ▶ Contact information for at least three references

Applications will be reviewed as they are received and will continue until the positions are filled. However, to ensure maximal consideration applicant should submit their materials by December 15, 2014.

New Mexico State University Assistant Professor

The Computer Science Department at New Mexico State University invites applications for two tenure-track positions at the Assistant Professor level, with appointments starting in the Fall 2015 semester. We are seeking strong candidates with research expertise that can effectively complement the research foci of the department; we are particularly interested in expertise in the areas of (a) data management and analysis; or (b) software engineering, programming languages, and compilers. Applications from women, members of traditionally under-represented groups, and other individuals interested in contributing to the diversity and excellence of the academic community are strongly encouraged. The minimum qualifications are a Ph.D. degree in Computer Science, or in a closely-related discipline, by the time of appointment, along with evidence of excellence in research and teaching. For the full position announcement, please visit <http://www.cs.nmsu.edu/wp13/faculty-opening/>

Contact: CS Search Chair

Email: cssearch@cs.nmsu.edu

Apply URL: <http://www.cs.nmsu.edu/wp13/faculty-opening>

Phone: 575-646-1038

Fax: 575-646-1002

New York Institute of Technology Computer Science Cyber Security Adjunct

New York Institute of Technology seeks an adjunct faculty for its Vancouver campus to offer instruction in its Master of Science in Information, Network and Computer Security, beginning 9/1/14.

PHD degree in Computer Science with a specialization in system security issues. Email curriculum vitae & cover letter by 8/22/14, to Van.incs.jobs@nyit.edu. All qualified candidates are encouraged to apply; however, Canadian Citizens and Permanent Residents will be given priority. EOE M/F/D/V.

Princeton University
Computer Science
Assistant Professor - Tenure Track

The Department of Computer Science at Princeton University invites applications for faculty positions at the Assistant Professor level. We are accepting applications in all areas of Computer Science. Applicants must demonstrate superior research and scholarship potential as well as teaching ability. A PhD in Computer Science or a related area is required. Candidates should expect to receive their PhD before Fall, 2015. More senior appointments may be considered for extraordinary candidates. Successful candidates are expected to pursue an active research program and to contribute significantly to the teaching programs of the department. Applicants should include a CV and contact information for at least three people who can comment on the applicant's professional qualifications. There is no deadline, but review of applications will be underway by December 2014.

Princeton University is an equal opportunity employer. All qualified applicants will receive consideration for employment without regard to

race, color, religion, sex, national origin, disability status, protected veteran status, or any other characteristic protected by law. This position is subject to the University's background check policy.

You may apply online at:
<http://jobs.cs.princeton.edu/>

Princeton University
Computer Science
Part-Time or Full-Time Lecturer

The Department of Computer Science seeks applications from outstanding teachers to assist the faculty in teaching our introductory course sequence or some of our upper-level courses.

Depending on the qualifications and interests of the applicant, job responsibilities will include such activities as teaching recitation sections and supervising graduate-student teaching assistants; grading problem sets and programming assignments; supervising students in the grading of problem sets and programming assignments; developing and maintaining online curricular material, classroom demonstrations, and laboratory exercises; and supervising undergraduate research projects. An advanced degree in computer science, or related field, is required (PhD preferred).

The position is renewable for 1-year terms, up to six years, depending upon departmental need and satisfactory performance.

To apply, please submit a cover letter, CV, and contact information for three references to <http://www.cs.princeton.edu/jobs/lecturerposition>

Princeton University is an equal opportunity employer. All qualified applicants will receive consideration for employment without regard to race, color, religion, sex, national origin, disability status, protected veteran status, or any other characteristic protected by law. Finalist candidates to be hired will be required to complete a successful background check.

University of Chicago
Department of Computer Science
Sr. Lecturer

The Department of Computer Science at the **University of Chicago** invites applications for the position of **Sr. Lecturer**. This position carries responsibility for teaching computer science courses and laboratories in the fall, winter and spring quarters and leading academic initiatives in the program.

This position involves advising undergraduates on their coursework and career paths. In collaboration with faculty, this senior lecturer will update, revise, and develop curriculum. In addition, this senior lecturer will train and evaluate graduate student lab instructors, as well as mentor junior faculty in pedagogy.

Applicants must have a PhD in Computer Science or a related field and have experience teaching Computer Science at an undergraduate level. The successful candidate will have exceptional competence in teaching and superior academic credentials.

The Chicago metropolitan area provides a diverse and exciting environment. The local economy is vigorous, with international stature in banking, trade, commerce, manufacturing, and transportation, while the cultural scene includes diverse cultures, vibrant theater, world-renowned symphony, opera, jazz and blues. The University is located in Hyde Park, a Chicago neighborhood on the Lake Michigan shore just a few minutes from downtown.

