

COMMUNICATIONS

CACM.ACM.ORG

OF THE

ACM

04/2016 VOL.59 NO.04

Gambling on Bitcoin



How SysAdmins
Devalue Themselves

Are We Headed toward
Another Global Tech Bust?

40 Years of Suffix Trees

Automating Proofs

The Internet and Inequality

Association for
Computing Machinery



Systor2016

**The 9th ACM International
Systems and Storage Conference**

June 6 – 8 • Haifa, Israel

We invite you to attend SYSTOR 2016, an ACM SIGOPS Systems and Storage Conference covering all aspects of systems research. Join us in Haifa from June 6 to June 8 for an exciting and enriching technical program, social events, and informal mingling with leading researchers, from both academia and industry.

Submission is still open for posters and highlight papers, so don't miss the opportunity to feature your work!

Registration is free of charge, and will be open in the near future.

Highlight paper submission deadline: May 6, 2016

Poster submission deadline: April 8, 2016

Program chairs

Mark Silberstein, Technion
Emmett Witchel, University of Texas

General chair

Katherine Barabash, IBM Research

Posters chair

Anna Levin, IBM Research

Steering committee head

Michael Factor, IBM Research

Steering committee

Ethan Miller, University of California
Santa Cruz

Liuba Shrira, Brandeis University

Dan Tsafir, Technion

Dalit Naor, IBM Research

Erez Zadok, Stony Brook University

Sponsored by



In cooperation with



Platinum sponsor



Gold sponsors



Sponsors



www.systor.org/2016/



Applicative 2016

June 1 – 2, 2016
New York City

APPLICATIVE 2016 will bring together practitioners and researchers to share the latest emerging technologies and trends in software development.

The conference consists of two tracks:

APPLICATION DEVELOPMENT will feature speakers from leading technology companies such as Google and Facebook, talking about how they are applying new technologies to the products they deliver. The track covers topics such as reactive programming, micro-services, single-page application frameworks, and other approaches that will help you build more robust applications and do it more quickly.

SYSTEMS SOFTWARE will explore topics that enable systems-level practitioners to build better software for the modern world. The speakers are involved in the design, implementation and support of novel technologies and low-level software supporting some of today's most demanding workloads.

For more information about the conference and how to register, please visit:

<http://applicative.acm.org>



Association for
Computing Machinery

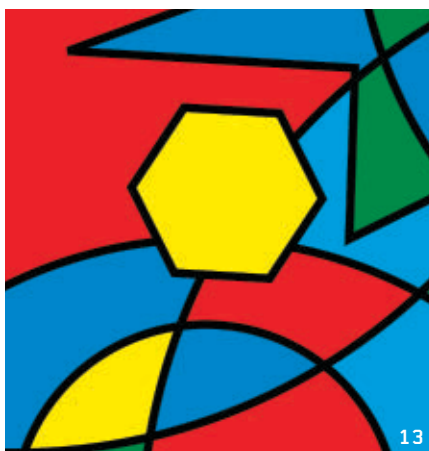
Departments

- 5 **Editor's Letter**
Are We Headed toward Another Global Tech Bust?
By Moshe Y. Vardi
-
- 7 **Cerf's Up**
Enrollments Explode! But diversity students are leaving...
By Vinton G. Cerf and Maggie Johnson
-
- 8 **Letters to the Editor**
Chaos Is No Catastrophe
-
- 10 **BLOG@CACM**
Sampling Bias in CS Education, and Where's the Cyber Strategy?
Mark Guzdial examines a logical fallacy in consumer science education; John Arquilla sees an absence of discussion about the use of information technologies in future conflicts.
-
- 37 **Calendar**
-
- 94 **Careers**

Last Byte

- 96 **Upstart Puzzles**
Sleep No More
By Dennis Shasha

News



- 13 **Automating Proofs**
Math struggles with the usability of formal proofs.
By Chris Edwards
-
- 16 **Existing Technologies Can Assist the Disabled**
Researchers consider how to adapt broadly available technology products for those battling physical impairments.
By Keith Kirkpatrick
-
- 19 **Search Engine Agendas**
Is Google trying to trick you on the way to the polls?
By Gary Anthes
-
- 22 **Marvin Minsky: 1927–2016**
By Lawrence M. Fisher
-
- 25 **A Decade of ACM Efforts Contribute to Computer Science for All**
By Lawrence M. Fisher

Viewpoints



- 28 **Global Computing**
The Internet and Inequality
Is universal access to the Internet a realistic method for addressing worldwide socioeconomic inequality?
By Kentaro Toyama
-
- 31 **Code Vicious**
GNL Is Not Linux
What's in a name?
By George V. Neville-Neil
-
- 33 **Technology Strategy and Management**
The Need for Corporate Diplomacy
Whether global companies succeed or fail often depends on how effectively they develop and maintain cooperative relationships with other organizations and governments.
By Mari Sako
-
- 36 **Viewpoint**
Beyond Viral
The proliferation of social media usage has not resulted in significant social change.
By Manuel Cebrian, Iyad Rahwan, and Alex "Sandy" Pentland

Practice



40

- 40 More Encryption Means Less Privacy**
Retaining electronic privacy requires more political engagement.
By Poul-Henning Kamp
-
- 43 Why Logical Clocks Are Easy**
Sometimes all you need is the right language.
By Carlos Baquero and Nuno Preguiça
-
- 48 How SysAdmins Devalue Themselves**
How to lose friends and alienate coworkers.
By Thomas A. Limoncelli



Articles' development led by **acmqueue**
queue.acm.org

Contributed Articles



50

- 50 How Colors in Business Dashboards Affect Users' Decision Making**
Business dashboards that overuse or misuse colors cause cognitive overload for users who then take longer to make decisions.
By Palash Bera
-
- 58 Multimodal Biometrics for Enhanced Mobile Device Security**
Fusing information from multiple biometric traits enhances authentication in mobile devices.
By Mikhail I. Gofman and Sinjini Mitra

About the Cover:



protocols that are secure even if no trusted third party is available. Cover illustration by Kollected Studio.

Bitcoin, and the technologies behind this digital currency as well as its security issues, is spotlighted in both Research Highlights articles this month. S. Meiklejohn et al. analyze the Bitcoin network, focusing on the growing gap between potential anonymity and actual anonymity. M. Andrychowicz et al. use Bitcoin to design

Review Articles

- 66 40 Years of Suffix Trees**
Tracing the first four decades in the life of suffix trees, their many incarnations, and their applications.
By Alberto Apostolico, Maxime Crochemore, Martin Farach-Colton, Zvi Galil, and S. Muthukrishnan

Research Highlights

- 75 Technical Perspective**
Fairness and the Coin Flip
By David A. Wagner
-
- 76 Secure Multiparty Computations on Bitcoin**
By Marcin Andrychowicz, Stefan Dziembowski, Daniel Malinowski, and Łukasz Mazurek



Watch the author discuss his work in this exclusive *Communications* video.
<http://cacm.acm.org/videos/secure-multiparty-computations-on-bitcoin>

- 85 Technical Perspective**
The State (and Security) of the Bitcoin Economy
By Emin Gün Sirer
-
- 86 A Fistful of Bitcoins: Characterizing Payments among Men with No Names**
By Sarah Meiklejohn, Marjori Pomarole, Grant Jordan, Kirill Levchenko, Damon McCoy, Geoffrey M. Voelker, and Stefan Savage



Watch the author discuss his work in this exclusive *Communications* video.
<http://cacm.acm.org/videos/a-fistful-of-bitcoins>



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

Bobby Schnabel

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 Madden

Co-Chairs, Publications Board

Jack Davidson and Joseph Konstan

Members-at-Large

Eric Allman; Ricardo Baeza-Yates;

Cherri Pancake; Radia Perlman;

Mary Lou Soffa; Eugene Spafford;

Per Stenström

SGB Council Representatives

Paul Beame; Jenna Neefe Matthews;

Barbara Boucher Owens

BOARD CHAIRS

Education Board

Mehran Sahami and Jane Chu Prey

Practitioners Board

George Neville-Niel

REGIONAL COUNCIL CHAIRS

ACM Europe Council

Dame Professor Wendy Hall

ACM India Council

Srinivas Padmanabhuni

ACM China Council

Jianguang Sun

PUBLICATIONS BOARD

Co-Chairs

Jack Davidson; Joseph Konstan

Board Members

Ronald F. Boisvert; Anne Condon;

Nikil Dutt; Roch Guerrin; Carol Hutchins;

Yannis Ioannidis; Catherine McGeoch;

M. Tamer Ozsu; Mary Lou Soffa; Alex Wade;

Keith Webster

ACM U.S. Public Policy Office

Renee Dopplick, Director

1828 L Street, N.W., Suite 800

Washington, DC 20036 USA

T (202) 659-9711; F (202) 667-1066

Computer Science Teachers Association

Mark R. Nelson, 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

cacm-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

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

Publications Assistant

Juliet Chance

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

acmhqhelp@acm.org

Letters to the Editor

letters@cacm.acm.org

WEBSITE

<http://cacm.acm.org>

AUTHOR GUIDELINES

<http://cacm.acm.org/>

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

vic@cacm.acm.org

NEWS

Co-Chairs

William Pulleyblank and Marc Snir

Board Members

Mei Kobayashi; Kurt Mehloth;

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; Carl Landwehr;

Carlos Jose Pereira de Lucena;

Beng Chin Ooi; Loren Terveen;

Marshall Van Alstyne; Jeannette Wing



PRACTICE

Co-Chair

Stephen Bourne

Board Members

Eric Allman; Peter Bailis; Terry Coatta;

Stuart Feldman; Benjamin Fried;

Pat Hanrahan; Tom Killalea; Tom Limoncelli;

Kate Matsudaira; Marshall Kirk McKusick;

George Neville-Niel; Theo Schlossnagle;

Jim Waldo

The Practice section of the CACM

Editorial Board also serves as

the Editorial Board of queue.acm.org.

CONTRIBUTED ARTICLES

Co-Chairs

Andrew Chien and James Larus

Board Members

William Aiello; Robert Austin; Elisa Bertino;

Gilles Brassard; Kim Bruce; Alan Bundy;

Peter Buneman; Peter Druschel; Carlo Ghezzi;

Carl Gutwin; Yannis Ioannidis;

Gal A. Kaminka; James Larus; Igor Markov;

Gail C. Murphy; 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;

Nina Balcan; Dan Boneh; Andrei Broder;

Doug Burger; Stuart K. Card; Jeff Chase;

Jon Crowcroft; Sandhya Dwaekadas;

Matt Dwyer; Alon Halevy; Norm Jouppi;

Andrew B. Kahng; Sven Koenig; Xavier Leroy;

Steve Marschner; Kobbi Nissim;

Steve Seitz; Guy Steele, Jr.; David Wagner;

Margaret H. Wright; Andreas Zeller

WEB

Chair

James Landay

Board Members

Marti Hearst; Jason I. Hong;

Jeff Johnson; Wendy E. MacKay

ACM Copyright Notice

Copyright © 2016 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 \$269.

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 acmhqhelp@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





DOI:10.1145/2892240

Moshe Y. Vardi

Are We Headed toward Another Global Tech Bust?

ENROLLMENTS IN COMPUTING-RELATED undergraduate degree programs are booming, about to establish a new record in North America. There is also a growing demand for computing courses by students who are not computing majors. In the U.S., President Obama recently announced a new \$4 billion initiative “to empower students with the computer science skills they need to thrive in a digital economy.” Of course, this popularity does not come without costs. The growing size of computing degree programs is clearly stressing academic units and putting pressure on the quality of education provided to students. In response to the insatiable demand, academic institutions are raising their level of investment in computing programs, but academic hiring is agonizingly slow!

What is driving the computing-enrollment boom is undoubtedly the global technology boom, epitomized by the global rise of “unicorns”—tech startups with a valuation of at least \$1 billion. *Fortune* magazine recently wrote “The billion-dollar technology startup was once the stuff of myth. Today they’re seemingly everywhere, backed by a bull market and a new generation of disruptive technology.” In January 2016, more than 170 companies were on the unicorn list. It is the dream of joining a unicorn that probably attracts many students to study computing.

We must remember, however, we have witnessed such booms in the past; the history of computing education is a history of booms and busts. Computing-related degree programs were introduced in the mid-to-late 1960s, and grew slowly during the 1970s. The in-

roduction of the IBM PC in 1981 made computing a household phenomenon and triggered the tech and enrollment boom in the 1980s. That boom was ended by a recession of the early 1990s.

By the mid-1990s, the Internet and the World-Wide Web had become household names, launching the dot-com boom; the growing popularity of the “Web” led to the founding of many Internet-based companies, commonly referred to as “dot-coms.” The NASDAQ Composite Index, a U.S. stock-market index that includes many tech companies, more than quintupled between 1995 and 2000. The excitement about the new technology and the demand from the job market led to a growing popularity of computing education; enrollments in North America nearly tripled between 1995 and 2000. Surging enrollments were stressing academic units, forcing institutions to increase staffing in those stressed units.

But by 1999 it was becoming increasingly clear the boom had become a speculative bubble. The NASDAQ Index peaked on March 10, 2000, declining almost 80% over the next two years. Numerous start-up companies went under, bringing down with them several telecommunication companies. The stock-market crash in the U.S. caused the loss of \$5 trillion in the market valuations from March 2000 to October 2002.

At the same time, the Internet and the Web enabled the globalization of software production, giving rise to the phenomenon of offshore outsourcing. There were daily stories in the media describing major shifts in employment that were occurring largely as a result of offshoring. Combined with the impact of the end of the dot-com boom,

these reports raised concerns about the future of computing as a viable field of study and work in developed countries. Computing enrollments in North America went into a steep dive, declining by more than 50% between 2004 and 2009.

I believe it is important to remember this history as we celebrate the rise of the unicorns. There are already some indications the current tech boom may be nearing its end. The media has started commenting “The signs of a new tech bubble are everywhere: easy money, widespread exuberance, hidden leverage, and mass participation by amateur investors.” What turns a bubble into a bust is a change in investors’ psychology. History tells us it does not take much for such a change to occur. The decline of stock markets around the world over the past few months suggests such a change may already be taking place. It is quite likely, also, that a tech bust would bring with it an enrollment bust.

So we should brace ourselves for another global tech and enrollment bust, but also keep in mind the long-term trend. At the trough of 2009, computing enrollments were higher than they were in 1995, before the start of the dot-com boom. Furthermore, a computing degree positions a graduate for solid career opportunities in almost every sector of the global economy, and not only in the tech sector. In the long term, computing is ascendant and will continue to shape the 21st century. Between booms and busts, up we go!

Follow me on Facebook, Google+, and Twitter.

Moshe Y. Vardi, EDITOR-IN-CHIEF

Copyright held by author.



ACM Books



MORGAN & CLAYPOOL
PUBLISHERS

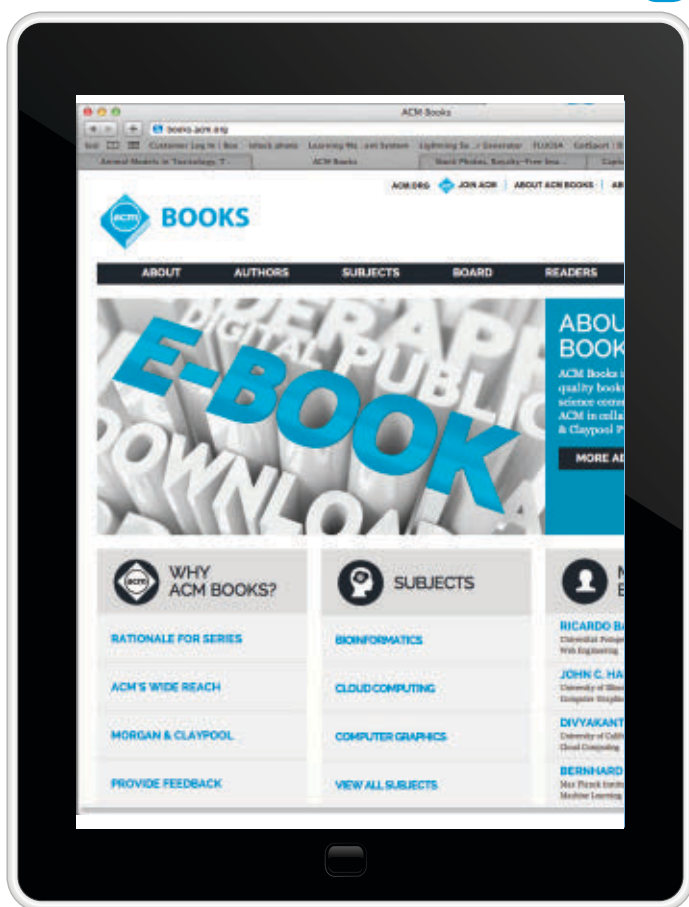
Publish your next book in the ACM Digital Library

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

Enrollments Explode! But diversity students are leaving...

I WANT TO return to a theme I have explored before: diversity in our discipline. To do this, I have enlisted the help of my colleague at Google, Maggie Johnson. We are both concerned the computer science community is still not benefiting from the diversity it could and should have. College students are more interested than ever in studying computer science (CS). There has been an unprecedented increase in enrollment in CS undergraduate programs over the past four years. Harvard University's introductory CS course—CS50—has recently claimed the spot as the most enrolled course on campus.^a An astounding 50% of Harvey Mudd's graduates received engineering degrees this year.^b The Taulbee Study is an annual survey of U.S. Ph.D.-granting institutions conducted by the Computing Research Association. Table 1 from the 2014 Taulbee report^c shows the increases CS departments are experiencing.

While the overall number of students in CS courses continues to increase, the number of women and underrepresented minority students who go on to complete undergraduate degrees is, on average, not growing at all. As noted in Table 2, recent findings show that while these students may begin a CS degree program, retaining them after their first year remains a serious issue.^d

Why is this important? The high-tech industry is putting enormous effort into diversifying its work force.^e First, there is a social justice aspect given the industry demand and the high salaries associated with that de-

mand. Second, high-tech companies recognize if they are going to create truly accessible and broadly useful products and services, a diverse workforce will best create them. Third, with the advent of an increasing amount of software in virtually every appliance ranging from cars to clocks to say nothing of smartphones, we are going to need every bit of system design and programming talent we can find to avoid collapse into a morass of incompatible, uncooperative, and generally recalcitrant devices in our homes, offices, cars, and on or in our persons. Whether we like it or not, programmable devices are much more malleable than electromechanical ones, potentially less expensive to make, and, possibly, easier to update. The Internet of Things is upon us and we need all hands on deck to assure utility, reliability, safety, security, and privacy in an increasingly online world.

What can faculty do in their own departments? There are several simple interventions that can increase student retention in CS programs. Here are some examples:

- Consider student interests when planning assignments.

Table 1. CS enrollment increases reported in 2014 Taulbee Survey.

	2013	2014	% change
B.S. CS Awarded	12,503	14,283	14.2
B.S. CS Enrollments	63,098	80,324	27.3
New B.S. CS Majors	17,207	20,351	18.3

Table 2. CS enrollment decreases reported in 2014 Taulbee Survey.^f

	2013	2014
% Women B.S. CS Graduates	14.2	14.0
% African-American B.S. CS Graduates	3.8	3.2
% Hispanic B.S. CS Graduates	6.0	6.8

- Provide early and consistent feedback on assignments.

- If you have teaching assistants, ensure they are aware of the best practices you follow.

- Emphasize that intellectual capacity—like a muscle—increases with effort. (You are not born with the ability to program!)

- Tell students about conferences and the benefits of attending conferences for targeted support groups.

- Women and minority students often believe they are not performing well, even when their grades tell a different story. It is important to tell women and minority students they will succeed if they stay.

- Be open and accessible to students. You may not know who needs a sounding board, but generally letting students know you are available can make it easier for them to ask for help or guidance.

- Consider helping to form student chapters of ACM-W and IEEE.

- A list of constructive steps, created by NCWIT, is here.^g

Faculty can make a huge difference in retaining our diversity students. As leaders in the CS field, your actions and words have a profound impact. When we lose the interest of a significant part of our diverse society, we suffer irretrievably. We cannot even calculate the opportunities we may have lost for the CS discipline. The next potential scientific breakthrough or blockbuster business might have come from someone whose interest we failed to keep. Please join us in highlighting this important opportunity and sharing these and your own solutions with your faculty. □

^g <https://www.ncwit.org/resources/top-10-ways-retain-students-computing/top-10-ways-retain-students-computing>

Vinton G. Cerf is vice president and Chief Internet Evangelist at Google. Maggie Johnson is Director of Education and University Relations at Google.

Copyright held by authors.

^a <http://www.thecrimson.com/article/2014/9/11/cs50-breaks-enrollment-records/?page=single>

^b <https://www.hmc.edu/about-hmc/2014/05/20/harvey-mudd-graduates-landmark-class/>

^c <http://cra.org/crn/wp-content/uploads/sites/7/2015/06/2014-Taulbee-Survey.pdf>

^d http://cra.org/crn/2015/05/booming_enrollments_what_is_the_impact/

^e <https://www.google.com/diversity/index.html>

^f <http://archive2.cra.org/uploads/documents/resources/crndocs/2013-Taulbee-Survey.pdf>

Chaos Is No Catastrophe

I APPRECIATED PHILLIP G. Armour's use of coupled pendulums as an analogy for software project management in his The Business of Software column "The Chaos Machine" (Jan. 2016) but would like to set the record straight on a few technical points. Chaos is already being exhibited when Armour's machine performs smoothly, in the sense future behavior is inherently unpredictable. What happened when the machine made a hop was not that it "hit a chaos point" but apparently some "resonance disaster" that caused it to exceed the range of operation for which it was built. Moreover, "turbulence" is not an appropriate description in this context, as it describes irregular movement in fluid dynamics. Chaotic behavior does not require three variables. The most basic instance—the double pendulum, with one rod hanging from the end of another rod—involves only two variables. And the technological solution for chaos is "control," which applies to software project management as well. Setting a project in motion, even one as simple as a single pendulum, then leaving it unattended, is not a good idea. A good case in point for how an unattended project can become chaotic is the construction of the new Berlin airport.

Günter Rote, Berlin, Germany

Author's Response:

Rote's point is well taken. The word "chaos" in general usage simply connotes disorder and unmanageability, and I was using that meaning rather than a more formal characterization—something beyond both my skill and my intent. Showing the device generated a lot of interesting discussion in the workshop—which was the point. And as Rote graciously acknowledges, it was an analogy for software-project management rather than a physics experiment.

Phillip G. Armour, Deer Park, IL

We Are Our Machines

I was encouraged by *Communications* addressing such weighty issues

as "lethal autonomous weapon systems" through Moshe Y. Vardi's Editor's Letter "On Lethal Autonomous Weapons" (Dec. 2015) and related Stephen Goose and Ronald Arkin "Point/Counterpoint" debate "The Case for Banning Killer Robots" in the same issue. Computing professionals should indeed be paying attention to the effects of the software and hardware they create. I agree with those like Goose who say use of technology in weapons should be limited. America's use of military force is regularly overdone, as in Iraq, Vietnam, and elsewhere. It seems like making warfare easier will only result in yet more wars.

ACM should also have similar discussions on other contentious public issues; for example, coal-fired power plants are probably today's most harmful machines, through the diseases they cause and their contribution to climate change.

ACM members might imagine they are in control of their machines, deriving only their benefit. But their relationship with machinery (including computers) is often more like worship. Some software entrepreneurs strive even to "addict" their users to their products.¹ Computing professionals should take a good look at what they produce, not just how novel or efficient or profitable it is but how it affects society and the environment.

Scott Peer, Glendale, CA

Reference

1. Schwartz, T. Addicted to distraction. *New York Times* (Nov. 28, 2015); http://www.nytimes.com/2015/11/29/opinion/sunday/addicted-to-distraction.html?_r=0

Author Responds:

I agree with Peer that Communications should hold discussions on public-policy issues involving computing and information technology, though I do not think ACM members have any special expertise that can be brought to bear on the issue of coal-fired power plants.

Moshe Y. Vardi, Editor-in-Chief

Liability and Braces

The Letters pages "Let the Liable Pay" (Jan. 2016) included a letter under that title by Jonathan Handel on software liability followed by a letter by Jamie Hale under the subhead "Hold the Braces and Simplify Your Code" on ways to make code in languages that use braces more readable. (These languages seem to follow a design principle emphasizing ease of writing programs over ease of reading them, so one would think developers interested in creating readable code would simply avoid using such languages.)

Consider the software that flies commercial airliners, in which an error can lead to significant liability, as measured in billions of dollars, not to mention the deaths of hundreds of people. Producing and certifying software of the required level of correctness, reliability, and robustness is expensive, and airplane manufacturers seek ways to minimize that expense. For economic and safety reasons, both Airbus and Boeing use the same braces-free language for their software, and millions of people trust their lives to it. It follows then, that in a reasonable world, all software that should be correct, reliable, and robust would be implemented in such a language and to such standards. All safety-critical software, as in cars, trains, and other vehicles, and that operates medical devices, should be required to be certified to similar standards. Such software also runs the Internet and all financial and commercial sites; for privacy protection, all software on phones and mobile devices; and, since software correctness cannot be guaranteed if run on an incorrect foundation, all operating systems correct software would run on. That this is usually not the case represents a serious condemnation of the software-engineering profession.

As for Hale's recommendation to write short blocks of code, while that is good advice, the labeling of termi-

nator markers, which he dismissed as unnecessary, is still a good idea. The language used in commercial flight software requires terminating `if` statements with `end if;`, records with `end record;`, case statements with `end case;`, and so on. Many constructs can be labeled with a name that then appears in the terminator—`end loop Name;`. This is because there is evidence block terminators are a common source of error, and such labeling reduces such errors, even for short blocks. Short terminators (such as a brace) are a greater source of error than long terminators (such as `end`), which is why braces are avoided. Safety and readability are designed into the language rather than as an afterthought.

The economic pressures to expand liability to more software would thus be expected to lead to a reduction in the use of languages with braces, along with an associated increase in readability.

Jeffrey R. Carter, Mesa, AZ

A Log Graph Too Far?

I really enjoyed George V. Neville-Neil's article "Time Is an Illusion Lunchtime Doubly So." (Jan. 2016) and am a big fan of his Kode Vicious columns as well. However, I could not fathom the intriguing figure (see

it here) in the article, labeled simply "PTP log graph," nor could I find any reference to it in the text that might shed light on what property of the Precision Time Protocol it is supposed to illustrate. Not being an electrical engineer myself, I thought it might be something obvious to someone in the field, yet a friend with a Ph.D. in electrical engineering was equally flummoxed. It was such an interesting chart I am keen to understand what it means—if I can. Prof. Neville-Neil, can you enlighten me?

John Beresniewicz, Half Moon Bay, CA

Author Responds:

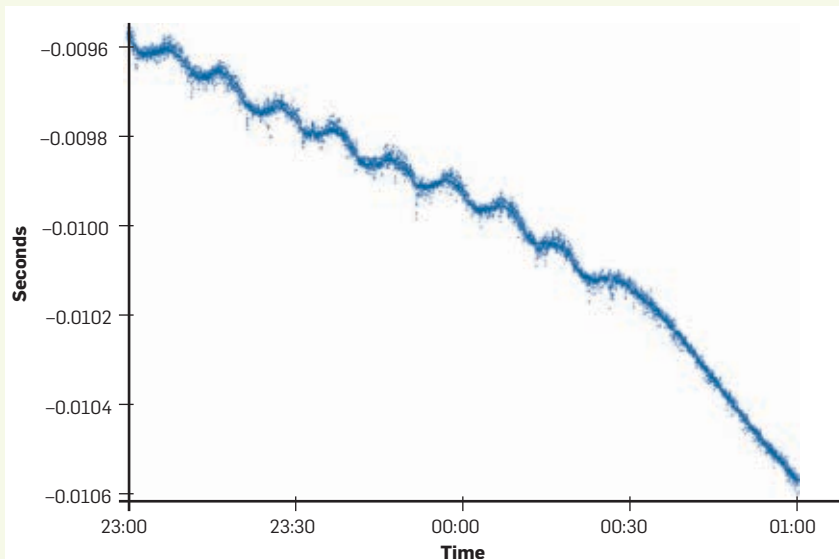
The PTP Log Graph in the figure showed the offset of a system clock that is not regulated by an outside time source (such as NTP and PTP). Without an outside time source, the clock wanders away from where we would expect it to be if the system's crystal oscillator was more stable, which it is not.

George V. Neville-Neil, Brooklyn, NY

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.

© 2016 ACM 0001-0782/16/04 \$15.00

PTP log graph.



Coming Next Month in COMMUNICATIONS

A Survey of Robotic Musicianship

ACM's 2016 General Election

The Challenges of Partially Automated Driving

Parallel Graph Analytics

Static Presentation Consistency Issues in Smartphone Mapping Apps

How to Increase the Security of Smart Buildings?

Delegation as Art

On the Naturalness of Software

Plus the latest news about light chips, AI-enhanced security, and coding as sport.

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.



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

DOI:10.1145/2892708

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

Sampling Bias in CS Education, and Where's the Cyber Strategy?

Mark Guzdial examines a logical fallacy in consumer science education; John Arquilla sees an absence of discussion about the use of information technologies in future conflicts.



Mark Guzdial
"The Inverse Lake Wobegon Effect"

<http://bit.ly/1PpjWmn>
January 11, 2016

Every episode of the radio variety show "A Prairie Home Companion" includes a segment in which host Garrison Keillor tells stories from his mythical hometown, Lake Wobegon. Each segment ends with, "Well, that's the news from Lake Wobegon, where all the women are strong, all the men are good looking, and all the children are above average." That notion, that "all the children are above average," is an example of what is known as the Lake Wobegon Effect (<http://bit.ly/1JYKFKr>), also known as "illusory superiority" (<http://bit.ly/23JkX33>).

The Lake Wobegon Effect is where we consider the small sample in our experience superior to the population overall. A concrete example: 80% of drivers consider themselves above-average drivers. Obviously that cannot

be true, but it is common that we think what we experience is above average.

The Inverse Lake Wobegon Effect is a term I am coining for a fallacy that I see sometimes in computer science (CS) education: we sample from a clearly biased source and assume the sample describes the overall population. We know we are observing a superior sample, but act like we are getting a randomly distributed sample. This is a form of sampling bias (<http://bit.ly/1R358iK>).

I introduce the term in a book I just published with Morgan & Claypool, *Learner-Centered Design of Computing Education: Research on Computing for Everyone*. One example of the Inverse Lake Wobegon Effect in CS education is assuming a successful undergraduate introductory curriculum will be similarly successful in high school. Students in undergraduate education are elite. In the U.S., undergraduates are screened in an application process and are in the top half of most scales (such as intellectual achievement and wealth). Elite students can learn under conditions in which

average students might not succeed, which educators call *aptitude-treatment interactions* (<http://bit.ly/1PiaGB6>).

Consider Bennedsen and Caspersen's work on predictors for success in introductory computing (<http://bit.ly/1TEkY3W>). Students in undergraduate education have better math grades and more course work than average students in high school, and both factors predict success in introductory CS courses. Think about the role of algebra in programming. There are high schools in Atlanta, GA, where less than half the students pass algebra. The same CS curriculum that assumes success with algebra is unlikely to work well for undergraduate and high school audiences.

Imagine a highly successful undergraduate introductory computing curriculum in which 80% of the students succeed; that is, 80% of students from the top half of whatever scale we are talking about. The same curriculum might fail for 60% of the general population.

We see a similar sampling error when we talk about using MOOC data to inform our understanding of learning. The edX website says it offers a platform for "exploring how students learn." Students who take MOOCs are overwhelmingly well-educated, employed, and from developed countries—characteristics that describe only a small percentage of the overall population. We cannot assume what we learn from the biased sample of MOOC participants describes the general population.

Psychologists are concerned many of their findings are biased because they oversample from "WEIRD" students:

They found people from Western, educated, industrialized, rich and democratic (WEIRD) societies—representing up to 80% of study participants, but only 12% of the world’s population—are not only unrepresentative of humans as a species, but on many measures they are outliers. (<http://bit.ly/1S11gQo>).

It is easy to fall prey to the Inverse Lake Wobegon Effect. Those of us who work at colleges and universities only teach undergraduate and graduate students. It is easy for us to believe those students represent all students. If we are really aiming at computing for *everyone*, we have to realize we do not see *everyone* on our campuses. We have to design explicitly for those new audiences.



John Arquilla
**“Toward a Discourse
 on Cyber Strategy”**

<http://bit.ly/1J6TPE9>
 January 15, 2016

While cyber security is a topic of discussion all over the world today—a discourse shifting in emphasis from firewalls to strong encryption and cloud computing—little is heard about broader notions of cyber strategy. Efforts to understand how future conflicts will be affected by advanced information technologies seem to be missing—or are taking place far from the public eye.

When David Ronfeldt and I first published *Cyberwar Is Coming!* (<http://bit.ly/1PAL6uW>) nearly a quarter-century ago, we focused on overall military operational and organizational implications of “cyber,” not just specific cyberspace-based concerns. It was our hope a wide-angled perspective would help shape the strategic conversation.

Sadly, it was not to be. Forests have been felled to provide paper for the many books and articles about how to protect information systems and infrastructure, but little has emerged to inform and guide future development of broader strategies for the cyber era.

There have been at least a few voices raised in strong support of a fresh approach to strategic thought in our time—interestingly, with some of the best contributions coming from naval strategists. Among the most trenchant insights were those of two senior U.S. Navy officers. Vice Admiral Arthur Cebrowski, with his concept of “network-

centric warfare,” emphasized this period of technological change would favor the network form of organization. Admiral Bill Owens, in his *Lifting the Fog of War* (<http://bit.ly/1SYvEuL>), argued for extensive information gathering and sharing in what he called a “system of systems.” Both were writing over 15 years ago, and their respective visions proved to be a bit too cutting-edge to gain much traction.

Around the same time, some astute naval officers in China were doing much the same. Then-Captain Shen Zhongchang, the People’s Liberation Army Navy’s R&D director, along with a few staff officers, appreciated the importance of networks and systems thinking—keying on the former as principal targets in future conflicts and directing their energies on battle doctrines. They understood huge increases in the information content of weaponry virtually decoupled range from accuracy, making possible “remote warfare” and demanding dispersal rather than concentration of forces in future wars.

Zhongchang’s team played a measurable role in shaping Chinese strategic thought. Overall, though, there has been little open debate of ideas about the age of cyberwar in world strategic circles. How different this is from the international discourse that arose over the prospect of nuclear war. In the first decade of the atomic age, a range of strategic ideas shaped lively debates. In the U.S., enthusiasm for nuclear weapons among senior policymakers led to ideas about waging preventive wars against enemies before they could acquire such capabilities. Thankfully, scholars and others involved in security affairs rose up in protest and, in 1954, President Eisenhower publicly renounced the idea the U.S. would ever wage preventive nuclear war.

Other countries were ahead of the U.S. on this point, including the then-Soviet Union, and even France, where Charles de Gaulle put the notion of endless nuclear arms racing to rest with the formulation all that was needed was an “arm-tearing-off” capacity for deterrence to work well. Mao Zedong adopted this view, too; so have most others who have developed nuclear weapons. Eventually, in part because of public debate—and sometimes because of protests in both countries—Moscow and

Washington came around to this view, and arms racing turned into the nuclear arms reductions we see today.

This is not the case with cyber. There is a raging arms race in virtual weaponry, secretly, in many countries, with concomitant plans for preemptive, preventive, and other sorts of Pearl Harbor-like actions. The potential for “mass disruption” (as opposed to mass *destruction*) is generally the focus of these efforts. The results could be costly if these ideas were ever acted upon. As Scott Borg, director of the U.S. Cyber Consequences Unit, noted: “An all-out cyber assault can potentially do damage that can be exceeded only by nuclear warfare.”

Yet instead of an outcry about this looming threat and a thoughtful discourse about how to bring these capabilities under control, efforts to develop ever-more-sophisticated weaponry of this sort proceed unabated. In some places, the complacency in the face of the potential threats is staggering. Witness the comments of the current U.S. “cyber czar,” Michael Daniel: “If you know about it, [cyber is] very easy to defend against.” In an age where the world has repeatedly seen how vulnerable commercial enterprises are, and where even sensitive information guarded by governments is breached, the statement that cyber attack is “easy to defend against” rings all too hollow.

What is needed now is a lively discourse on cyber strategy. It should probably begin with consideration of whether offense or defense dominates, as parsing the peril in this way will affect the larger debate about continuing the cyber arms race or, instead, searching out various ways to craft sustainable, behavior-based international cyber arms control agreements. The wisdom or folly of using cyber weaponry in preemptive or preventive actions—à la Stuxnet—should also be openly debated.

In an earlier era, atomic scientists played central roles in guiding and informing the key nuclear debates—in the military, government, and among the mass public. In this era, it may be up to computer scientists and information technology experts to provide a similar service—and *now* is the time. ■

Mark Guzdial is a professor at the Georgia Institute of Technology. John Arquilla is a professor at the U.S. Naval Postgraduate School.

© 2016 ACM 0001-0782/16/04 \$15.00

SHAPE THE FUTURE OF COMPUTING.

JOIN ACM TODAY.

ACM is the world's largest computing society, offering benefits and resources that can advance your career and enrich your knowledge. 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
- ☐ Student Membership with ACM 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 _____

Purposes of ACM

ACM is dedicated to:

- 1) Advancing the art, science, engineering, and application of information technology
- 2) Fostering the open interchange of information to serve both professionals and the public
- 3) Promoting the highest professional and ethics standards

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

1-800-342-6626 (US & Canada)
1-212-626-0500 (Global)

Hours: 8:30AM - 4:30PM (US EST)
Fax: 212-944-1318

acmhelp@acm.org
acm.org/join/CAPP

Automating Proofs

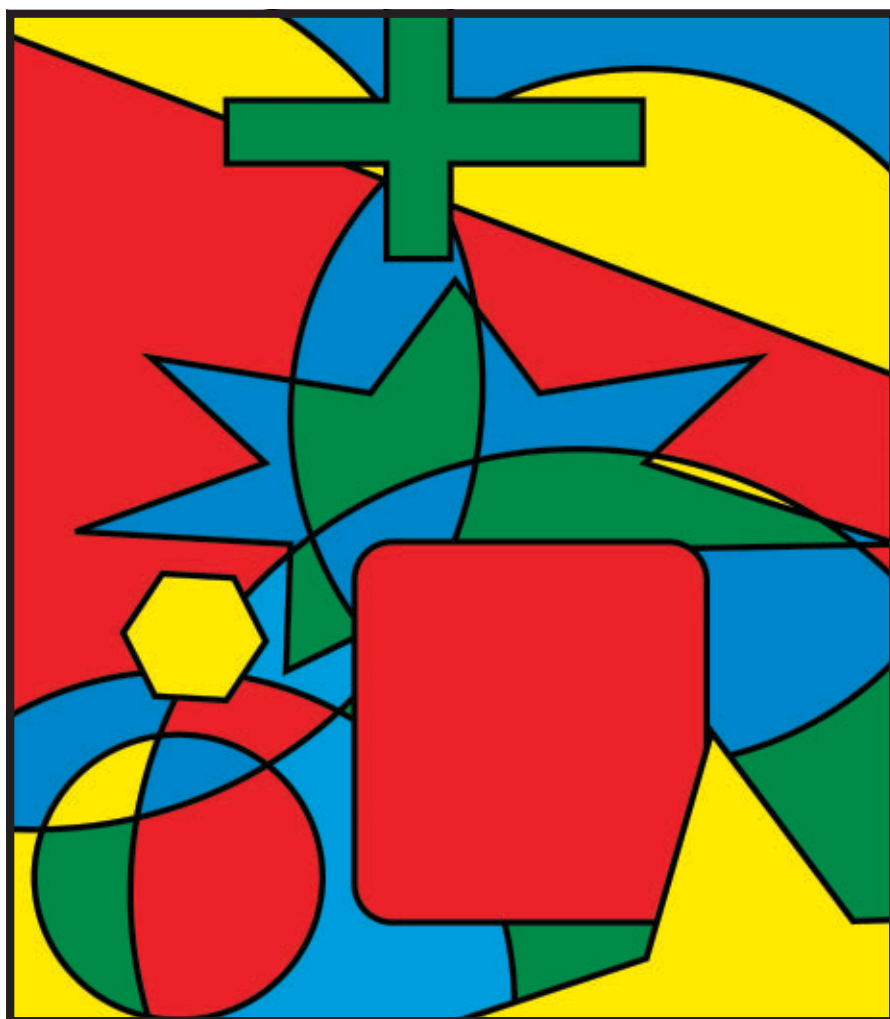
Math struggles with the usability of formal proofs.