Applicants must apply on line at the University of Chicago Academic Careers website at <http://tinyurl.com/pf3sdt>.

Applicants must upload a cover letter, curriculum vitae with a list of publications and a one page teaching statement. In addition, three reference letters that address the candidate's teaching qualifications will be required. Review of complete applications, including reference letters, will begin November 15, 2014, and continue until the position is filled.

All qualified applicants will receive consideration for employment without regard to race, color, religion, sex, national origin, age, protected veteran status or status as an individual with disability.

The University of Chicago is an Affirmative Action/Equal Opportunity/ Disabled/Veterans Employer.

University of Tennessee at Martin
Assistant Professor of Computer Science

UTM seeks to fill a tenure-track position beginning ASAP 2015. A PhD in CS (or related field) required. Candidates who are ABD may apply for a position as Lecturer, but a PhD is required for tenure and appointment as an Assistant Profes-



THE CHINESE UNIVERSITY OF HONG KONG

Applications are invited for:-

Faculty of Engineering
Professors / Associate Professors / Assistant Professors
(Ref. 1415/039(255)/2)

The Faculty invites applications for several faculty posts at Professor / Associate Professor / Assistant Professor levels, with prospect for substantiation, in the interdisciplinary area of 'Big Data Analytics', which is a new strategic research initiative supported by the University's Focused Innovations Scheme and will complement current/planned strengths in different Departments under the Faculty. To lead the big data research initiative, senior professors in this area are particularly welcome. Currently, the Faculty is seeking candidates in the following areas:

- Theoretical, mathematical and algorithmic aspects in large data analytics;
- Large scale software systems and architecture in large data analytics;
- Combining big data analytics with statistical modelling and operations research methods for optimal decision making;
- Application areas in large data analytics (including information systems and the Web, bioinformatics, financial engineering, logistics and supply chain management, public health, social networks, etc.)

Applicants should have (i) a PhD degree; and (ii) a strong scholarly record demonstrating potential for teaching and research excellence. The appointees will be expected to (a) teach undergraduate and postgraduate courses; (b) develop a significant independent research programme with external funding; and (c) supervise postgraduate students. Appointments will normally be made on contract basis for three years initially, which, subject to performance and mutual agreement, may lead to longer-term appointment or substantiation later. Applications will be accepted until the posts are filled. Further information about the Faculty is available at <http://www.erg.cuhk.edu.hk>.

Salary and Fringe Benefits

Salary will be highly competitive, commensurate with qualifications and experience. The University offers a comprehensive fringe benefit package, including medical care, plus a contract-end gratuity for appointments of two years or longer, and housing benefits for eligible appointees. Further information about the University and the general terms of service for appointments is available at <http://www.per.cuhk.edu.hk>. The terms mentioned herein are for reference only and are subject to revision by the University.

Application Procedure

Please send full resume, copies of academic credentials, a publication list with abstracts of selected published papers, details of courses taught and evaluation results (if any), a research plan, a teaching statement, together with names of three to five referees, to the Dean, Faculty of Engineering by e-mail to recruit-bda@erg.cuhk.edu.hk. For enquiries, please contact Professor John C.S. Lui, the leader of the strategic initiative (e-mail: cslui@cse.cuhk.edu.hk). Applicants are requested to clearly indicate that they are applying for the posts under 'Big Data Analytics Initiative'. The Personal Information Collection Statement will be provided upon request. Please quote the reference number and mark 'Application - Confidential' on cover.

sor. Candidates must teach a variety of computer science courses typical of a 4-year CS program. Apply at <http://www.utm.edu/departments/personnel/employment.php>. Review of applications will begin 9/15/2014 and continue until the position is filled.

Washington State University Vancouver Assistant Professor

COMPUTER SCIENCE FACULTY – Washington State University Vancouver invites applications for a full-time tenure-track position at the assistant professor level beginning 8/16/2015. Candidates are sought with expertise in **database systems** and **data management**. Additional expertise in **operating systems**, **cloud computing**, **Hadoop** and/or **data mining** are also desired.

Required qualifications: Ph.D. in Computer Science or Software Engineering by the employment start date and demonstrated ability to (1) develop a funded research program, (2) establish industrial collaborations, (3) teach undergraduate/graduate courses, and (4) contribute to our campus diversity goals (e.g. incorporate issues of diversity into mentoring, curriculum, service or research). **Preferred qualifications:** (1) already have published promising scholarly work in the field and (2) relevant industrial background.