OVER THE PAST two decades, mathematicians have succeeded in bringing computers to bear on the development of proofs for conjectures that have lingered for centuries without solution. Following a small number of highly publicized successes, the majority of mathematicians remain hesitant to use software to help develop, organize, and verify their proofs.

Yet concerns linger over usability and the reliability of computerized proofs, although some see technological assistance as being vital to avoid problems caused by human error.

Troubled by the discovery in 2013 of an error in a proof he co-authored almost 25 years earlier, Vladimir Voevodsky of the Institute for Advanced Study at Princeton University embarked on a program to not only employ automated proof checking for his work, but to convince other mathematicians of the need for the technology.

Jacques Carette, assistant professor in the department of computing and software at McMaster University in Ontario, Canada, and a promoter of the idea of ‘mechanized mathematics’, says, “There are both technical and social forces at work. Even though mathematicians doing research are trying to find new knowledge, they are quite conservative about the tools they



An example of a four-color map. The four-color map theorem says no more than four colors are required to color the regions of a two-dimensional map so no two adjacent regions have the same color.

use. Tools can take some time to adopt: tools such as (the algebra systems) Maple and Mathematica took a solid 20 years before they became pervasive.

“Personally, once I got over the hurdle of learning how these things work, it now speeds me up. I can be bold in my conjectures and the computer will tell me that I’m wrong. The computer is good at spotting problems in the small details: things that humans are typically really bad at.”

Jeremy Avigad, a professor in the philosophy and mathematical sciences departments at Carnegie Mellon University, says of formal proof technology: “I believe that it will become commonplace. It’s a natural progression. We care that our math is precise and correct and we now have a technology that helps in that regard. But the technology is not yet ready for prime time. There is a gap: it’s usability.”

For mathematicians working on problems that seemed insurmountable if tackled purely by hand, the usability gap was less of an issue than trust in the results by others. The length and complexity of a proof of the 1611 conjecture by Johannes Kepler on the most efficient method for packing spheres that was developed by Thomas Hales while working at the University of Michigan in the late 1990s led to a reviewing process that took four years to complete. Even after that length of time, reviewers claimed they could only be 99% certain of the proof’s correctness. To fully demonstrate the proof’s correctness, Hales started a collaborative project called FlySpeck built on the use of automated proof-checking software. The group completed its work early in 2015 with the publication of a paper that described the process used.

In building the original proof of the Kepler conjecture, Hales and his colleagues had to develop software that performed computation in a way that lent itself to building a reliable proof. A large part of the proof lies in a lengthy series of inequalities that needed to be demonstrated using computation. Rather than rely on conventional floating-point arithmetic and the imprecision that introduces, the group had to develop software to perform interval arithmetic and automated differentiation using Taylor expansions “so we were able to get

“Once I got over the hurdle of learning how these things work, it now speeds me up. I can be bold in my conjectures and the computer will tell me that I’m wrong.”

rigorous upper and lower bounds,” Hales explains.

Similar issues of trust greeted the first computerized proof of the theorem that argued only four colors are needed to distinguish between adjacent regions on a 2D map. Forty years ago, working at the University of Illinois at Urbana-Champaign, Kenneth Appel and Wolfgang Haken developed computer programs to demonstrate there were no counterexamples to the theorem. Still, the programs were tedious to check by hand.

In 2005, Georges Gonthier of Microsoft Research and Benjamin Werner of French research institute Inria used a computerized proof assistant, Coq, to develop a proof that did not rely on counterexamples generated by programs. Moving to a position where the Coq kernel could be trusted involved less manual inspection.

“Most proof systems are built on a very small trusted base. In terms of what is carrying out inferences, that code base is very small. Then on top of that, you write very complicated software that parses the code, but ultimately, it calls this very small thing that actually checks the proof,” Avigad explains. “People have also built mechanisms where you can have a complex proof system that checks the proof and then have another checker check the proof independently. The probability of there being a mistake in one and the other having the same mistake is very low.”

Avigad adds: “Mathematicians are

great pragmatists. People will do what they need to do and if they see something is useful, they will apply it. But right now the feeling is that there is a very steep learning curve. You spend some time learning logic and formal methods, then spend some time learning to prove what are really quite trivial formal theorems.”

A key stumbling block is the level of detail required by the computer to prepare a formal proof. Despite the apparent rigor of mathematical notation and language, “there is a lot that’s left implicit,” Avigad says. “In a proof, there are many small inferences. Specifying each one explicitly is not something we typically do.”

Automated proof assistants need far more detail from the mathematician, says Tobias Nipkow, a member of the theorem-proving group at the Technical University of Munich’s department of information technology. “We are still at the stage where in order to formalize anything non-trivial, you frequently have to include a significant amount of foundational material that a mathematician would simply take for granted.”

One approach is to build online archives of formalized mathematical knowledge that could be accessed by proof tools. An example of such a repository is the Archive of Formal Proofs (<http://afp.sourceforge.net/>), which uses Sourceforge to hold a collection of proof libraries and examples written for the Isabelle prover software, one of the tools used for the FlySpeck project.

“Equally important are more powerful automatic proof procedures. This would enable mathematicians to take bigger steps in their proofs and reduce the tedium of computer-assisted proofs,” Nipkow adds.

Carette says a key problem in developing more usable proof assistants with the help of online repositories is the breadth of techniques needed to produce the software. “The technical challenges involved in these two enterprises are quite different. They tend to be done by different people.”

A further issue is the tendency for branches of mathematics to form silos that use quite different techniques to approach the problem of building proofs. Large-scale proofs such as FlySpeck rely on extensive computation,

as well as the analytical steps associated with mathematical proofs.

"Systems to do computation have been entirely different to those that do proofs," says Carette.

Even with trust in the results and increased usability of the tools used to develop computerized proofs, what happens if the proof is only ever processed by a machine, without calling for a human to understand what it contains?

Says Avigad, "Formal verification can cover the correctness part of a proof, but it doesn't convey the knowledge."

Adds Carette, "Being able to understand the proof is a real issue. But a trend that's started recently is to write a paper that says in the introduction, 'this paper is a reformulation of a set of results'. You start with a set of proofs that are machine-checked, but the paper you write is an explanation of what is going on. That appears to work. People feared the explanation step would go away, but it hasn't."

Even with formal verification as a basis, there remain fears that errors will still creep into proofs, says Avigad. "People in the community are very sensitive to this, but software has been built on an architecture that contains safeguards designed to maintain correctness."

As the technology becomes more widespread, a rapid shift in acceptance could take place in certain sub-disciplines, if not in mathematics as a whole, Nipkow says. "The ease with which mathematics can be formalized with a proof assistant depends to some degree on the subject area. As a result, in some areas we may see more of an enthusiasm for such formalizations develop. "I don't really see that yet. But, I do expect to see further, isolated formal landmark proofs," Nipkow says, pointing to examples such as Hales' proof of the Kepler conjecture and the work on the four-color map theorem.

Carette adds, "You may see situations where, at the highest level, if the paper doesn't come with a computer-assisted proof, it's likely that the result will be rejected. A rapid change from almost nobody using the technology to computer-assisted proofs becoming a de facto standard? It can happen."

Having better understanding among those writing the software for mathe-

maticians will help improve the spread of the technology, Carette says. "I'm not sure we understand the process that mathematicians go through when they do their work to be able to predict which technical advance will provide the biggest win. What is being discovered more and more now is that proof alone is insufficient."

"Mathematics is made up of a lot of things: proof; computation; knowledge management; plotting graphs. All of these have to come together to make a useful system."

What is required, in effect, is a successor to the Maple software, originally developed in 1980 at the University of Waterloo to support automated proofs with an extensive set of related tools.

However, there is a problem with incentivizing the development of more complete automated mathematical assistants. "The work is slowed down by the amount of development effort needed that isn't rewarded. If you don't get a paper out of a development, academia doesn't recognize it as work. That's changing, but slowly," says Carette, pointing to the work performed by research institutes, where more time is available for staff to develop software.

Avigad says he expects computers ultimately "will help us find new ways to do mathematics that are currently too complex for us to do manually. It's a very exciting time. Over the decades, mathematics will change." **C**

Further Reading

Hales, T. et al

A Formal Proof of the Kepler Conjecture, *ArXiv* (2015)
<http://arxiv.org/abs/1501.02155>

Avigad, J., and Harrison, J.

Formally Verified Mathematics, *Communications of the ACM*, April 2014, Vol. 57, No. 4

Blanchette, J.C., Haslbeck, M.,

Matichuk, D., Nipkow, T.,

Mining the Archive of Formal Proofs, *Proceedings of the International Conference on Intelligent Computer Mathematics (CICM)* 2015, July 2015

Gonthier, G.,

Formal Proof – The Four Color Theorem, *Notices of the American Mathematical Society*, Vol. 55, No. 11, p1382 (2008)

Chris Edwards is a Surrey, U.K.-based writer who reports on electronics, IT, and synthetic biology.

© 2016 ACM 0001-0782/16/04 \$15.00

ACM Member News

DEVADAS SHIFTS FOCUS ON HARDWARE TO COMPUTER SECURITY



As a child in India, Srinivas Devadas knew he would pursue an advanced degree and become an

educator because "It's coded into my DNA. One side of my family got engineering degrees, and the other liberal arts." Engineering and technology were the clear winners. "I played with circuit boards and discrete transistors, and built radios and walkie-talkies at age 11," he recalled.

Devadas earned his bachelor's degree in electrical engineering from the Indian Institute of Technology in Madras, and his master's degree and doctorate, both also in that discipline, from the University of California, Berkeley. He joined the faculty of the Massachusetts Institute of Technology in 1988, and now holds that institution's Edwin Sibley Webster Professor of Electrical Engineering and Computer Science Chair.

"Fundamentally, I'm a parallel processing hardware designer. I look for ways to improve the hardware to ameliorate application, software performance, and battery consumption."

Yet computer security has been his main focus for a decade. He developed Aegis, a secure chip incorporating a silicon biometric technology called Physical Unclonable Function (PUF), which makes Radio-Frequency Identification (RFID) chips "unclonable" by dynamically generating a nearly unlimited number of unique volatile keys for each chip.

In 2005, Devadas and Tom Ziola co-founded Verayo, in San Jose, CA, to productize Aegis. Yet, he says, "I'm still my inner child; the chip isn't much more complicated than my childhood electronics."

Says Devadas, "I love two things: sports and research. I learn something new every couple of years."

—Laura DiDio

Existing Technologies Can Assist the Disabled

Researchers consider how to adapt broadly available technology products for those battling physical impairments.

MORE THAN 20% of U.S. adults live with some form of disability, according to a September 2015 report released by the U.S. Centers for Disease Control and Prevention. The latest generation of smartphones, tablets, and personal computers are equipped with accessibility features that make using these devices easier, or at least, less onerous, for those who have sight, speech, or hearing impairments. These enhancements include functions such as screen-reading technology (which reads aloud text when the user passes a finger over it); screen-flashing notification when a call or message comes in for the hearing impaired; and voice controls of basic functions for those who are unable to physically manipulate the phone or computing device's controls.

Other technologies that can help the disabled have or are coming to market, and not all of them are focused simply on providing access to computers or smartphones. Irrespective of the accessibility provided, most market participants agree more needs to be done to help those with disabilities to fully experience our increasingly digital world.

Mobile Access Technologies

Thanks to the ubiquity of PCs, smartphones, and tablets, a significant number of accessibility-related applications and enhancements are in use today. The aforementioned screen readers are interfaces that have been developed to make it easier for people to view and interact with content on their computers, and vary in complexity and features offered. Screen reader software can range in cost from free, such as the Orca software that works with applications such as OpenOffice, Firefox, and the Java platform, to for-pay options such as Serotek's



iBraille Notes allows the vision impaired to type Braille on the iPhone, iPad, and iPod Touch, with audio feedback.

System Access, which provides access to Microsoft Windows, Outlook, Adobe Reader, and Skype.

While each screen reader features its own command structure, most are designed so the operator can send commands via keystroke or a braille display to the computer and instruct it to read text out loud, and then have the computer's voice synthesizer read a line or full screen of text. More advanced features allow for voice or braille control over spellchecking, verifying the position of a cursor, or modifying content.

Other technology designed to help people who cannot see interact more easily with their computing devices includes a software/hardware solution that reads content on the computer and then provides output in braille. The software captures words and images from web pages, then converts that content into a digital version of braille, which is then used to electromechanically control a set of pins contained in cells, which are arranged side-by-side.

When a blind person touches each cell, the pin configurations are reconfigured to represent the next line of the text being read. Some examples of these types of refreshable braille displays include the 40-cell Freedom Scientific's Focus 14's Ultra-Portable Wireless Braille Display (\$1,295) and the larger, 80-cell Alva BC860 Braille Display (\$8,995), which offers simultaneous connectivity with two computers or a computer and a smartphone.

Manufacturers of smartphones have not ignored this market, either. Apple patented a technology for "hover-sensitive devices" in 2011 that could detect hand gestures made near the screen. Rival Samsung has provided support for its Airview feature, which lets users enlarge text or activate apps without touching the screen, on certain Galaxy devices running Google Android.

Frederic Pollmann, a researcher at the University of Bremen's Digital Media Group in Germany, has been working on the issue of accessibility and smart de-

vices, which led to the development of a mobile app called HoverZoom. HoverZoom is a finger-detection function that significantly enlarges the area of the keyboard under one's finger to make the underlying keyboard more readable and easier to use. This enables people who have issues with fine motor control, such as Parkinson's disease sufferers, to more easily use the device since they do not need to place their fingers directly on a small surface to activate a key.

The app addresses a significant issue that likely will become more prevalent as the Baby Boomer generation moves into old age: fading or failing capabilities.

"We have a use-case where people are used to using a smartphone now, and don't need glasses," Pollmann says. "But in five years, they may need them in order to use the smartphone."

Accessing Life

A key concern of both researchers and educators has been the focus on technology for entertainment or productivity, perhaps in lieu of focusing on tools that help people with daily tasks and activities. While the growing use of technology in game consoles has helped drive development of assistive technologies, some researchers believe not enough is being done to figure out how such technologies can be specifically adapted to help those with significant disabilities.

"When we see we already have technology like the Kinect, which we use for dancing games, it's sad to see that no one is thinking about how we can put this technology to use for better reasons," says Markus Pröll, founder of Xcessity Software Solutions, a Graz, Austria-based developer of human-computer interaction technologies.

Pröll and his team developed assistive technology using the Microsoft Kinect that allows severely disabled people to access a computer completely hands-free. By using the Kinect's sensors to track a person's head movements and facial expressions, the movement impaired can control the mouse-cursor and mouse buttons without using their extremities.

Other developers also are working on applications designed to address specific, real-world problems faced by those with disabilities. Digit-Eyes, an iOS application that creates QR code labels that can be affixed to everyday items

and then read by the autofocus camera included on the iPhone, is one example of how technology already embedded in today's devices simply needs to be adapted to focus on accessibility issues, such as by printing item-specific labels on household items like coffee cups, telephones, or even toothbrushes.

Meanwhile, robotics researchers at Carnegie Mellon University (CMU) are developing assistive robots to help blind travelers. Starting with a humanoid robot called Baxter made by ReThink Robotics of Boston, the researchers modified Baxter to provide both physical and visual assistance at an information desk in a busy transit center when human workers are not available. The ultimate goal for the project is to integrate the robot with a smartphone navigation app and then, eventually, to introduce mobile robots that could physically guide blind people in a manner similar to guide dogs.

Pröll says for those dealing with disabilities, today's largest hurdles are not simply technological, but are related to overcoming issues with interfaces. "We have all the technology in place that is needed to gather any signals you can imagine from the body," he says. "Of course, I'm talking about brain-computer interface research and such, but generally, we can get so much data from any body movements, to eye movement and eye tracking. [But] it's always an interfacing problem with existing applications."

Pröll contends for those with severe disabilities such as ALS, traditional input and control interfaces such as touchscreens and even voice commands are impossible to use, and require more sophisticated alternatives, such as eye tracking and brain-wave measurement.

Accessing Health

Kyle Rector, a graduate student at the University of Washington, developed a software application called Eyes-Free Yoga to assist and guide blind or sight impaired people into six yoga positions, such as the Warrior I and Tree positions. Eyes-Free Yoga uses geometry to calculate the proper angles needed to complete a yoga pose, and then reads the person's body positioning using the Kinect's cameras and skeletal-tracking technology. The application compares the user's body

Milestones

2 Papers Share Dijkstra Prize

The E.W. Dijkstra Prize Committee granted the 2015 Edsger W. Dijkstra Prize in Distributed Computing jointly to two papers:

► Michael Ben-Or, "Another Advantage of Free Choice: Completely Asynchronous Agreement Protocols," in *Proceedings of the Second ACM Symposium on Principles of Distributed Computing*, pages 27-30, August 1983. <http://dl.acm.org/citation.cfm?id=806707>

► Michael O. Rabin, "Randomized Byzantine Generals," in *Proceedings of Twenty-Fourth IEEE Annual Symposium on Foundations of Computer Science*, pages 403-409, November 1983. <http://bit.ly/1HwxtDh>

In these papers published in close succession in 1983, Ben-Or and Rabin started the field of fault-tolerant randomized distributed algorithms, according to the prize committee.

Ben-Or and Rabin were the first to use randomness to solve a problem, consensus in an asynchronous distributed system subject to failures, which had probably no deterministic solution. In other words, they were addressing a computability question and not a complexity one, and the answer was far from obvious.

Ben-Or and Rabin's algorithms opened the way to a large body of work on randomized distributed algorithms in asynchronous systems, not only on consensus, but also on both theoretical problems, such as renaming, leader election, and snapshots, as well as applied topics, such as dynamic load balancing, work distribution, contention reduction, and coordination in concurrent data structures.

The Edsger W. Dijkstra Prize in Distributed Computing is given for outstanding papers on the principles of distributed computing, whose significance and impact on the theory and/or practice of distributed computing has been evident for at least a decade. The prize includes an award of \$2,000, sponsored jointly by the ACM Symposium on Principles of Distributed Computing (PODC) and the EATCS Symposium on Distributed Computing (DISC).

positioning against the correct pose geometry, and provides verbal instructions and auditory feedback to guide the person into the proper position.

Rector chose Kinect because of its open source software, as well as the widespread availability of Kinect hardware. She acknowledges the biggest challenge was “documenting [the set-up process] well enough so someone with a screen reader can download and install the software and [set up] the Kinect’s cameras without assistance.”

Meanwhile, Eelke Folmer, an associate professor of computer science and the head of the University of Nevada Reno’s Human Plus Lab, worked with Tony Morelli of Central Michigan University, John Foley of the State University of New York (SUNY) Cortland, and Lauren Lieberman of SUNY Brockport to develop a project called VI Fit, which creates modified, personal computer versions of popular Nintendo Wii games. The first title, VI Tennis, uses a modified Wii remote control to provide haptic feedback, along with audio and speech effects, allowing blind players to “see” the ball and play a version of the game. Folmer has since published adaptations of the Wii Bowling game, as well as Pet-n-Punch, a game inspired by the Whack-a-Mole game.