Duties include: (1) teaching at undergraduate and graduate levels including the topics of database systems and management, and operating systems, cloud computing, and/or data mining; (2) participation and documentation of distinguished scholarly activities including research, innovative teaching and laboratory development; (3) securing external funding for research programs; and (4) service to the department and university through committee work, recruitment, and interaction with industry.

WSU Vancouver serves about 3,000 graduate and undergraduate students and is **fifteen miles north of Portland, Oregon**. The rapidly growing School of Engineering and Computer Science (ENCS) equally values both research and teaching. WSU is Washington's land grant university with faculty and programs on four campuses. For more information: <http://ecs.vancouver.wsu.edu>. WSU Vancouver is committed to building a culturally diverse educational environment.

To apply: Please visit www.wsujobs.com and search postings by location. Applications must include: (1) cover letter with a clear description of experience relevant to each of the required and preferred qualifications; (2) vita including a list of at least three references, and (3) A statement (**two page total**) of how candidate's research will expand/complement the current research in ENCS and a list of the existing ENCS courses the candidate can teach and any new courses the candidate proposes to develop. Application deadline is **November 28, 2014**.

WASHINGTON STATE UNIVERSITY IS AN EQUAL OPPORTUNITY/AFFIRMATIVE ACTION EDUCATOR AND EMPLOYER. Members of ethnic minorities, women, special disabled veterans, veterans of the Vietnam-era, recently separated veterans, and other protected veterans, persons

of disability and/or persons age 40 and over are encouraged to apply. WSU employs only U.S. citizens and lawfully authorized non-U.S. citizens.

York University Canada Research Chair in Digital Media (Tier 2)

The Department of Electrical Engineering and Computer Science, York University invites applications for a tenure-track position in the area of Digital Media at the **Assistant Professor or Associate Professor level** with a Canada Research Chair (CRC) Tier 2 appointment to commence no later than July 1, 2015. The Department offers programs in Computer Engineering, Computer Science, Computer Security, Electrical Engineering, Software Engineering and Digital Media.

This position will attract a highly-successful early career researcher with an established and innovative program of research and teaching in the area of digital media. The successful candidate will have demonstrable expertise in one or more of the following: computer graphics, virtual reality, human machine interaction, spatial audio, haptics, natural language processing/synthesis, visual analytics and closely allied areas.

Linked to the University Strategic Plan, which identifies Digital Cultures as a compelling opportunity for development, this appointment is part of a pan-university strategic investment in the area. The successful candidate will be expected to interact with existing researchers in related areas within the department and to build linkages to other faculty hires related to Digital Culture

across the university who work at the intersection of arts and sciences. Tier 2 CRC Chairs are research-intensive faculty positions allowing the holder to grow their research program through prioritization on research and access to infrastructure funding. The awards have 5-year terms, are renewable once and are intended for exceptional emerging researchers — less than 10 years post terminal degree at the time of CRC application. Information about the CRC program can be found at <http://www.chairs.gc.ca>.

York University offers a world-class, interdisciplinary academic experience in Toronto, Canada's most multicultural city. York is a centre of innovation, with a thriving community of almost 60,000 faculty, staff and students.

Applicants should complete the online application process at <http://lassonde.yorku.ca/new-faculty/> with a detailed CV, statement of contribution to research and teaching, links to scholarly and/or creative work, evidence for their eligibility for a CRC Tier 2 appointment and 3 reference letters. **The deadline for receipt of applications is November 30, 2014.**

All York University positions are subject to budgetary approval. York University is an Affirmative Action (AA) employer and strongly values diversity, including gender and sexual diversity, within its community. The AA program, which applies to Aboriginal people, visible minorities, people with disabilities and women, can be found at <http://yorku.ca/acadjobs> or by calling the AA office at 416-736-5713. All qualified candidates are encouraged to apply; however, Canadian citizens and Permanent Residents will be given priority.

Faculty Search

ShanghaiTech University

ShanghaiTech University invites highly qualified candidates to fill multiple tenure-track/tenured faculty positions as its core team in the School of Information Science and Technology (SIST). Candidates should have exceptional academic records by international standards or demonstrate strong potential in cutting-edge research areas of information science and technology. English fluency is required and overseas academic experience is highly desired.

ShanghaiTech aims to become a world-class research university for training future scientists, entrepreneurs, and technological leaders. Located in Zhangjiang High-Tech Park in the cosmopolitan Shanghai, we shall trail-blaze a new education system in China. Besides establishing and maintaining a world-class research profile, faculty candidates must also contribute substantially to graduate and undergraduate education

Academic Disciplines: We welcome candidates in all cutting edge areas of information science and technology. Our recruitment focus includes, but is not limited to: computer architecture and software, cloud and high performance computing, computational foundations, data mining and analysis, visualization, computer vision, machine learning, data sciences and statistics, IC designs, solid-state electronics, high speed and RF circuits, embedded systems, intelligent and signal processing systems, smart energy/power devices and systems, next-generation networking, control systems, robotics, sensor networks as well as inter-disciplinary areas involving information science and technology.

Compensation and Benefits: Salary and startup funds are highly competitive, commensurate with experience and academic accomplishment. We also offer a comprehensive benefit package to employees and eligible dependents, including housing benefits. All regular faculty members will join our new tenure-track system commensurate with international practice for tenure evaluation and promotions.

Qualifications:

- A well articulated research plan and demonstrated record/potentials;
- Ph.D. (Electrical Engineering, Computer Engineering, Computer Science, Statistics, or related field);
- A minimum relevant research experience of 4 years.

Applications: Submit (in English, PDF) a cover letter, a 2-page research plan, a CV plus copies of 3 most significant publications, and names of three referees to: sist@shanghaitech.edu.cn by October 31st, 2014 (or until positions are filled). More information is at <http://www.shanghaitech.edu.cn>.

[CONTINUED FROM P. 112] wake boards going faster than I'm comfortable driving my Chevy half-ton on the highway. That's not for me. I get my jollies in the wild, hunting for brand new insect species. I send my specimens to an old buddy at the university's entomology department, then play the waiting game. Sometimes, I hit paydirt ... and then comes the best part—I get to name them.

Gryllus oklahomas was my last find. A field cricket under a piece of rotten bark. Named that little dude in honor of my state and kept on looking for more.

I find the fist-size knot in the base of a dead pine tree. Sort of a cubbyhole in the shape of a stop sign. The little hexagon has too many straight lines to be from nature, so I stoop down on creaking knees and take a look.

The smell of burning wood wafts from the hole. Peering into the thing, I see that it holds what looks like a plastic cube with an ember in it. And around the lip of the hexagonal hole, I see a brownish leg that is moving in precise jerks. It has a claw tip and it's busy scratching ... building something. Staring at it for a second, I realize it's making another little arm, a perfect copy of itself.

"What in the ...," I ask the empty woods.

When I was a kid, I used to find fairy nests near the creek that ran behind my parents' place. Little shacks made of sticks and moss and leaves, placed around a shade tree or in a sun-dappled clearing. In those days kids roamed free, and I spent a whole summer hunting those fairy nests with a kind of magic in my heart. One night at dinner, I finally told my mama

Thousands of them, the color of dirt and leaves, dragging themselves like a living carpet over the ground.

And even though I start out walking at a reasonable pace ... before long I'm running.

about what I'd found. I will always remember the little smile she gave me. All of a sudden I knew exactly where those fairy houses came from.

I knew, but I never stopped searching.

Something tickles my hand and I give a yelp. In the dirt, I see a handful of marbles. Only they're moving on lots of legs, like pill bugs, or roly-polys, as we once called them. The insects are trundling and falling over a piece of bark, peeling splinters from the wood. The size of thimbles, each one has a raspy spot on its belly. They drag themselves over the wood and shred little pieces off. I watch one pick up a splinter with tiny mandibles and climb right into the hexagonal hole.

He's tending the fire in there. Keeping energy going to his factory.

This is a whole new deal. I climb onto my knees and rifle through my field bag. Pick up one of the little crawlers with tweezers and drop it into a glass specimen jar. I screw the lid on tight and wonder if it really needs air holes. Can't say whether this little dude breathes or not. For the life of me, it looks like the bug is made of some kind of *metal*.

Government land, you know? Hard to say what the scientists are doing in those fenced-off buildings. Only thing I know is that life likes to break free.

Life likes to spread.

Getting to my feet, I shade my eyes and look deeper into the woods. Now, I notice a lot of the trees are dead. More than usual. And it may just be my old eyes, but I feel like there's a haze over everything. A thin smear of smoke from more of those miniature power plants ... more smoldering factories out there in the sun-baked woods.

I stand still, and the movement of

the crawlers seeps into view. Thousands of them, the color of dirt and leaves, dragging themselves like a living carpet over the ground. The hair goes up on the backs of my arms. Somewhere out there, far off, I hear a tree splinter and crack. A shadow sweeps and I hear a hollow thump.

The captured crawler clinks against its jar and I flinch a little bit. Time to get back to the truck. I shrug my pack on tighter and turn around. And even though I start out walking at a reasonable pace ... before long I'm running.