“A lot of those kids don’t participate in regular physical activities because it’s not safe,” Folmer says, referencing a study conducted by his collaborator Lauren Liebermann, who found parents of the visually impaired often are concerned about the risk of falling or other hazards that come from exercising in an outdoor, uncontrolled environment. “I looked at these exercise games, and I thought they were pretty fun, you can do them independently, and they are safe to play,” Folmer says.

Another issue impacting the availability of assistive technology is a lack of a centralized push for accessible solutions from the disabled community. Because the needs and challenges of blind people are distinct from the needs of those with other impairments, such as hearing loss, muscular control issues, or other disabilities (such as dyslexia), there is no centralized advocate for increased accessibility.

Clearly, those with disabilities have backing from government and industry organizations. The U.S. Department of Labor Office of Disability Employment

Because the needs of the blind are distinct from the needs of those with other impairments, there is no centralized advocate for increased accessibility.

Policy (ODEP) serves as an advocate for those with disabilities, and the Assistive Technology Industry Association (ATIA) is an association of manufacturers supportive of the development of assistive technologies. However, because the needs and challenges of blind people are distinct from the needs of those with other impairments, such as hearing loss, muscular control issues, or other disabilities (such as dyslexia), there is no single advocate from the disabled community itself to push for greater innovation.

Nonetheless, another group operating out of the University of Washington is trying to address disabilities from a holistic perspective. The DO-IT (Disabilities, Opportunities, Internetworking, and Technology) Center is a non-profit organization dedicated to empowering people with disabilities through technology and education. Working with school-age children and college students, DO-IT seeks grants and funding to promote awareness and accessibility; since its inception in the early 1990s, it has received grants totaling more than \$55 million.

The Center’s largest program, Access Computing, provides funds to increase the participation of students with disabilities in the computing field. Led by Sheryl Burgstahler, founder of the DO-IT Center, and Richard Ladner, a professor of computer science and engineering at the University of Washington, the program is designed to help disabled students get more involved in the computing field, which may lead to better integration of accessibility features in the applications and technologies of the future.

“There’s a need for leaders in the

disability community,” explains Burgstahler. “Oftentimes, leaders might know a lot about their own community, like blindness, but those people don’t tend to know a lot about learning disabilities or Asperger’s. Our programs are all about leadership, and so we expect students to learn about different disabilities, and be advocates for the whole community, not just themselves.”

Still, the relatively small market sizes for those with specific disabilities makes it difficult for mainstream technology or hardware providers to justify the development, production, or distribution of accessible technology aimed specifically at each of those communities. That is where technologies that have been successfully used in other fields can and should be examined to see how they might be used to address accessibility issues.

“Unfortunately, for companies, it’s not always marketable to have every single add-on for every single disability, because you don’t know how big your audience will be,” Rector says.

However, Pröll says that looking at existing solutions in adjacent markets, and seeing how they can be adapted for use in accessibility, may help enlarge the overall potential market size for a specific technology.

“I’m using some face-tracking technology that is being used in the animation market,” Pröll says. “Putting these technologies to use, and thinking about how people with disabilities can use it, is the approach we need to take.”

Further Reading

University of Washington Disabilities, Opportunities, Internetworking, and Technology (DO-IT) Center
<http://www.washington.edu/doit/>

Morelli, T., Liebermann, L., Foley, J., and Folmer, E.
An Exergame to Improve Balance in Children who are Blind. *Foundations of Digital Interactive Games*, April 2014
http://fdg2014.org/papers/fdg2014_wip_13.pdf

Eyes-Free Yoga: An Exergame Using Depth Cameras for Blind & Low Vision Exercise
https://youtu.be/cm_ghJPqj70

Keith Kirkpatrick is principal of 4K Research & Consulting, LLC, based in Lynbrook, NY.

© 2016 ACM 0001-0782/16/04 \$15.00

Search Engine Agendas

Is Google trying to trick you on the way to the polls?

IN THE NOVEL *1984*, George Orwell imagines a society in which powerful but hidden forces subtly shape peoples' perceptions of the truth. By changing words, the emphases put on them, and their presentation, the state is able to alter citizens' beliefs and behaviors in ways of which they are unaware.

Now imagine today's Internet search engines did just that kind of thing—that subtle biases in search engine results, introduced deliberately or accidentally, could tip elections unfairly toward one candidate or another, all without the knowledge of voters.

That may seem an unlikely scenario, but recent research suggests it is quite possible. Robert Epstein and Ronald E. Robertson, researchers at the American Institute for Behavioral Research and Technology, conducted experiments that showed the sequence of results from politically oriented search queries can affect how users vote, especially among undecided voters, and biased rankings of search results usually go undetected by users. The outcomes of close elections could result from the deliberate tweaking of search algorithms by search engine companies, and such manipulation would be extremely difficult to detect, the experiments suggest.

Writing in *Proceedings of the Na-*

“Unregulated election-related search engine rankings could pose a significant threat to the democratic system of government.”

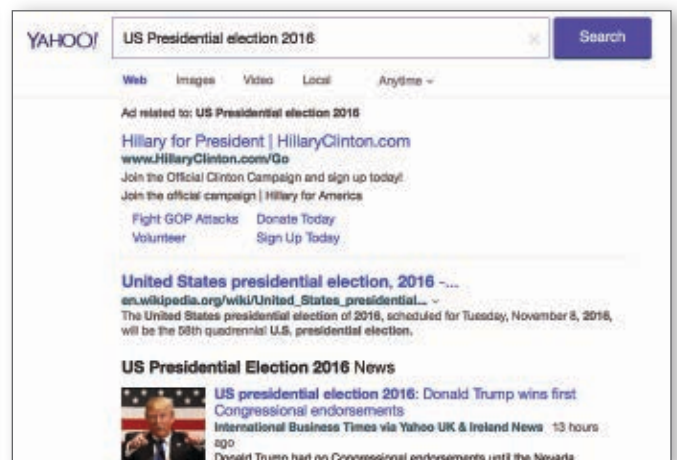
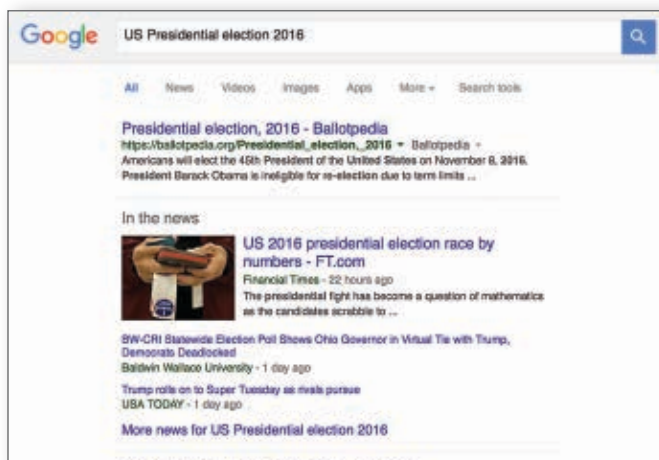
tional Academy of Sciences, Epstein and Robertson conclude, “Given that search engine companies are currently unregulated, our results ... [suggest] that such companies could affect—and perhaps are already affecting—the outcomes of close elections worldwide ... Unregulated election-related search engine rankings could pose a significant threat to the democratic system of government.” Epstein says his concerns center on Google, because of its dominant position, with two-thirds of the search engine market in the U.S. and 90% in Europe.

A spokeswoman for Google derided the notion the company might attempt to influence elections by calling it a

“conspiracy theory.” She cited a statement by Google senior vice president of Search Amit Singhal that “Google has never ever re-ranked search results on any topic (including elections) to manipulate user sentiment. Moreover, we do not make any ranking tweaks that are specific to elections or political candidates. From the beginning, our approach to search has been to provide the most relevant answers and results to our users, and it would undermine people’s trust in our results, and our company, if we were to change course.”

The Experiments

Epstein and Robertson conducted five double-blind experiments to determine if biased search engine rankings might actually sway elections. In each of the first three experiments, 102 people recruited from the public in San Diego were given brief biographies of both candidates in the 2010 Australian election for prime minister and then were asked to state their preferences based on the biographies. Then the subjects were given alternate search engine results—with links to real websites, which they were encouraged to explore—bearing on the election. The rankings of some of the results put one candidate near the top of the search results, while some ranked the other candidate higher and some



Research has shown the order in which the results of search engine queries are presented can affect how users vote.

were balanced between the two. The subjects, who were unfamiliar with the Australian election, were then asked how they would vote based on all the information at hand. A statistical analysis showed the subjects came to view more favorably the candidates whose search results ranked higher on the page, and were more likely to vote for them as a result.

In another experiment, Epstein and Robertson selected 2,150 demographically diverse subjects during the 2014 Lok Sabha elections in India, in which 430 million votes were cast. They found voters were similarly subject to unconscious manipulation by search engine results. In particular, the larger sample size revealed subjects who had reported a low familiarity with subjects were more likely to be influenced by manipulation of search engine results, suggesting manipulation attempts might be directed at these voters.

Depending on how the experiments were structured, between zero and 25% of the subjects said afterward they had detected bias in the search engine rankings. However, in a counterintuitive result, those subjects who reported seeing bias were nevertheless more likely to be influenced by the manipulation; they apparently felt there must be a good reason for the

bias, and so it tended to validate their choice of candidates.

The researchers found search engine rankings could shift voter preferences by 20% to 80% depending on demographics such as party affiliation and income level, suggesting manipulation could be targeted at certain groups. “This is incredibly important from a practical perspective, especially when we’re talking about companies that maintain massive profiles about people,” Epstein says.

Epstein says the shift comes from the widespread belief that search engine results that rank high are somehow better or more “correct” than lower-ranking items. This view is constantly reinforced as people run queries—such as, “What is the capital of Uganda?”—for which the correct answer invariably appears at the top of the results page.

What to Do?

Epstein admits there is no evidence that any search engine company has ever tried to manipulate election-related search rankings, but he says the results of his experiments are cause for concern because they show how easily that could be done, either at the direction of the management of a search company, or by a “rogue employee” with hands-on access to complex

search algorithms. Even absent deliberate manipulation, search engine rankings can become self-reinforcing through the “digital bandwagon effect,” in which users see top-ranked candidates as somehow better and more worthy of their respect.

One solution to the problem could be an “equal-time rule” (in the U.S., the equal-time rule mandates radio and television broadcast stations airing content by a political candidate must provide an equivalent opportunity to any opposing political candidates who request it) that requires search companies to mix the results of searches about election-related matters so no candidate has any rank advantage, Epstein says. “Either search engine companies are going to have to do this voluntarily, or they will see standards set by an industry association or a non-profit or by government,” he says, “because if we don’t start moving in that direction, the free and fair election will, for all intents and purposes, be meaningless.”

Another possibility might be to post warnings at the top of political search results—similar to those that now flag advertisements—telling users the order in which results are shown may reflect bias in favor of the candidate(s) ranked near the top. Epstein acknowledges search companies are unlikely to

Education

A New Framework to Define K-12 Computer Science Education

For most states and school districts, the notion of computer science for *every* student is a relatively new and unexplored topic. Responding to parent demand for their children to have access to computer science, there has been a major shift in thinking by states and school districts about how to make computer science part of core academic work. They are asking big questions of the computing community: What is the appropriate scope and sequence for K-12 computer science? What does the community expect every student to learn in elementary school, in middle school, or by the time they graduate high school? And why?

CSTA, ACM, and Code.org are joining forces with more than 100 advisors within the computing community (higher ed faculty, researchers, and K-12 teachers, many of whom are also serving as writers for the framework), several states and large school districts, technology companies, and other organizations to steer a process to build a framework to help answer these questions. A steering committee initially comprised of the CSTA, ACM, and Code.org will oversee this project.

The framework will identify key K-12 computer science concepts and practices we expect students exiting grades 2,

5, 8, and 12 to know. This effort will not develop educational standards. We expect that states and school districts will use the framework to create their own frameworks, guidance, and standards, and the CSTA has its own independent process for developing detailed K-12 computer science standards (<http://csta.acm.org/Curriculum/sub/K12Standards.html>).

Underpinning this effort is our belief that computer science provides foundational learning benefiting every child. Computer science gives students a set of essential knowledge and skills important for students’ learning and for their future careers and interests. This work

is about defining the *basic* expectations for what *every* student should have a chance to learn about K-12 computer science to prepare for the emerging demands of the 21st century—not just to major in computer science or secure jobs as software engineers.

The projected release date for the framework is summer 2016. More information, including monthly updates and how to get involved, can be found at K12CS.org.

— Mark Nelson is Executive Director of CSTA

— Mehran Sahami is Chair of the ACM Education Board

— Cameron Wilson is Chief Operating Officer of Code.org

There is no evidence that any search engine company has ever tried to manipulate election-related search rankings.

voluntarily adopt an equal-time rule or the warnings, but he says either or both could be built into the browser, acting automatically when search results contain the names of political candidates.

The idea government might play a role in regulating search engines is not new, and it is strongly opposed by search companies and by many First Amendment watchdogs. Frank Pasquale, now a professor of law at the University of Maryland, in 2008 wrote a paper recommending the establishment of a Federal Search Commission. He argues free speech concerns do not apply to search because search engines act more like common carriers, which are subject to regulation, than like media outlets, which enjoy First Amendment protections.

As for why we need federal regulation, Pasquale says, “When a search engine specifically decides to intervene, for whatever reason, to enhance or reduce the visibility of a specific website or a group of websites ... [it] imposes its own preferences or the preferences of those who are powerful enough to induce it to act.”

Beyond Elections

Concerns about algorithms that search, select, and present information extend beyond search companies. “I’ve looked at the black box algorithms behind Google, Facebook, Twitter, and the others,” Pasquale says, “and I’m pretty troubled by the fact that it’s so hard to understand the agenda that might be behind them.” He supports the idea of a “trusted advisory committee” of technical, legal, and business experts to advise the Federal Trade Commission on the fairness

of the algorithms of those companies.

Not only are the algorithms behind these major services complex and secret, often users do not know that any selection logic or personalization occurs at all, says Karrie Karahalios, a professor of computer science at the University of Illinois and co-director of the Center for People and Infrastructures. In a study involving 40 of her students, more than half were “surprised and angered” to learn there was a “curation algorithm” behind the Facebook News Feed. She says such “invisible algorithms,” in the interest of efficiency, can mislead people by acting as secret “gatekeepers” to information.

Karahalios recommends browsers offer graphical cues to users to show how the algorithms work, so users know why they are seeing certain results. For example, when an item ranks high in search results because it has many links to other things, she suggests that might be signaled with a larger type font. She also says users should have some control over how the algorithms work. “I think it is important to have some levers that users can poke and prod to see changes in the algorithmic system,” she says.

In 2014, Karahalios and several colleagues presented five ideas by which algorithms, even secret ones, might be “audited” for bias by outside parties. In one, the Sock Puppet Audit, computer programs would impersonate actual users, generating test data and analyzing the results. Similarly, the testing and evaluation of algorithms could be crowd-sourced, by some mechanism such as Amazon’s Mechanical Turk.

The advocates of audits agree these ideas present technical and legal difficulties, but they say some kind of external checking on the fairness of these ubiquitous services is needed. “Putting warnings on search results is not enough,” Karahalios says.

Luciano Floridi, a professor of philosophy and ethics of information at the University of Oxford, says the power and secrecy of Google is worrisome in part because of the company’s near-monopoly power. “Nothing wrong has happened so far, but that’s not a strategy,” he says; “that’s like keeping your fingers crossed.” He says recent revelations that Volkswagen AG manipulated engine software to fool regulators and

consumers are not reassuring.

Floridi says the risk of mischief is compounded because Google’s users are not customers in the retail commercial sense. “They are not accountable because users are not paying for searches,” he says. “We don’t have customers’ rights with Google.”

Floridi advises Google on “the right to be forgotten” regulations by the European Union. He says he finds his contacts at the company to be “open-minded and sensible” about ideas for regulating search. “If it makes good sense socially speaking, and if it makes good sense business-wise, then there is a conversation on the table,” he says. **■**

Further Reading

Bracha, O., Pasquale, F.,
Federal Search Commission? Access, fairness, and accountability in the law of search. *Cornell Law Review*, September 2008
http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1002453

Epstein, R.,
The search engine manipulation effect and its possible impact on the outcomes of elections. *Proceedings of the National Academy of Sciences*, Aug. 18, 2015
<http://www.pnas.org/content/112/33/E4512.abstract>

Pasquale, F.,
The black box society: the secret algorithms that control money and information. *Harvard University Press*, 2015
<http://www.hup.harvard.edu/catalog.php?isbn=9780674368279>

Sandvig, C., Hamilton, K., Karahalios, K., and Langbort, C.,
Auditing algorithms: research methods for detecting discrimination on Internet platforms. 64th Annual Meeting of the International Communication Association, May 22, 2014
http://acawiki.org/Auditing_Algorithms:_Research_Methods_for_Detecting_Discrimination_on_Internet_Platforms

Zittrain, J.,
Engineering an election – digital gerrymandering poses a threat to democracy. *Harvard Law Review Forum*, Jun 20, 2014
<http://harvardlawreview.org/2014/06/engineering-an-election/>

Videos – How Google Works
<https://www.youtube.com/watch?v=Md7K90FfJhg>
<https://www.youtube.com/watch?v=3tNpYpcU5s4>

Gary Anthes is a technology writer and editor based in Arlington, VA.

© 2016 ACM 0001-0782/16/04 \$15.00

Marvin Minsky: 1927–2016

MARVIN MINSKY, AN American scientist working in the field of artificial intelligence (AI) who co-founded the Massachusetts Institute of Technology (MIT) AI laboratory, wrote several books on AI and philosophy, and was honored with the ACM A.M. Turing Award, passed away on Sunday, Jan. 24, 2016 at the age of 88.

Born in New York City, Minsky attended the Ethical Culture Fieldston School, the Bronx High School of Science, and Phillips Academy, before entering the U.S. Navy in 1944. After leaving the service, he attended Harvard University, where he earned a bachelor's degree in mathematics in 1950. He then went to Princeton University, where he built the first randomly wired neural network learning machine, the Stochastic Neural Analog Reinforcement Calculator (SNARC), before earning his Ph.D in mathematics there in 1954.

Doctorate in hand, Minsky was admitted to the group of Junior Fellows at Harvard, where he invented the confocal scanning microscope for thick, light-scattering specimens, decades in advance of the lasers and computer power needed to make it useful; today, it is in wide use in the biological sciences.

He began teaching at MIT in 1958; the following year, he joined John McCarthy in founding the MIT Artificial Intelligence Laboratory (today known as the Computer Science and Artificial Intelligence Laboratory, or CSAIL).

At the time of his death, he was the Toshiba Professor of Media Arts and Sciences, and professor of electrical engineering and computer science, at CSAIL.

Beginning in the early 1950s, Minsky worked on computational ideas to characterize human psychological processes, and produced theories on how to endow machines with artificial intelligence. Work in the new laboratory included attempts to model hu-

Minsky worked on computational ideas to characterize human psychological processes, and produced theories on how to endow machines with artificial intelligence.

man perception and intelligence, as well as efforts to design and build practical robots.

Minsky had argued that space exploration, undersea mining, and nuclear safety would be vastly simpler with manipulators driven locally by intelligent computers or remotely by human operators. He foresaw that microsurgery could be done by surgeons who work at one end of a telepresence system at a comfortably large scale while at the other end machines do the chores required at the small scale where tiny nerve bundles are knitted together or clogged blood vessels are reamed out. In support of this, Minsky designed and built mechanical hands with tactile sensors, and an arm with 14 degrees of freedom.

In the late 1960s, Minsky began to work on perceptrons, simple computational devices that capture some of the characteristics of neural behavior. Minsky and Seymour Papert showed what perceptrons could and could not do. Together they wrote the book *Perceptrons*, which is considered a foundational work in the analysis of artificial neural networks.

Minsky and Papert continued their collaboration for decades, bringing together Minsky's computational

ideas with Papert's understanding of developmental psychology. They worked both together and individually to develop theories of intelligence and radical new approaches to childhood education using Logo, the educational programming language developed by Papert and his colleagues. Together, they developed the first Logo "turtle" robot.

Minsky's best-known work from the mid-1970s centers on a family of ideas he called the Theory of Frames. In his paper "A Framework for Representing Knowledge" (<http://bit.ly/1PezuKf>), Minsky wrote, "the ingredients of most theories both in Artificial Intelligence and in Psychology have been on the whole too minute, local, and unstructured to account—either practically or phenomenologically—for the effectiveness of common-sense thought." He tried to address those issues by considering several theories of intelligence, then "pretending to have a unified, coherent theory" based on his proposal to label data structures in memory as frames and considering how frames must work, individually and in groups. Frames have become the primary data structure of AI Frame Languages, and are a major part of knowledge representation and reasoning schemes.

Minsky and Papert also developed what came to be called The Society of Mind theory, which attempts to explain how intelligence could be a product of the interaction of non-intelligent parts. Minsky said his greatest source of ideas about the theory came from his work in trying to create a machine that uses a robotic arm, a video camera, and a computer to build with children's blocks. In 1986, Minsky published *The Society of Mind* (amzn.to/1NOJ0lu), a book on the theory written for a general audience.

Minsky also wrote about the potential for communication with extraterrestrials ("Communication with Alien Intelligence," bit.ly/1NOJ7xl), offering arguments to support the notion that





Marvin Minsky at his home in Brookline, MA.

we will be able to converse with aliens on our first meeting “because we’ll both think in similar ways.”

In 2006, Minsky published *The Emotion Machine* (<http://bit.ly/1QFEfPy>), a book critiquing theories of how human minds work and suggesting alternative theories, often replacing simple ideas with more complex ones. He wrote that our resourceful intelligence arises from many ways of thinking (search, analogy, divide and conquer, elevation, reformulation, contradiction, simulation, logical reasoning, and impersonation) that are spread across many levels of mental activity (instinctive reactions, learned reactions, deliberative thinking, reflective thinking, self-reflective thinking, and self-conscious emotions).

Minsky was awarded the fourth-ever ACM A.M. Turing Award (known as the “Nobel Prize of Computing”) in 1969, “for his central role in creating, shaping, promoting, and advancing the field of Artificial Intelligence.” He also received the Japan Prize in 1990, the International Joint Conference on Artificial Intelligence (IJCAI) Award for Research Excellence in 1991, and the Benjamin Franklin Medal from the Franklin Institute in 2001. In 2006, he was inducted as a Fellow of the Computer History Museum, and in 2011, Minsky was inducted into IEEE Intelligent Systems’ AI’s Hall of Fame for the “significant contributions to the

field of AI and intelligent systems.” In 2014, Minsky was presented with the Dan David Prize in the field of “Artificial Intelligence, the Digital Mind.” He was also awarded the 2013 BBVA Foundation Frontiers of Knowledge Award in the Information and Communication Technologies category.

During his tenure, Minsky served as doctoral advisor to, among others:

- Manuel Blum, recipient of the ACM A.M. Turing Award in 1995 “In recognition of his contributions to the foundations of computational complexity theory and its application to cryptography and program checking.”

- Daniel Bobrow, developer of the TENEX operating system, president of the American Association for Artificial Intelligence, chair of the Cognitive Science Society, editor-in-chief of the journal *Artificial Intelligence*, and part of the team that received the 1992 ACM Software Systems Award for its work on the Interlisp programming environment.

- Danny Hills, co-founder of super-computer manufacturer Thinking Machines Corporation, Judge Widney Professor of Engineering and Medicine at the University of Southern California, and recipient of the ACM Grace Murray Hopper Award in 1989 “for his basic research on parallel algorithms and for the conception, design, implementation, and commercialization of the Connection Machine.”

- Gerald Jay Sussman, Panasonic Professor of Electrical Engineering at MIT, recipient of ACM’s Karl Karlstrom Outstanding Educator Award in 1990.

- Ivan Sutherland, recipient of the ACM A.M. Turing Award in 1988 “for his pioneering and visionary contributions to computer graphics, starting with Sketchpad, and continuing after,” and the Kyoto Prize in Advanced Technology in 2012 for “pioneering achievements in the development of computer graphics and interactive interfaces.”

Patrick Henry Winston, Ford Professor of Artificial Intelligence and Computer Science at MIT, recalled, “Many years ago, when I was a student casting about for what I wanted to do, I wandered into one of Marvin’s classes. Magic happened. I was awed and inspired. I left that class saying to myself, ‘I want to do what he does.’ I have been awed and inspired ever since. Marvin became my teacher, mentor, colleague, and friend. I will miss him at a level beyond description.”

Winston added, “Marvin’s impact was enormous. People came to MIT’s Artificial Intelligence Laboratory from everywhere to benefit from his wisdom and to enjoy his deep insights, lightning-fast analyses, and clever jokes. They all understood they were witnessing an exciting scientific revolution. They all wanted to be part of it.”

Moshe Vardi, Karen Ostrum George Distinguished Service Professor in Computational Engineering and director of the Ken Kennedy Institute for Information Technology at Rice University, as well as editor-in-chief of *Communications*, said, “Minsky was an out-of-the-box thinker, which he demonstrated already as a graduate student, when he built the most useless machine ever (<http://bit.ly/1PQ6pV1>), which did nothing but switch itself off.”

ACM president Alexander L. Wolf said Minsky “began his work at a time when computing was like a newly discovered continent, vast and unexplored. The many paths he blazed were important, not only because they were first but because they led us to a better place.”

Lawrence M. Fisher is Senior Editor/News for ACM Magazines.

© 2016 ACM 0001-0782/16/04 \$15.00

PHOTO BY L. BARRY HETHERINGTON, COURTESY OF MIT

A Decade of ACM Efforts Contribute to Computer Science for All

IN LATE JANUARY, U.S. President Barack Obama asked Congress to approve \$4.1 billion in spending in the coming fiscal year to support the Computer Science for All initiative, aimed at providing computer science education in U.S. public schools. Obama pointed out computer science is no longer “an optional skill” in the modern economy,” yet “only about a quarter of our K–12 (kindergarten through 12th grade) schools offer computer science. Twenty-two states don’t even allow it to count toward a diploma.”

While many organizations have contributed to the national effort to see real computer science exist and count toward graduation requirements in U.S. public schools, former ACM CEO John R. White said, “ACM has been there from the beginning.” Indeed, White contends Obama’s Computer Science for All initiative “in a way represents the culmination of more than a decade of effort initiated by the ACM.”

Computer science education in public schools has been a main focus for ACM since the 1990s. “This concern for, and commitment to, K–12 computer science resulted in the formation of the Computer Science Teachers Association (CSTA, <http://www.csta.acm.org/>) in the 2004 timeframe,” noted White. “Supporting the launch of CSTA moved ACM’s efforts from a series of task forces concerned with K–12 computer science education to a national effort focused on supporting and growing the community of computer science teachers.”

CSTA founding director Chris Stephenson, who now is head of computer science education programs at Google, said that even before the official formation of CSTA, its future leaders were working to raise the national consciousness regarding CS education.



U.S. President Barack Obama discussing his Computer Science for All plan to give students across the country the chance to learn computer science in school.

She said the ACM Model Curriculum, published by the ACM K–12 Task Force in 2003, was “a germinal work, making the argument that computer science was a rigorous academic discipline with a body of knowledge that could and should be reflected in computing courses in schools.”

In 2005, CSTA published its first official white paper, *The New Educational Imperative: Improving High School Computer Science Education* ([http://](http://bit.ly/1O1haT1)

bit.ly/1O1haT1), which Stephenson described as a “companion piece to what eventually became the *CSTA K–12 Computer Science Standards*” (<https://csta.acm.org/Curriculum/sub/K12Standards.html>), addressed the link between K–12 computer science education and national technological competitiveness and “provided a strong call to policy makers to begin addressing computer science education at the state and local level.”

White recalled that in 2007, these myriad CS education efforts were augmented with the creation of the ACM Education Policy Committee (EPC, <http://www.acm.org/public-policy/education-policy-committee>), an organization “focused on education policy as it related to K–12 computer science” with the goal of seeing “real computer science exist and count in U.S. high schools.”

Among the organizations joining the effort to get CS education into public schools are the National Center for

“ACM has been there from the beginning,” said former ACM CEO John R. White.

Women and Information Technology (NCWIT, <https://www.ncwit.org/>) the National Science Foundation (NSF, <http://www.nsf.gov/>), and Code.org, as well as corporations such as Microsoft and Google.

White said two major events “really helped move the K–12 computer science education effort into high gear.” One was the release of the CSTA/ACM-EPC report *Running on Empty*, which highlighted the deplorable state of computer science education in the 50 states; the other was Congressional action to create Computer Science Education Week (<https://csedweek.org/>), an annual program dedicated to inspiring K–12 students to take interest in computer science launched by the Computing in the Core Coalition and now organized by Code.org. “These events, along with NSF’s efforts to nurture the emergence of new high school level computer science courses, set the foundation for a real transformation in high school-level computer science education.”

Early Days

In 2005, Cameron Wilson joined ACM as director of the ACM Policy Office in Washington, D.C. He recalled that early on, he, ACM CEO White, and CSTA’s Stephenson wanted to evaluate the state of CS education in U.S. public schools, only to learn “computer science really isn’t represented in K–12.” In trying to pin down what was keeping CS education out of schools, they asked, “what are the policy im-

“What we’ve seen in the past three years is this impressive groundswell of interest ... to take computer science seriously or to do more to boost computer science instruction.”

plications? Why doesn’t computer science education really exist in the K–12 space? Is this a curriculum problem? Is this an image problem? Is this a policy problem? The more the community at large looked at these issues, it was definitely all of those.”

That was the impetus for the formation of the ACM Education Policy Committee (EPC), chaired by Robert (“Bobby”) Schnabel, who only left the group in November to take on the roles of ACM CEO and executive director. The goal of the committee, Wilson said, “was to unpack the policy issues around computer science education and to figure out what we could do to advance the field in K–12 education.”

It took several years to find their way, Wilson said, “in terms of figuring out just what are the policy issues, because it turns out in education that policy and implementation, which means what actually gets taught, are deeply linked so the policies that are at the state or federal or local level are all contributing to an ecosystem of what actually gets taught in schools. Part of our goal was figuring out what are the policy levers that you would need to pull to expand CS K–12 instruction.”

The EPC determined it needed to assess the state of CS education for each of the 50 U.S. states, resulting in the 2010 report *Running on Empty: The Failure to Teach K–12 Computer Science in the Digital Age*, (<http://runningonempty.acm.org/>). In that study, Wilson explained, the EPC tried to answer two policy-related questions:

- ▶ To what extent do states have education standards around computer science?
- ▶ Do computer science courses at the high school level count toward a core graduation requirement or are they simply elective?

Wilson worked with co-authors Leigh Ann DeLyser of Carnegie Mellon University (now at CSNYC.org, an organization established in 2013 to ensure all New York City public school students have access to CS education), Mark Stehlik of Carnegie Mellon, and CSTA’s Stephenson. Wilson said the report found that “states don’t have standards around computer science edu-



Middle school students at a Computer Science for All event.

cation generally, and the ones that do exist are really about basic technology literacy and using technology are not focused on allowing students to create technologies.” At the time, just nine states allowed computer science to count toward math or science requirements for high school graduation.

Around the same time, the EPC launched Computer Science Education Week as a collaborative, (computing) community-based event around computer science education. The first Computer Science Education Week took place in December 2009 as a joint effort led and funded by ACM with the cooperation and deep involvement of CSTA, NCWIT, NSF, the Anita Borg Institute, the Computing Research Association, Google, Intel, and Microsoft.

Today, the annual Computer Science Education Week is supported by 350 partners and 100,000 educators worldwide, and includes the Hour of Code—a one-hour introduction to computer science designed to demystify code and show that anybody can learn the basics. During 2014’s Computer Science Education Week, Obama became the first U.S. president to write a line of code as part of the Hour of Code.

“Computer Science Education Week came first, and then *Running on Empty* came out, and we bootstrapped both of those things into a new coalition of industry and non-profits called Computing in the Core,” said Wilson. “The main goal of Computing in the Core was to help be a steward for Computer Science Education Week, and to help advocate for policies at the state and federal level. At the time, we were just focused on federal policy because we were pretty small, with a shoestring budget, and we just didn’t have the resources to work at the state level.”

2013 saw the launch of Code.org, a “non-profit dedicated to expanding access to computer science, and increasing participation by women and underrepresented students of color. Our vision is that every student in every school should have the opportunity to learn computer science. We believe computer science should be part of core curriculum, alongside other courses such as biology, chemistry, or algebra.”

Recalled Stephenson, “Through its participation in the ACM Educa-

Computer Science for All

U.S. President Barack Obama’s proposal for the Computer Science for All initiative includes:

- \$4 billion in funding for states in the federal budget for FY2017 (beginning Oct. 1 this year), as well as \$100 million in funding directly for school districts that provide students greater access to computer science (CS) education. The president said the goal of this funding is to have “hands-on” computer science courses in every public high school, as well as expanding CS learning opportunities for students in elementary (grades K through 6) and middle schools (typically grades 7 through 9).
- The National Science Foundation and the Corporation for National and Community Service (<http://1.usa.gov/1QizE3h>), “a federal agency that helps more than 5 million Americans improve the lives of their fellow citizens through service,” will invest more than \$135 million over the next five years in training teachers to teach computer science.
- Expanding access to prior NSF-supported programs and professional learning communities through the CS10k Initiative (<https://cs10kcommunity.org/>) that led to the creation of more inclusive and accessible CS curriculum.
- Obama asked governors, mayors, CEOs, philanthropists, creative media, technology professionals, and education leaders/professionals “to deepen their CS commitments” and for governments, education leaders, business leaders, and others to get involved.

The White House said 30 school districts, as well as the statewide district of Hawaii, already had committed to expanding opportunities for CS education in their schools, while Delaware has announced it will launch an online CS education class for its students.

—L.M.F.

tion Policy Committee, CSTA leaders helped establish Computer Science Education Week and worked as part of the Computing in the Core Coalition to build a powerful network of education, association, and industry representatives committed to improving computer science education nationally. Policy events in D.C. helped focus attention on education and workforce problems and connect computer science education to the national conversation about jobs. This coalition, including the National Science Foundation, also was instrumental in managing the first two CS Education Weeks, supporting the CS 10K project, and launching Code.org.”

Eventually, it made sense to merge Computing in the Core and Code.org, and ACM loaned Wilson to Code.org for a year to help get the new organization off the ground. Since then, said Wilson, now in the permanent roles of COO and vice president of Government Affairs for Code.org, “What we’ve seen in the past three years is this tremendous groundswell of interest from teachers, from parents, from students, and then from hundreds of school districts and dozens of states, to take computer science seriously or to do more to boost computer science instruction.

“All of the things we did prior to that helped contribute to that overall groundswell. That’s what the

President has really tapped into; this is now clearly a national movement that’s being state-led, and he helped contribute a bully pulpit to it, and also has proposed a substantial amount of funding around this, which Congress will ultimately have to figure out whether they’re going to appropriate or not.”

In a joint statement following Obama’s announcement, ACM CEO Schnabel and ACM president Alexander L. Wolf observed that ACM “has played a major, seminal role in raising the visibility of computer science education and the need for more attention to it in schools.”

The association, they said, “is dedicated to continuing to support the progress of computing education worldwide through its close relationship with and support of CSTA, its world-leading development of computing curricula, and its conferences and publications on computer science education. It looks forward to building on the increased momentum created by [the president’s] announcement to partner with all groups that are dedicated to increasing the quality and availability of computing education worldwide.”

Lawrence M. Fisher is Senior Editor/News for Communications.

© 2016 ACM 0001-0782/16/04 \$15.00

Global Computing

The Internet and Inequality

Is universal access to the Internet a realistic method for addressing worldwide socioeconomic inequality?

FACEBOOK CEO MARK ZUCKERBERG echoed the hopes of many when he launched Internet.org to bring more of the world's population online.² He said: "The richest 500 million have way more money than the next six billion combined. You solve that by getting everyone online, and into the knowledge economy—by building out the global Internet."⁸

But, does giving everyone the Internet really reduce inequality?

Zuckerberg might have been thinking of the economics of diminishing returns, in which technology has a kind of saturation effect—the more of a particular technology you have, the less incremental economic value it contributes. Some economists argue, therefore, that giving developing-world nations a new technology facilitates fast growth, allowing them to catch up with developed countries whose own growth with respect to that technology slows down.¹

This may in fact be happening. World Bank economists Christoph Lakner and Branko Milanovic estimate

In and of themselves, digital platforms for those with little social and educational capital are meaningless.

global inequality has declined over the last 20 years,⁷ a period during which the number of mobile phone accounts worldwide grew so much it almost exceeded the human population.³ Maybe the rapid spread of digital tools is shrinking disparities.

More concrete evidence, however, suggests universal connectivity is not the solution to inequality. If anything, digital technologies might aggravate inequalities. Within the U.S., for example, massive open online courses are overwhelmingly completed by col-

lege-educated professionals, not jobless high school dropouts.⁹ The actor/director Zach Braff raised over \$3 million through crowdfunding, while the average Kickstarter campaign raises only \$6,000.^{4,6} And over the last four decades, digital technologies penetrated every corner of American life—Google and Facebook became household words, and even homeless people enjoy connectivity at public libraries. Yet, the rate of poverty stagnated, social mobility decreased, and inequality skyrocketed.

There are two reasons why technology dissemination in and of itself does not address inequality. First, the rich can always afford more and better technology. Wealthier people benefit more from innovation because they can afford fancy gadgets and premium services. There is no digital keeping up with the Joneses. Second, eliminating the *digital* divide does not eliminate socioeconomic divides. Those with better education, more wealth, and greater influence can accomplish more with the same technology. Bill Clinton or Bill Gates could accomplish

more than you or I with a week's unlimited use of the Internet. Similarly, most *Communications* readers can do more with connectivity than someone in rural Uganda who has not completed primary school.

This is what I call technology's "Law of Amplification," and it is exactly what MOOC completion statistics and Braff's use of Kickstarter bear out. Technology is a tool; it amplifies existing human capacities. This means that if anything, indiscriminate dissemination of digital technology tends to aggravate inequalities. Technology helps only when there is firm intention—economically, politically, culturally—to push against the gradient of inequality.

"Not so fast!" you might say. We cannot know what the U.S. would have been like without Silicon Valley, so we cannot know digital technology's actual impact with certainty. That is true, but it does not change how we should respond. In any country where there is rising inequality, the possibilities for technology's role must be one of the following:

- ▶ Technology is making inequality worse.
- ▶ Technology has little or no effect on inequality, but other forces are increasing inequality.
- ▶ Technology is actually alleviating inequality, but other inequality-causing forces are so powerful as to overpower it.

The first two options imply technology cannot solve inequality by itself. The third option might suggest doubling down on technology. But consider again that in the U.S., talented, well-funded entrepreneurs have been working as hard and as fast as they can to churn out new products in a culture that supports them. If tech innovation at full speed is not enough to counter bad socioeconomic forces, maybe those forces need to be addressed directly.

Incidentally, what about the convergence among countries alluded to earlier, the one that causes commentators like *New York Times* columnist Thomas Friedman to argue "the world is flat"?⁵ Individuals in developing countries that have a good education, a hefty inheritance, or strong political ties are able to use technology to their advantage and catch up with their de-



Digital Green's video production technique as demonstrated in Jharkhand, India, in 2013. Digital Green produces participatory localized videos for farmers in India, Ethiopia, Ghana, and Afghanistan; <http://www.digitalgreen.org/>.

veloped-world peers. Most of the decline in global inequality is generally attributed to the growth of China and India. But while technology has played a role in their rise as nations, its value is unevenly distributed among their citizens. Digital tools have helped the Indian elite become an IT superpower, but cellphones for India's undereducated have done little to enrich them. In a country of 1.25 billion people and 900 million mobile accounts, three-quarters of the country still struggles on less than USD\$2.50 a day.¹⁰ In and of themselves, digital platforms for

those with little social and educational capital are meaningless.