In the truck, I don't take it easy on the accelerator. I'm on a dirt road for a few miles, meandering alongside razor-wire fences. My tires chew the rocks loudly, and I can't see anything but my dust trail in the rear-view, which is fine with me. When I finally stop at the sign to get onto paved road, it gets quiet except for my ragged breathing. I'm gripping the steering wheel, knuckles like mountain ridges.

Then I hear the scratching sound from next to me.

Fingers shaking, I swipe all the trash off the passenger seat. A curled yellow newspaper, a pair of work gloves, and an *Auto Trader* magazine waterfall onto the floorboard. Underneath, I find more of my friends-with-no-names. Guess I must have left my window open while I was exploring the woods. The crawlers are busy making themselves at home, now pulling strips out of the seat fabric.

In the seat back, they're carving out a neat hexagon.

I take a deep breath and put my foot on the gas. A shaky smile has got onto my face. The garden of life, see ... she's way beyond the ken of humankind. The textbooks say we've barely scratched the surface. At the first exit, I head off toward the university. I'm pretty sure these bugs are made of metal, and I'm pretty sure they were built and not born. But if nature doesn't care, then neither do I.

I'm already thinking of names. ▣

Daniel H. Wilson is the *New York Times* bestselling author of *Robocalypse* and its sequel *Robogenesis*. He earned a Ph.D. in robotics from Carnegie Mellon University. Follow him on Twitter @danielwilsonPDX or visit his website at <http://www.danielwilson.com>

© 2014 ACM 0001-0782/14/10 \$15.00

From the intersection of computational science and technological speculation, with boundaries limited only by our ability to imagine what could be.

DOI:10.1145/2662948

Daniel H. Wilson

Future Tense Garden of Life

*When machines are in the natural world,
what in the world is still unnatural?*

THE GARDEN OF life is complex, way beyond the ken of humankind. Textbooks say science has only stumbled upon around 10% of all existing species of plants and animals. There could be from 10 to 100 million more. Critters, big and small (mostly small), are living and reproducing and dying as they have for eternity ... without a human being ever so much as laying eyes on them.

It's a wide world out there for a taxonomist.

There are more living things hidden in the wilds than we'll ever know. Life likes to break free and spread. And what you can find will surprise you.

I'm on what I call one of my long jaunts. A jaunt is supposed to be short by definition, but I enjoy the paradox. In fact, I enjoy it just about every weekend and holiday. Come Saturday morning, I pull on my hiking boots, tuck my pant legs into them, and lace them up tight. Out here in the mountains, there's a particular area that's all mine to explore. Miles of government land surrounding some kind of research center. Scrubby deer trails meandering through scalp-prickling heat. Tick-infested pine trees and plenty of poison ivy. But every now and then, you'll find a cool hollow. Caves gouged out of sweating granite. Plenty of microclimates are hiding there in the rough country, off the horse trails and far from where idiot four-wheelers scream and churn mud.

Worse it is out here, the better I like it.

Some people get a thrill jumping from a plane. Out on the lake, kids will get those [CONTINUED ON P. 111]



IMAGE BY ALICIA KUBISTA/ANDRIJ BORYS ASSOCIATES

Computing Reviews

Connect with our Community of Reviewers

“There aren’t too many walk-in technical bookshops where I live; checking out reviewer comments on new books provides me with an equivalent experience.”

- Graham Jenkins



Association for
Computing Machinery

ThinkLoud

www.computingreviews.com

BIG IDEAS START SMALL



**SIGGRAPH
ASIA 2014
SHENZHEN**

CONFERENCE 3 DEC - 6 DEC
EXHIBITION 4 DEC - 6 DEC
**SHENZHEN CONVENTION
& EXHIBITION CENTER**
SA2014.SIGGRAPH.ORG

REGISTER NOW TO SAVE

At SIGGRAPH Asia, meet the people you want to meet. Get to learn, be enthralled and inspired by quality content and exhibits in ways you will never be at work or school.

From now until **15 October 2014, 15:00 Shenzhen, China time** we offer you early bird discounts of up to **20%** if you register online.

Complete registration details can be found at sa2014.siggraph.org/registration-travel.

Sponsored by



Supported by



中国科学院深圳先进技术研究院
SHENZHEN INSTITUTES OF ADVANCED TECHNOLOGY
CHINESE ACADEMY OF SCIENCES

Tsinghua-Tencent
清华-腾讯联合实验室



In Cooperation with



虚拟现实技术与系统国家重点实验室
STATE KEY LABORATORY OF VIRTUAL REALITY TECHNOLOGY AND SYSTEMS