I saw this firsthand when I moved to Bangalore in 2004 to start a new research group at Microsoft. I explored how digital tools could support education, agriculture, and healthcare to alleviate poverty. However, projects that worked well as research pilots failed when we tried to scale them up. The problem was that in our research, we could control the social context. But in scaling up, if implementing institutions were ineffective, or if potential beneficiaries lacked basic skills, new gadgetry just did not help.



Three boys at a computer literacy training center near Jhansi, India, in 2005.



A computer literacy class for girls in a low-income community initiated by the author in Bangalore, India, in 2004.

And yet, it was exactly where people were less skilled that we most wanted to provide support. For technology to help the world's least privileged, it requires exactly the human foundation that is missing to begin with.

So, what is to be done? And more specifically, how can computer scientists concerned about inequality contribute? The Law of Amplification points the way.

First, those already doing something to counter inequality should use technology to amplify their efforts. Indiscriminate dissemination of technology is futile, but targeted use to amplify progressive forces can be effective. For instance, social ac-

tivists fighting for progressive policy should use whatever communication channels they have available to get their message out. (They should be braced for an ongoing effort, however, because it is their voice, not the technology, which is the primary cause of change.)

Second, those with the proverbial hammer in search of nails should work with good carpenters. The impact of individuals or organizations that are combatting inequality successfully can often be amplified with good technology. Of course, where there aren't good carpenters, the tools won't work themselves.

Third, socioeconomic disparities

can be addressed directly by helping "have-nots" with education, mentorship, funding, and introductions to influential people. Technologists can propagate their own skills and networks, for example, by teaching programming or connecting budding entrepreneurs with seasoned ones. Conversely, giving someone the latest gadget does little in and of itself to help that person close any disparities. A metronome is a useful tool, but it does not make the concert musician.

Finally, all of us can engage the political process as citizens. Technologists often look down on politics, but inequality is a political issue. The computing industry has immense influence, and we can engage in ways beyond the skills of our professional training—just as some already do with political issues that matter to them, such as immigration.

Because technology amplifies human forces, if we ensure social currents are appropriately directed, then all of our amazing technology will work in humanity's favor. **G**

References

1. Abramovitz, M. Catching up, forging ahead, and falling behind. *The Journal of Economic History* 46, 2 (Feb. 1986), 385–406; <http://www.jstor.org/stable/2122171>
2. Best, M.L. The Internet that Facebook built. *Commun. ACM* 57, 12 (Dec. 2014), 21–23; <http://cacm.acm.org/magazines/2014/12/180792-the-internet-that-facebook-built/fulltext>
3. Ericsson. Ericsson Mobility Report: On the Pulse of the Networked Society (2014); <http://www.ericsson.com/res/docs/2014/ericsson-mobility-report-november-2014.pdf>
4. Fernandez, M.E. Zach Braff raises money—and ire—with Kickstarter campaign for new film. *NBC News* (May 22, 2013); <http://www.nbcnews.com/pop-culture/pop-culture-news/zach-braff-raises-money-ire-kickstarter-campaign-new-film-f6C10026213>
5. Friedman, T.L. *The World Is Flat [updated and expanded]: A Brief History of the Twenty-First Century*. Macmillan, 2006.
6. Heyman, S. Keeping up with Kickstarter. *New York Times* (Jan. 15, 2015); <http://www.nytimes.com/2015/01/15/arts/international/keeping-up-with-kickstarter.html>
7. Lakner, C. and Milanovic, B. Global income distribution: From the fall of the Berlin Wall to the Great Recession. World Bank Policy Research Working Paper 6719 (2013); http://www-wds.worldbank.org/serlet/WDSContentServer/WDSP/IB/2013/12/11/000158349_20131211100152/Rendered/PDF/WPS6719.pdf
8. Levy, S. Zuckerberg explains Facebook's plan to get the entire planet online. *Wired* (Sept. 25, 2013); <http://www.wired.com/2013/08/mark-zuckerberg-internet-org/>
9. Selinger, J.J. *MOOC U: Who Is Getting the Most Out of Online Education and Why*. Simon and Schuster, 2014.
10. Telecom Regulatory Authority of India. Information Note to the Press. Press Release No. 25/2014. New Delhi, (May 12, 2014); <http://traai.gov.in/WriteReadData/WhatsNew/Documents/Press%20Release-TSD-Mar14.pdf>

Kentaro Toyama (toyama@umich.edu) is W.K. Kellogg Associate Professor at the University of Michigan School of Information and author of *Geek Heresy: Rescuing Social Change from the Cult of Technology*.

Copyright held by author.



Kode Vicious

GNL Is Not Linux

What's in a name?

Dear KV,

I keep seeing the terms “Linux” and “GNU/Linux” online when I am reading about open source software. The terms seem to be mixed up or confused a lot and generate a lot of angry mail and forum threads. When I use a Linux distro am I using Linux or GNU? Does it matter?

What's in a Name?

Dear Name,

What, indeed, is in a name? As you have already seen, this quasi-technical topic continues to cause a bit of heat in the software community, particularly in the open source world. You can find the narrative from the GNU side by utilizing the link provided in the postscript appearing at the end of this column, but KV finds that narrative lacking, and so, against my better judgment about pigs and dancing, I will weigh in with a few comments.

If you want the real back story on the GNU folks and FSF (Free Software Foundation), let me suggest you read Steven Levy's *Hackers: Heroes of the Computer Revolution*, which is still my favorite book about that period in the history of computing, covering the rise of the minicomputer in the 1960s through the rise of the early microcomputers in the 1970s and early 1980s. Before we get to the modern day and answer your question, however, we have to step back in time to the late 1960s, and the advent of the minicomputer.

Once upon a time, as all good stories start, nearly all computer software

cost some amount of money and was licensed to various entities for use. That time was the 1950s and 1960s, when, in reality, very few individuals could afford a computer, and, in fact, the idea that anyone would want one was scoffed at by the companies who made them, to their later detriment. Software was developed either by the hardware manufacturer to make its very expensive machines even moderately usable, or by the government, often in collaboration with universities.

About the time of the advent of the minicomputer—which came along about the same time KV was born, screaming, because he knew he would have to fix them and their brethren someday—two key innovations occurred. Ken Thompson and Dennis Ritchie invented Unix, a now well-known reaction to the development of Multics, and computer hardware took one of its first steps toward affordability. No longer would the whole university have to share and time-slice a single large mainframe; now each department, if that department had \$30,000 or so, could share a less powerful machine, but among a much smaller group of people.

Before Unix, operating systems were large, complicated, and mostly non-portable. Because Unix was simple and written in a new, portable assembler—which we now call C—it was possible for much smaller groups of people to write significant software with a lot less effort. There were good tools for writing portable operating systems and systems software, including a compiler,



linker, assembler, and debugger—tools we now take for granted. All of these advances were significant and important, but they had one thing holding them back from even broader acceptance: licensing.

The Unix system and its tools were written and owned by AT&T, which, at that time, was the largest monopoly in the U.S. It did not have to license anything, of course, because it was *the* phone company, and *the only* phone company, which put it in a unique position, best summed up by Lily Tomlin in an old comedy sketch: “We don’t care, we don’t have to.”

The truth was many people inside AT&T did care and they were able to get AT&T to sell “cheap” licenses for institutions such as universities that had to pay only \$1,000 to get the entire source. Companies had to pay quite a bit more, but they were able to spread the cost over their entire computing infrastructure. Having the source meant you could update and modify the operating

system. If you want to see innovation in any type of software, it is very important to have the source. In 2016, more than 50 years after all these changes started, it is now common to have access to the source, because of the open source movement, but this was uncommon at the start of the Unix age.

Over the course of the Unix era, several open source operating systems came to the fore. One was BSD (Berkeley Software Distribution), built by CSRG (Computer Software Research Group) at UC Berkeley. The Berkeley group had started out as a licensee of the AT&T source, and had, early on, written new tools for AT&T's version of Unix. Over time, CSRG began to swap out parts of the system in favor of its own pieces, notably the file system and virtual memory, and was the first to add the TCP/IP protocols, giving the world the first Internet (really DARPA-net)-capable Unix system.

At about the same time, FSF had, supposedly, been developing its own operating system (Hurd), as well as a C compiler, linker, assembler, debugger, and editor. The effort to build tools worked out better for FSF than its effort to build an operating system, and, in fact, I have never seen a running version of Hurd, though I suspect this column will generate an email message or two pointing to a sad set of neglected files. The GNU tools were, in a way, an advancement, because now software developers could have an open source set of tools with which to build both new tools and systems. I say, "in a way," because these tools came with two significant downsides. To understand the first downside, you should find a friend who works on compilers and ask if he or she has ever looked inside gcc (GNU C compiler), and, after the crying stops and you have bolstered your friend's spirits, ask if he or she has ever tried to extend the compiler. If you are still friends at that point, your final question should be about submitting patches upstream into this supposedly open source project.

The second downside was religious: the GPL (GNU Public License). If you read *Hackers*, it becomes quite obvious why FSF created the GPL, and the copyleft before it. The people who created FSF felt cheated when others took the software they had worked on—and which was developed under various gov-

ernment grants—started companies, and tried to make money with it. The open source community is very clearly split over the purity of what one develops. There are those who believe no one should be able to charge for software or to close it off from others, and those who want to share their knowledge, whether or not the receiver of that knowledge makes a buck with it.

All of this background brings us to Linux and its relationship to the GNU tools. Linux is an operating system kernel, initially developed by Linus Torvalds in reaction to the Minix operating system from Andrew Tanenbaum. Torvalds used the GNU tools—compiler, linker, assembler—to turn his C code into an operating system kernel and then launched it upon the world. He released the code under a GPLv2 license—the one it maintains to this day—rather than taking on GPLv3, which is even more restrictive than its predecessors. Other people took up the code, modified it, improved it, and built new tools and systems around it.

Now, to the point about naming. When you build a house, you use many tools: hammers, saws, drills, and so forth. When the house is complete, do you call that a Craftsman/House, a Makita/House, or a Home Depot/House? Of course you don't. We do not name things after the tools we use to build them; we name things in ways that make sense because they describe the whole of the thing clearly and completely. Linux is an operating system kernel and some associated libraries that present a mostly, but not always, Posix, Unix-like system on top of which people write software. In point of fact, Linux distributions do not ship with the GNU tools, which must be installed from packages later. Linux is a thing unto itself, and the GNU tools are things unto themselves. That one might be used to work on the other is irrelevant, as is the fact that I am holding a Craftsman hammer from Home Depot right now ... must put the hammer down.

The whole GNU/Linux naming silliness is probably a case of kernel envy, not unlike the programmers who feel their ultimate achievement is to write a process or thread scheduler, which I addressed in my February 2014 column "Bugs and Bragging Rights" (<http://cacm.acm.org/>

[magazines/2014/2/171691-bugs-and-bragging-rights/fulltext](http://cacm.acm.org/magazines/2014/2/171691-bugs-and-bragging-rights/fulltext)), where I wrote, "I think the propensity for programmers to label their larger creations as operating systems comes from the need to secure bragging rights. Programmers never stop comparing their code with the code of their peers."

Why is this important? There are two reasons. One is intellectual honesty. KV prefers to see the credit go to those who did the work. Linus Torvalds and his team have built an important artifact, out of many tools, that many people use each day, so the credit goes to them, not to the producers of the tools. It takes more than a compiler, linker, and editor to build something as complex as an operating system, or even the operating system kernel, and many of the tools that go into building Linux have nothing at all to do with FSF or GNU. Should we now rename all of our systems as GNU/APACHE/FOO/BAR? Only a lawyer would think of that, and by now you all know what I think of letting lawyers name projects. The second reason this is important is to point out that while GNU stands for "GNU is not Unix," that was a reaction against the AT&T Unix of the 1980s. Now it might as well be "GNU is not Linux," because the tool is not the thing the tool builds. But then, GNL doesn't sound as good.

KV

P.S. If you want to read the GNU side of this story, pour yourself a strong beverage and start here: <http://www.gnu.org/gnu/linux-and-gnu.html>.

Related articles on queue.acm.org

Open Source to the Core

John Hubbard

<http://queue.acm.org/detail.cfm?id=1005064>

A License to Kode

George Neville-Neil

<http://queue.acm.org/detail.cfm?id=1217262>

Desktop Linux: Where Art Thou?

Bart Decrem

<http://queue.acm.org/detail.cfm?id=1005067>

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.

Technology Strategy and Management

The Need for Corporate Diplomacy

Whether global companies succeed or fail often depends on how effectively they develop and maintain cooperative relationships with other organizations and governments.

GOVERNMENTS SET RULES; businesses operate by following these rules. This idealized notion of political economy is more inaccurate today than ever before. Business leaders, including technology entrepreneurs, must participate in rulemaking due to deregulation and liberalization, prominent global risks (such as climate change and migration) that do not respect national borders, and digital technology that is spewing new issues requiring new rules. Business leaders are expected to be corporate diplomats.

Corporate diplomacy is not about turning businessmen into part-time politicians or statesmen. Rather, it involves corporations taking part in creating, enforcing, and changing the rules of the game that govern the conduct of business. It goes well beyond delegating external communications and lobbying to a public relations agency or a law firm. Precise understanding of corporate diplomacy would help businesses compete more effectively in the global economy. This column clarifies corporate diplomacy, its benefits and challenges.

Corporate Diplomacy: Taking Stock of History

Many formal rules govern the conduct of business. National regulators en-



Airbnb is one example of a “sharing economy” business that effectively used corporate diplomacy to defeat the recent Proposition F intended to restrict short-term rentals in San Francisco, CA.

force rules: in the U.S., the Patent and Trademark Office grants intellectual property rights, the Internal Revenue Service administers tax rules, and the Federal Trade Commission and the Department of Justice impose rules

on mergers and acquisitions. Other countries have similar regulators. In order to deal with cross-border business activities, national regulators create international treaties to mutually recognize each other’s rules, and es-

COMMUNICATIONS APPS

Access the latest issue, past issues, **BLOG@CACM**, News, and more.



Available for iPad, iPhone, and Android



Available for iOS, Android, and Windows

<http://cacm.acm.org/about-communications/mobile-apps>



Association for Computing Machinery

establish international organizations to harmonize rules. The world is far from flat, however. General Electric encountered this when its 2001 plan to acquire Honeywell was approved by the U.S. regulators, but faced fierce opposition from the European Commission. The enforcement of competition policy differs from country to country.

The recent history of liberalization and globalization points to the rising importance of corporate diplomacy. Since the end of the Cold War, national governments lost autonomy in a globalizing economy. They are sharing powers with businesses, international organizations, and nongovernmental organizations.¹ There is a steady decline in corporate income taxes, from 49.1% in 1981 to 32.5% in 2013 on average in Organisation for Economic Co-operation and Development (OECD) countries. States reduce taxes to make their locations more attractive to foreign direct investment, but in the process render themselves less resourceful to solve social problems. Business corporations engage in 'regime shopping' before choosing locations. Tax breaks raise public expectations that businesses will help solve societal problems. Civil society creates the rules of the game on 'fair taxes,' not government or business alone. Public protests against tax avoidance caused Google to pay the U.K. government £20 million in voluntary taxes in December 2012. This was a climb down by Google CEO Eric Schmidt, who earlier said that paying less tax was "just capitalism." Businesses that fail to engage proactively in corporate diplomacy are criticized and cannot establish their legitimate role in society.

This is not just a phenomenon of the

The recent history of liberalization and globalization points to the rising importance of corporate diplomacy.

past few decades. Corporate diplomacy has always been important in some locations. National borders are a fairly recent human invention. The Peace of Westphalia in 1648 created the basis for national self-determination and sovereignty, and throughout the 17th and 18th centuries, the Hudson Bay Company in North America and the East India Companies in India operated as company-states. They had authority to acquire territory, coin money, maintain forts and armies, make treaties, and administer justice. These functions of the state were carried out by private companies before colonial administration took over. Establishing and maintaining this took corporate diplomacy.

Corporate Diplomacy Hotspots

History gives pointers to corporate diplomacy opportunities and challenges. Corporate diplomacy matters irrespective of whether governments are present or absent. Governments present in regulated industries require corporations to seek license to operate and to comply with standards. Energy, mining, and infrastructure are good examples of such sectors, and companies in them have strong government affairs departments. Such sectors often are seen to embody national interest, and corporate diplomacy with host country governments is vital. The distinction between diplomacy and corporate diplomacy can be blurred. In 2013, Argentina expropriated the assets of Repsol YPF, the Argentinian subsidiary of the Spanish oil company. The Spanish parent used diplomacy with the government of Spain to pressure the government of Argentina for compensation. The failure of the Chinese oil company CNOOC to acquire the U.S. company Unacol in 2005 was due in part to failed corporate diplomacy when the U.S. Congress framed this acquisition as the Chinese state acting behind CNOOC. The company learned its lesson and successfully acquired Nexen in Canada. Through corporate diplomacy, both YPF and CNOOC reframed existing rules on national security.

Where governments do not exist, or have withdrawn (for example, via privatization or outsourcing) rule making and rule enforcement by businesses are also important. Government outsourc-

ing³ is rife with corporate diplomacy. Business corporations such as G4S and Serco bid for and negotiate the terms of the outsourcing contracts, and engage in subtle but important corporate diplomatic work to create rules to define the respective responsibilities of the government and the private sector. For example, rules on decent treatment of detainees and asylum seekers are prescribed in international human rights law, the signatories of which are nation-states. Yet, when a government outsources the management of immigration detention services to private sector firms, as the Australian government has done, those firms become responsible de facto for enforcing the law.

Corporate Diplomacy in the Digital Economy

Digital technology creates a significant corporate diplomatic hotspot. Information and communication technologies have challenged existing rules for intellectual property, privacy, and data security. It has also challenged competition policy with network externalities, giving rise to charges of monopolistic behaviors by Microsoft, Amazon, and Google. No wonder, lobbying by corporate America has spread from the old economy to the new economy. In 2012, Google was the second biggest corporate lobby in Washington D.C., spending \$18.2 million. (GE was first, spending \$21.4 million.⁴) Technology firms now have a significant presence in Washington, D.C. Corporate diplomacy has become important in this sector.

Technology startups used to disregard corporate diplomacy. Uber started in 2010 offering an online chauffeur service that enabled customers to book a ride quickly using a mobile device. Uber did not own the cars but contracted with private car owners and drivers. Uber was neither a taxi service nor a limousine service. Its business did not fit the conventional regulatory framework that usually regulated taxis and limousines separately. Uber often ignored regulations in a city and just started operations to avoid lengthy regulatory approvals. It built a presence and proved its value to users, relying on citizen support for its commercial success. It is useful to explore the case of Uber to see why and how corpo-

It is useful to explore the case of Uber to see why and how corporate diplomacy became important.

rate diplomacy became important.

Peer-to-peer collaboration can make consumer choice and sovereignty paramount. Uber and other companies claim that the “sharing economy” they are building can self-regulate. Consumers ‘vote with their money’, making regulatory oversight redundant. Ratings by users are a transparent self-regulatory mechanism to make the market function well. Irrespective of whether these claims are true, service professions (including medical, legal, financial) have self-regulation that operates in the shadow of government regulation to be effective. Self-regulating professions operate only as long as they are seen to be acting as trustees of public interest. Ignoring governments is not sustainable.

Uber’s growth has been impressive: it now operates in over 300 cities in 67 countries. However, it failed to thwart bans or partial bans in cities in Australia, Germany, India, and Thailand. In 2015, Uber hired David Plouffe (political strategist and former campaign manager for President Obama) and Rachel Whetstone (Google’s head of communications), to boost the company’s public policy team and to maximize smooth sailing with city regulators. Uber’s political strategy must deal with important vested interests, notably licensed taxi drivers. Consumer groups who benefit from more convenient and cheaper rides can help thwart bans. Supporting the green ‘low emission’ agenda of some cities can help as well. Corporate diplomacy is required to manage these stakeholders.

Because of experiences like this Silicon Valley startups are taking corporate diplomacy seriously. Uber, Airbnb, and other firms with the “sharing economy” business model must create new rules of the game. It

is better to influence the creation of new rules proactively than have inimical rules imposed.

Conclusion

Many people might agree with Ronald Reagan’s quip “the nine most terrifying words in the English language are: ‘I’m from the government and I’m here to help.’” Efforts to keep the government at bay can create a blind spot with respect to corporate diplomacy.

Business leaders must negotiate with governments to influence rules that affect their environment to their advantage. In doing so, they do well to recognize that as corporate diplomats, they establish norms that legitimize the conduct of business. Thus, business leaders participate in building institutions, which are both formal rules and social norms.²

Corporate diplomacy happens in areas where governments are present as regulators, service providers, and owners of assets. Corporate diplomacy is equally needed where governments are absent due to deregulation or weak law enforcement capacity. It is also required in new markets with new technology where rules are yet to be made. Creating new rules requires tactics somewhat different from conventional lobbying. Corporate diplomacy is a mind-set that sees the role of business as working *with* governments to create societal rules that govern the conduct of business. Corporate diplomats should not scare public officials by stating: “I’m from the corporation and I’m here to help.” Keeping governments at bay does not guarantee business success, nor will keeping corporations at bay lead to successful regulation. Corporate diplomacy is a promising way forward for understanding how to create and change rules for better outcomes. ■

References

1. Matthews, J.T. Power shift. *Foreign Affairs* (Jan./Feb. 1997).
2. North, D.C. *Institutions, Institutional Change and Economic Performance*. Cambridge University Press, New York, 1990.
3. Sako, M. The business of the state. *Commun. ACM* 57, 7 (July 2013), 28–30.
4. The Washington wishing-well. Schumpeter column, *The Economist* (June 13, 2015).

Mari Sako (mari.sako@sbs.ox.ac.uk) is Professor of Management Studies at Saïd Business School, University of Oxford, U.K.

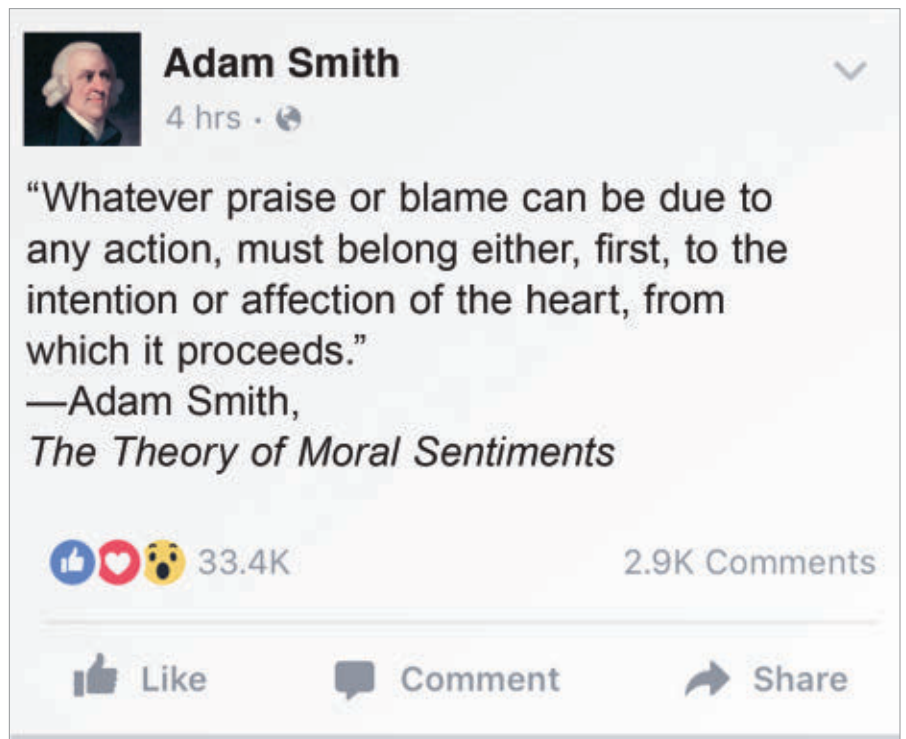
Copyright held by author.

Viewpoint Beyond Viral

The proliferation of social media usage has not resulted in significant social change.

THE GOLDEN AGE of social media coincides with a worldwide leadership crisis, manifested by our seeming inability to address any major global issue in recent years.³² These days, no one—be they a charismatic leader or a nameless crowd—seems to be able to make issues popular for long enough to mobilize society into action. As a result of this leadership vacuum, social progress of all sorts seems to have become stymied and frozen. How can this happen precisely in a time when social media, praised as the ultimate tool to raise collective awareness and mobilize society, has reached maturity and widespread use? Here, we argue the coexistence of social media technologies with “The End of Power”¹⁸ is anything but a coincidence, presenting the first techno-social paradox of the 21st century.

In recent years, we have witnessed social media playing a major role in social mobilization events of historic proportions, such as the Arab Spring, the Occupy Wall Street movement, Ukraine’s Euromaidan, and the chaos generated by the England Riots and Boston Marathon bombing manhunt. There has been substantial emphasis on the role of digital social media platforms, particularly Facebook and Twitter, as the facilitators of these mobilizations. Data availability has made it possible, for the first time, to observe the evolution of these events in detail.^{10,11,13,33} Analysis of these events makes it clear that political activists find it difficult to use social media to create mass mobilization; and even when they succeed it is difficult to



sustain the focus of the protest until it is able to mobilize politicians, institutions, and society at large. As a result, most of these events burst upon the scene, occupy our attention for a few days, and then fade into oblivion with nothing substantial having been accomplished. Given all we have learned about social mobilization, why isn’t social media a more reliable channel for constructive social change?

A related observation is that national intelligence agencies are failing to anticipate social uprisings, even when they extensively monitor personal social media networks. Recent global surveillance leaks from Edward Snowden

and others show not a single instance where analysis of social media predicted a social uprising or public movement. Social media has been much better at providing the fuel for unpredictable, bursty mobilization than at steady, thoughtful construction of sustainable social change.

Coordinated collective action is a fundamental aspect of all collective intelligence and social decision-making processes. However, despite progress on understanding social mobilization processes we are still a long way from developing a reliable, quantitative theory. In other words, we have developed models able to predict the

online spread of ideas and news, yet we lack models to predict the behavior change produced by this very same campaign. We argue these failures of use and prediction are not caused by a lack of expertise in data analysis, but by an insufficient focus on the underlying incentive structures—the hidden network of interpersonal motivations that provide the engine for collective decision making and action.

A number of large-scale social mobilization experiments have revealed the important role of incentive structures in realistic, adversarial settings. These planetary-scale experiments include the DARPA Network Challenge to locate 10 weather balloons tethered at random locations all over the continental U.S., which was won by our team using a recursive incentive scheme to recruit an estimated two million searchers within 48 hours; the DARPA Shredder Challenge, in which we recruited over 3,500 individuals to collaboratively assemble real shredded documents; and the most recent U.S. State Department's Tag Challenge, in which we recruited volunteers to locate individuals "at large" in remote cities within 12 hours and won again using the very same incentive scheme. In each challenge, all competing teams had the same type of message (that is, find the balloons, assemble shreds, find the target individuals), and many of them managed to create viral campaigns that reached large populations and created awareness, yet the efficiency of the strategies varied widely and was strongly correlated with the manner in which their incentive design matched the motivations of the participants. Even in the simple task of finding balloons, we saw teams tapping into people's incentives toward personal profit, charity, reciprocity, or entertainment, with varying degrees of success. Some incentive structures posed by competing teams were compatible with the internal incentive structures of the individuals, and could therefore switch them 'on', activating a network cascade of actions, whereas others did not succeed to do so.

We believe incentive networks play an important middle layer between higher-order concepts such as ideologies and culture, and the digital finger-

Why isn't social media a more reliable channel for constructive social change?

prints left by social movements in online digital platforms such as Twitter and Facebook. Ideologies and culture shape what individuals want to achieve as they go about their daily life, how they relate to each other's well-being, and how they help each other achieve those goals. This can be mapped into a network of incentives where each individual payoff depends on others individual payoff. Incentive structures are shaped by more abstract underlying processes, but can be mapped quantitatively by these large-scale collective action experiments.

The inability to sustain and transfer bursts of social mobilization in order to create lasting social change is rooted in the design of today's digital social media. Today's social media is designed to maximize information propagation and virality (through optimization of clicks and shares) to the detriment of engagement and consensus building. For instance, Onnela and Reed-Tsochas¹⁹ demonstrate that even when external signals are absent, digital social influence spontaneously assumes an unstable all-or-nothing nature. The result is "flash fads," the ever-changing inception, competition, and death of new fads that annihilate each other, as they compete for people's attention, with no long-lasting result.³¹ Effective social mobilization is a product of both information diffusion and action recruitment incentives, yet the pressures of the social media business have focused on diffusion to the detriment of incentives for recruiting people to act. Even from the business perspective, social media is extraordinarily ineffective at the goal of recruitment to action, for example, clicking through ads to purchase. The best

Calendar of Events

April 3–6

ISPD'16: International Symposium on Physical Design, Santa Rosa, CA,
Sponsored: ACM/SIG,
Contact: Fung Yu Young,
Email: fyyoung@cse.cuhk.edu.hk

April 4–8

SAC 2016: Symposium on Applied Computing, Pisa, Italy,
Sponsored: ACM/SIG,
Contact: Sascha Ossowski,
Email: sascha.ossowski@urjc.es

April 11–14

CPS Week '16: Cyber Physical Systems Week 2016, Vienna, Austria,
Contact: Radu Grosu,
Email: grosu@cs.sunysb.edu

April 12–14

HSCC'16: 19th International Conference on Hybrid Systems: Computation and Control (part of CPS Week), Vienna, Austria,
Contact: Alessandro Abate,
Email: a.abate@tudelft.nl

April 12–14

ICCPs '16: ACM/IEEE 7th International Conference on Cyber-Physical Systems (with CPS Week 2016), Vienna, Austria,
Contact: Ian Mitchell,
Email: mitchell@cs.ubc.ca

April 12–14

IPSN '16: The 14th International Conference on Information Processing in Sensor Networks (co-located with CPS Week 2016), Vienna, Austria,
Contact: George J. Pappas,
Email: pappasg@seas.upenn.edu

April 18–21

EuroSys '16: 11th EuroSys Conference 2016, London, U.K.,
Sponsored: ACM/SIG,
Contact: Peter R Pietzuch,
Email: prp@doc.ic.ac.uk

minds of our generation may no longer be thinking about how to make people click ads (as Hammerbacher famously said in 2013),¹⁵ but they have only progressed to thinking about how to make people click ‘share’ and ‘like’.

The bias of commercial social media toward virality has led most researchers and practitioners studying social movements to focus on the dynamics of information diffusion, with particular focus on conditions that cause viral information propagation. But reliable a priori prediction of which content ‘goes viral’ does not seem to be within reach. Leading network science scholars like Duncan Watts,³⁰ Jon Kleinberg,¹⁷ and Matthew Jackson¹² have long argued that viral propagation is highly unpredictable, and that our selective observation of successful campaigns provides us with a false narrative of its underlying causes.

Furthermore, although it is possible to engineer ‘viral features’ into products,² viral propagation usually has more to do with the incentives’ underlying message spreading than with the message itself, especially in contested

domains such as politics. Recruitment of people via content creation has been a craft since its inception,⁴ and it is likely to stay as a craft industry for the foreseeable future, given its dependence on immediate individualized sociocultural context. In contrast, if we shift our efforts toward the mapping of incentives, then we may be able to better determine the suitability of content for recruitment to action and to create lasting social change.

In addition to the bias of commercial social media toward virality, research may overemphasize virality because of two pragmatic considerations. First, equating social mobilization with viral information propagation renders the phenomenon amenable to analysis using tools from epidemiology and public health.^{9,24} However, this epidemiological perspective is only useful in a population with conducive socio-political incentives, that is, a society already “switched on.” The second reason behind the emphasis on information virality is a phenomenon we may dub *network measurability bias*, which refers to the tendency to focus on processes that are easily observable within digital social networks (such as ‘likes’ and ‘re-tweets’), while neglecting key latent processes such as the ideological, cultural, and economic incentives of actors. Social media is an amazing new instrument that allows social scientists to measure social information spread in real time, yet is almost totally blind to other relevant factors,³³ such as framing processes,⁶ reflection,⁵ consensus formation, or argumentation processes,^{23,25} which are important in connecting content to sustained motivation.

Much progress has been made in understanding incentives in the economic, social, and political sciences. Hurwicz, Maskin, and Myerson received the 2007 Nobel Prize in Economic Sciences for their development of Mechanism Design, a mathematical toolbox designed to uncover and leverage the true preferences of individuals participating in strategic interactions. In addition, laboratory experiments have recently been able to identify how social structure and dynamics shape incentives via stylized repeated cooperation games

such as the Prisoner’s Dilemma and Ultimatum Games.^{21,29} These strategic scenarios may well be far from the incentives that move people in the real world,²⁷ but they can serve as a first “probe” to experimentally uncover dynamic incentive networks, and provide a complement to large-scale social network experiments geared toward behavioral change.^{8,14,26}

Information spreading is key to the formation of collective beliefs, opinions, and attitudes. But incentives play an equally important role. Convincing someone of an idea is one thing. Recruiting them to incur substantial time, effort, and risk toward supporting a cause requires much more. What is needed are new experimental paradigms and observational tools that elicit not only communication dynamics, but also the dynamics of underlying individual, social, and cultural incentives operating in social mobilization processes. Results from these experiments should help us develop a new generation of social media, which can go beyond flash fads and viral memes toward consensual construction of sustained change.

Individuals are not atoms. Without the correct incentive structure, a group of individuals cannot mobilize into a sophisticated problem-solving crowd, let alone change society. This is the tragedy of a completely open and equally connected society: when people discuss social issues online, it is very difficult to reliably quantify the importance of the different issues being raised. Both awareness (how many people care about a certain issue) and persistence (how long do they care about this issue) exhibit heavy-tailed distributions.^{3,7,16,20} This makes it difficult for citizens, including scientists studying the phenomenon, to establish clear thresholds of importance for prioritizing among the myriad of potential issues. Without meaningful thresholds for action, the set of alternative issues end up canceling each other out, leading to ‘slacktivism’, and leaving military or economic force the only path to change.

Individual and collective attention are finite, and the capacity of media platforms and their algorithms’ ability to infer, manipulate, and capture attention seems to improve continuous-

São Paulo School of Advanced Science on



Algorithms, Combinatorics and Optimization

July 18–29, 2016 University of São Paulo, Brazil
<http://sp-school2016.ime.usp.br>

Sponsored by FAPESP, the school will host courses on advanced research topics and is aimed at MSc and PhD students, young researchers, and exceptional undergraduate students. Participants may apply for local expenses and travel grants.

Lecturers

R. Kleinberg (Cornell)	Y. Kohayakawa (USP)
A. Kostochka (UIUC)	D. Král’ (Warwick)
C.L. Lucchesi (UFMS)	F.K. Miyazawa (UNICAMP)
R. Morris (IMPA)	F.M. de Oliveira (USP)
S. Robins (USP)	L. Tunçel (Waterloo)
E. Upfal (Brown)	D. Williamson (Cornell)

Speakers

C.C. de Souza (UNICAMP)	K. Jansen (Kiel)
M. Kiwi (U. Chile)	B. Reed (McGill)
J.L. Szwarcfiter (UFRJ)	



ly. But without social media that also promotes complex coordination and institution building, in the end nothing is achieved. We need a deeper understanding of how to tap into network incentives, and for activating the right incentives through information filtering and consensus building.

However, unlike message content and social network structure, incentives are far less visible. They manifest themselves through the actions of individuals, and often a particular action comes from multiple incentives. Before we produce a practical theory of social mobilization, we need to develop new ways of measuring, influencing, and modeling incentives in networks, and for interpreting individual action in their light. Our efforts in the large-scale mobilization challenges are only a first small step in that direction.

Adam Smith is considered by many to be the intellectual father of the idea that only observable actions matter: people act in the market, and an invisible hand produces an efficient outcome without knowing the private information and motivations behind people's actions. But in his *Theory of Moral Sentiments*, Smith made it very clear that a true understanding of social phenomena must incorporate the multitude of psychological and cultural motives. By moving our attention from observable viral processes to modeling their underlying motivational dynamics, we would pay tribute to Smith's nuanced understanding of human nature. And, perhaps, along the way, design the next generation of social media. C

References

- Alstott, J. et al. Homophily and the speed of social mobilization: The effect of acquired and ascribed traits. *PLOS ONE* 9, 4 (2014), e95140.
- Aral, S. and Walker, D. Forget viral marketing—Make the product itself viral. *Harvard Business Review* (2011), 34–35.
- Bakshy, E. et al. Everyone's an influencer: Quantifying influence on Twitter. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*. ACM, 2011.
- Bartels, R. *The History of Marketing Thought*. Publishing Horizons, Columbus, OH, 1988.
- Baumer, E.P. et al. Reviewing reflection: On the use of reflection in interactive system design. In *Proceedings of the 2014 Conference on Designing Interactive Systems* (2014), ACM, 93–102.
- Benford, R.D. and Snow, D.A. Framing processes and social movements: An overview and assessment. *Annual Review of Sociology*, (2000), 611–639.
- Blumm, N. et al. Dynamics of ranking processes in complex systems. *Physical Review Letters* 109, 12 (2012), 128701.
- Bond, R.M. et al. A 61-million-person experiment in social influence and political mobilization. *Nature* 489, 7415 (2012), 295–298.
- Braha, D. Global civil unrest: Contagion, self-organization, and prediction. *PLOS ONE* 7, 10 (2012), e48596.
- Conover, M.D. et al. The digital evolution of occupy Wall street. *PLOS ONE* 8, 5 (2013), e64679.
- Conover, M.D. et al. The geospatial characteristics of a social movement communication network. *PLOS ONE* 8, 3 (2013), e55957.
- Golub, B. and Jackson, M.O. Using selection bias to explain the observed structure of Internet diffusions. In *Proceedings of the National Academy of Sciences* 107, 24 (2010), 10833–10836.
- González-Bailón, S. et al. The dynamics of protest recruitment through an online network. *Scientific Reports* 1 (2011).
- Gutiérrez-Roig, M. et al. Transition from reciprocal cooperation to persistent behaviour in social dilemmas at the end of adolescence. *Nature Communications* 5 (2014).
- Hammerbacher, J. Charlie Rose and Jeff Hammerbacher talk Data Science in Healthcare; <http://www.cloudera.com/content/cloudera/en/resources/library/aboutcloudera/jeff-hammerbacher-charlie-rose.html>
- Karsai, M. et al. Small but slow world: How network topology and burstiness slow down spreading. *Physical Review E* 83, 2 (2011), 025102.
- Liben-Nowell, D. and Kleinberg, J. Tracing information flow on a global scale using Internet chain-letter data. In *Proceedings of the National Academy of Sciences*, 105, 12 (2008), 4633–4638.
- Naim, M. *The End of Power: From Boardrooms to Battlefields and Churches to States, Why Being In Charge Isn't What It Used to Be*. Basic Books, 2014.
- Onnela, J.P. and Reed-Tsochas, F. Spontaneous emergence of social influence in online systems. In *Proceedings of the National Academy of Sciences* 107, 43 (2010), 18375–18380.
- Papadopoulos, F. et al. Popularity versus similarity in growing networks. *Nature* 489, 7417 (2012), 537–540.
- Peysakhovich, A. et al. Humans display a 'cooperative phenotype' that is domain general and temporally stable. *Nature Communications* 5 (2014).
- Pickard, G. et al. Time-critical social mobilization. *Science* 334, 6055 (2011), 509–512.
- Rahwan, I. et al. Laying the foundations for a world wide argument web. *Artificial Intelligence* 171, 10 (Oct. 2007), 897–921.
- Rutherford, A. Limits of social mobilization. In *Proceedings of the National Academy of Sciences* 110, 16 (2013), 6281–6286.
- Schneider, J. et al. A review of argumentation for the social semantic Web. *Semantic Web* 4, 2 (Feb. 2013), 159–218.
- Shirado, H. et al. Quality versus quantity of social ties in experimental cooperative networks. *Nature Communications* 4 (2013).
- Stefanovitch, N. et al. Error and attack tolerance of collective problem solving: The DARPA shredder challenge. *EPJ Data Science* 3, 13 (2014).
- Tang, M. et al. Reflecting on the DARPA red balloon challenge. *Commun. ACM* 54, 4 (Apr. 2011), 78–85.
- Tsvetkova, M. and Macy, M.W. The social contagion of generosity. *PLOS ONE* 9, 2 (2014), e87275.
- Watts, D.J. *Everything Is Obvious: How Common Sense Fails Us*. Random House LLC, 2012.
- Weng, L. et al. Competition among memes in a world with limited attention. *Scientific Reports* 2 (2012).
- World Economic Forum (WEF). *Outlook on the Global Agenda, 2014*; <http://reports.weforum.org/outlook-global-agenda-2015/>
- Zuckerman, E. The first Twitter revolution? *Foreign Policy* 14 (2011).

Manuel Cebrian (manuel.cebrian@data61.csiro.au) is Research Group Leader with the Data61 Unit at the Commonwealth Scientific and Industrial Research Organisation (CSIRO), Australia.

Iyad Rahwan (irahwan@mit.edu) is an associate professor of Media Arts and Sciences at the Media Lab, Massachusetts Institute of Technology.

Alex "Sandy" Pentland (pentland@mit.edu) directs the MIT Connection Science and Human Dynamics labs and previously helped create and direct the MIT Media Lab and the Media Lab Asia in India.

Copyright held by authors.

INTERACTIONS



ACM's *Interactions* magazine explores critical relationships between people and technology, showcasing emerging innovations and industry leaders from around the world across important applications of design thinking and the broadening field of interaction design.

Our readers represent a growing community of practice that is of increasing and vital global importance.



To learn more about us, visit our award-winning website <http://interactions.acm.org>

Follow us on Facebook and Twitter



To subscribe: <http://www.acm.org/subscribe>

Association for Computing Machinery



Article development led by **acmqueue**
queue.acm.org

Retaining electronic privacy requires more political engagement.

BY POUL-HENNING KAMP

More Encryption Means Less Privacy

WHEN EDWARD SNOWDEN made it known to the world that pretty much all traffic on the Internet was collected and searched by the U.S. National Security Agency (NSA), the U.K. Government Communications Headquarters (GCHQ), and various other countries' secret services as well, the IT and networking communities were furious and felt betrayed.

A wave of activism followed to get traffic encrypted so as to make it impossible for NSA to indiscriminately snoop on the entire world population. When all you have is a hammer, all problems look like nails, and the available hammer was the SSL/TLS encryption protocol, so the battle cry was “SSL/TLS/HTTPS everywhere.” A lot of nails have been hit with that!

After an animated plenary session in Vancouver, the Internet Engineering Task Force (IETF) published “Best Current Practice 188” (<https://tools.ietf.org/html/bcp188>), which declared that pervasive monitoring is a technical attack that should be mitigated in the design of IETF protocols where possible. Now, with this manifesto in hand, SSL/TLS and encryption are being hammered into and bolted onto protocols and standards throughout the IETF working groups.

Victory—*privacy*—seemed certain. Or maybe not.

► Kazakhstan recently announced that a “state root certificate” would have to be installed on all computers





wanting to use SSL/TLS/HTTPS out of the country.

► France's ministry of the interior is working on banning free WiFi connections and the use of the Tor protocol and networks.

► President Obama urged high-tech and law enforcement leaders to make it more difficult for terrorists to use technology to escape from justice.

Other countries, notably the U.K., are also working to clamp down on encryption. The Great Firewall of China has been in operation for a number of years, and for all we know, the NSA's total monitoring of the Internet continues unabated 2.5 years after Snowden revealed it to the world. The things worth noting here are:

► Kazakhstan did not just require criminals to install the "state root certificate" so their communications could be scrutinized, it required everybody in Kazakhstan to do so.

► France will not just ban criminals from using free WiFi and Tor, it will ban anybody and everybody from using them.

► While Obama wants to make it "harder for terrorists," I don't think he contemplates Apple offering an "OS X terrorist edition" or that terrorists will take an FBI-sponsored "Are you a terrorist?" quiz to find out if they should be using it.

Whatever the high-tech and law enforcement leaders decide, it will apply to everybody.

How Did More Encryption Cause Less Privacy?

In Terry Pratchett's *Going Postal*, the hero postmaster, Moist von Lipwig, has a knack for noticing what is not in a text. He would have had a field day with BCP188 because none of the following words are anywhere to be found:

- law
- court
- crime
- human
- secret
- warrant
- espionage
- constitution
- jurisdiction

It was not by accident, mind you, the authors of the document deliberately

stayed clear of anything that could even faintly smell of “politics.” Unfortunately, that is not the way politics works. Politics springs into action the moment somebody disagrees with you because of their political point of view, even if you think you do not have a political point of view.

In spite of leaving out all those “hot” words, the substance of BCP188 is still a manifesto declaring a universal human right to absolute privacy in electronic communications—no matter what.

That last bit is half the trouble—no matter what.

Even against law enforcement.

Even if law enforcement has a court order.

Even if ...

No matter what.

To be totally fair, BCP188 nowhere states “no matter what.” The real reason the result ends up being “no matter what” is the SSL/TLS protocol, when properly configured, works as advertised: there is no way to break it.

The other half of the trouble is the hallmark of a civilized society is a judicial system that can right wrongs, and therefore human rights are always footnoted. The United Nation’s Human Rights Charter has §29.2, which explains:

“In the exercise of his rights and freedoms, everyone shall be subject only to such limitations as are determined by law solely for the purpose of securing due recognition and respect for the rights and freedoms of others and of meeting the just requirements of morality, public order and the general welfare in a democratic society.”

Politicians, whose jobs are to maintain “public order” and improve “the general welfare,” follow the general principle that if criminals can use X to commit crimes, the legal system should be able to use X to solve crimes, with only two universally recognized exemptions: when “X = your brain” and when “X = your spouse.”

For instance, U.S. kids learn in school that the Fourth Amendment affords a right to privacy, but that is only the first half of it. The second half details precisely how and why you may lose that privacy:

“The right of the people to be secure in their persons, houses, papers, and effects, against unreasonable search-



When all you have is a hammer, all problems look like nails, and the available hammer was the SSL/TLS encryption protocol, so the battle cry was “SSL/TLS/HTTPS everywhere.” A lot of nails have been hit with that!



es and seizures, shall not be violated, and no Warrants shall issue, but upon probable cause, supported by Oath or affirmation, and particularly describing the place to be searched, and the persons or things to be seized.”

As this example also shows, wise lawmakers are wary of making it too easy for the legal system, so they add checks and balances.

Political strategies regarding cryptography are all horrible: Kazakhstan brutally inserts state monitors into the middle of all encrypted traffic. France forbids all online anonymity. The U.S. wants backdoors built into all crypto. These ideas are all based on the same principle: If we cannot break the crypto for a specific criminal on demand, we will preemptively break it for everybody. And whatever you may feel about politicians, they do have the legitimacy and power to do so. They have the constitutions, legislative powers, courts of law, and police forces to make this happen.

The IT and networking communities overlooked a wise saying from soldiers and police officers: “Make sure the other side has an easier way out than destroying you.”

But we didn’t, and they are.

Slapping unbreakable crypto onto more and more packets is just going to make matters worse. The only way to retain any amount of electronic privacy is through political engagement. 

Related articles on queue.acm.org

More Encryption Is Not the Solution

Poul-Henning Kamp

<http://queue.acm.org/detail.cfm?id=2508864>

Hickory Dickory Doc

George Neville-Neil

<http://queue.acm.org/detail.cfm?id=2791303>

Compliance Deconstructed

J. C. Cannon and Marilee Byers

<http://queue.acm.org/detail.cfm?id=1160449>

Poul-Henning Kamp (phk@FreeBSD.org) is one of the primary developers of the FreeBSD operating system, which he has worked on from the very beginning. He is widely unknown for his MD5-based password scrambler, which protects the passwords on Cisco routers, Juniper routers, and Linux and BSD systems.

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



**Sometimes all you need
is the right language.**

BY CARLOS BAQUERO AND NUNO PREGUIÇA

Why Logical Clocks Are Easy

ANY COMPUTING SYSTEM can be described as executing sequences of actions, with an *action* being any relevant change in the state of the system. For example, reading a file to memory, modifying the contents of the file in memory, or writing the new contents to the file are relevant actions for a text editor. In a distributed

system, actions execute in multiple locations; in this context, actions are often called events. Examples of events in distributed systems include sending or receiving messages, or changing some state in a node. Not all events are related, but some events can cause and influence how other, later events occur. For example, a reply to a received email message is influenced by that message, and maybe by prior messages received.

Events in a distributed system can occur in a close location, with different processes running in the same machine, for example; or at nodes inside a datacenter; or geographically spread across the globe; or even at a larger scale in the near future. The relations of potential cause and effect between events are fundamental to the design of distributed algorithms. These days hardly any service can claim not to have some form of dis-

tributed algorithm at its core.

To make sense of these cause-and-effect relations, it is necessary to limit their scope to what can be perceived inside the distributed system itself—*internal causality*. Naturally, a distributed system interacts with the rest of the physical world outside of it, and there are also cause-and-effect relations in that world at large. For example, consider a couple planning a night out using a system that manages reservations for dinner and a movie. One person makes a reservation for dinner and lets the other person know with a phone call. After receiving the phone call, the second person goes to the system and reserves a movie. A distributed system has no way of knowing the first reservation has actually caused the second one.

This *external causality* cannot be detected by the system and can only be approximated by *physical time*.

(Time, however, totally orders all events, even those unrelated—thus, it is no substitute for causality—and wall clocks are never perfectly synchronized.^{11,16}) This article focuses instead on internal causality—the type that can be tracked by the system.

Happened-Before Relation

In 1978, Leslie Lamport defined a partial order, referred to as *happened*

before, that connects events of a distributed system that are potentially causally linked.⁸ An event c can be the cause of an event e , or c happened before e , iff (if and only if) both occur in the same node and c executed first, or, being at different nodes, if e could know about the occurrence of c thanks to some message received from some node that knows about c . If neither event can know about the

other, then they are said to be concurrent.

Using the example of dinner and movie reservations, Figure 1 shows a distributed system with three nodes. An arrow between nodes represents a message sent and delivered. Both Bob's positive answer to the dinner suggestion by Alice and Chris's later request to join the party are influenced by Alice's initial question about plans for dinner.

In this distributed computation, a simple way to check if an event c could have caused another event e (c happened before e) is to find at least one directed path linking c to e . If such a connection is found, this partial order relation is marked $c \rightarrow e$ to denote the happened-before relation or potential causality. Figure 1 has $a_1 \rightarrow b_2$ and $b_2 \rightarrow c_3$ (and, yes, also $a_1 \rightarrow c_3$, since causality is transitive). Events a_1 and c_2 are concurrent (denoted $a_1 \parallel c_2$), because there are no causal paths in either direction. Note $x \parallel y$ if and only if $x \nrightarrow y$ and $y \nrightarrow x$. The fact Chris was bored neither influenced Alice's question about dinner, not the other way around.

Thus, the three possible relations between two events x and y are: (a) x might have influenced y , if $x \rightarrow y$; (b) y might have influenced x , if $y \rightarrow x$; (c) there is no known influence between x and y , as they occurred concurrently $x \parallel y$.

Causal Histories

Causality can be tracked in a very simple way by using *causal histories*.^{3,14} The system can locally assign unique names to each event (for example, node name and local increasing counter) and collect and transmit sets of events to capture the known past.

For a new event, the system creates a new unique name, and the causal history consists of the union of this name and the causal history of the previous event in the node. For example, the second event in node C is assigned the name c_2 , and its causal history is $H_c = \{c_1, c_2\}$ (shown in Figure 2). When a node sends a message, the causal history of the send event is sent with the message. When the message is received, the remote causal history is merged (by set union) with the local history. For example, the delivery of the first message from node A to B

Figure 1. Happened-before relation.

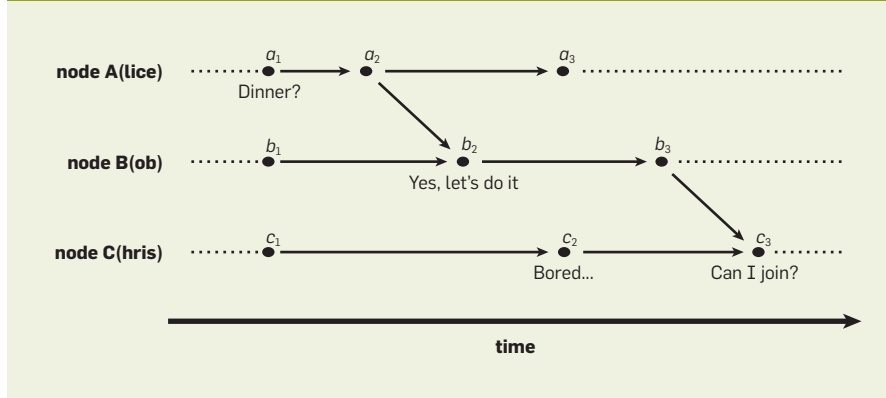


Figure 2. Causal histories.

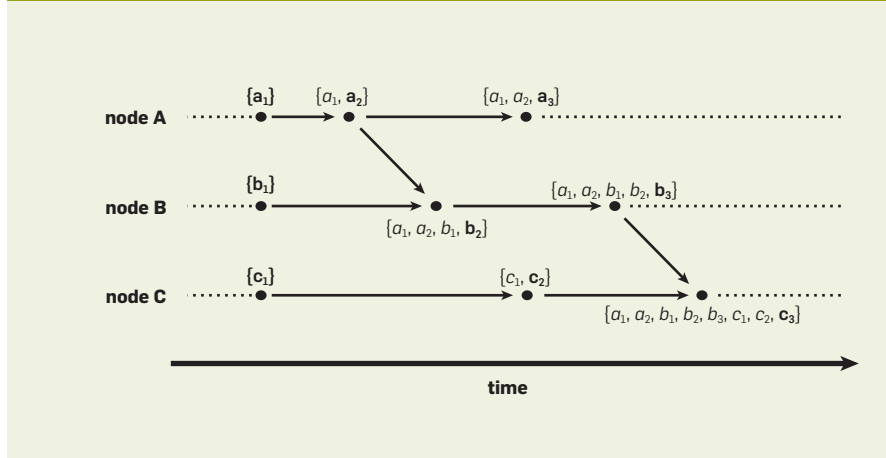
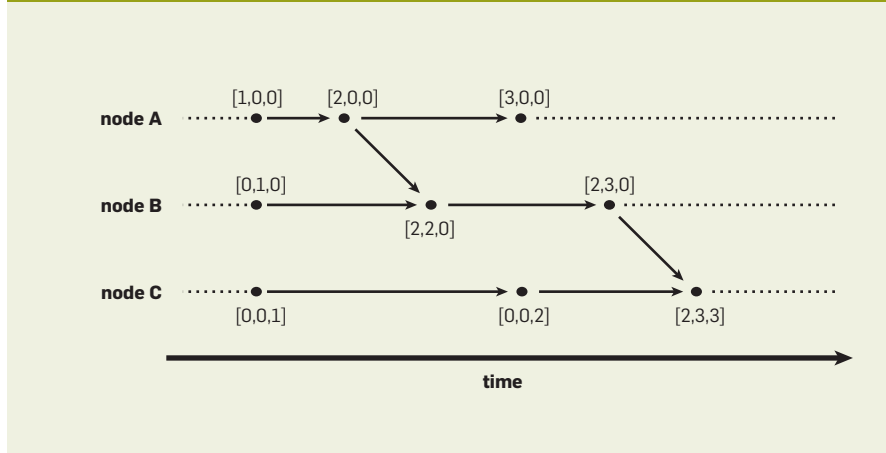


Figure 3. Vector clocks.



merges the remote causal history $\{a_1, a_2\}$ with the local history $\{b_1\}$ and the new unique name b_2 , leading to $\{a_1, a_2, b_1, \mathbf{b}_2\}$.

Checking causality between two events x and y can be tested simply by set inclusion: $x \rightarrow y$ iff $H_x \subseteq H_y$. This follows from the definition of causal histories, where the causal history of an event will be included in the causal history of the following event. Even better, marking the last local event added to the history (distinguished in bold in the figure) allows the use of a simpler test: $x \rightarrow y$ iff $x \in H_y$ (for example, $a_1 \rightarrow b_2$, since $a_1 \in \{a_1, a_2, b_1, \mathbf{b}_2\}$). This follows from the fact a causal history includes all events that (causally) precede a given event.

Causality Tracking

It should be obvious by now that causal histories work but are not very compact. This problem can be addressed by relying on the following observation: the mechanism of building the causal history implies if an event b_3 is present in H_y , then all preceding events from that same node, b_1 and b_2 , are also present in H_y . Thus, it suffices to store the most recent event from each node. Causal history $\{a_1, a_2, b_1, b_2, b_3, c_1, c_2, c_3\}$ is compacted to $\{a \mapsto 2, b \mapsto 3, c \mapsto 3\}$ or simply a vector $[2, 3, 3]$.

Now the rules used with causal histories can be translated to the new compact vector representation.

Verifying that $x \rightarrow y$ requires checking if $H_x \subseteq H_y$. This can be done, verifying for each node, if the unique names contained in H_x are also contained in H_y and there is at least one unique name in H_y that is not contained in H_x . This is immediately translated to checking if each entry in the vector of x is smaller or equal to the corresponding entry in the vector of y and one is strictly smaller (such as, $\forall i: V_x[i] \leq V_y[i]$ and $\exists j: V_x[j] < V_y[j]$). This can be stated more compactly as $x \rightarrow y$ iff $V_x < V_y$.

For a new event the creation of a new unique name is equivalent to incrementing the entry in the vector for the node where the event is created. For example, the second event in node C has vector $[0, 0, 2]$, which corresponds to the creation of event c_2 of the causal history.

Finally, creating the union of the two causal histories H_x and H_y is equivalent to taking the pointwise maximum of the corresponding two vectors V_x and V_y (such as, $\forall i: V[i] = \max(V_x[i], V_y[i])$). Logic tells us that, for the unique names generated in each node, only the one with the largest counter needs to be kept.

When a message is received, in addition to merging the causal histories, a new event is created. The vector representation of these steps can be seen, for example, when the first message from a is received in b , where taking the pointwise maximum leads to $[2, 1, 0]$ and the new unique name finally leads to $[2, 2, 0]$, as shown in Figure 3.

This compact representation, known as a *vector clock*, was introduced around 1988.^{5,10} Vector comparison is an immediate translation of set inclusion of causal histories. This equivalence is often forgotten in modern descriptions of vector clocks and can turn what is a simple encoding problem into an unnecessarily complex and arcane set of rules, going against logic.

As shown thus far, when using causal histories, knowing the last event could simplify comparison by simply checking if the last event is included in the causal history. This can still be done with vectors, if you keep track of the node in which the last event has been created. For example, when questioning if $x = [2, 0, 0] \rightarrow y = [2, 3, 0]$, with boldface indicating the last event in each vector, you can simply test if $x[0] \leq y[0]$ ($2 \leq 2$) since you have marked the last event in x was created in node A (that is, it corresponds to the first entry of the vector). Since marking numbers in bold is not a practical implementation, however, the last event is usually stored outside the vector (and is sometimes called a *dot*): for example, $[2, 2, 0]$ can be represented as $[2, 1, 0]$ b_2 . Notice that now the vector represents the causal past of b_2 , excluding the event itself.

In an important class of applications there is no need to register causality for all the events in a distributed computation. For example, to modify replicas of data, it often suffices to

Figure 4. Causal histories with only some relevant events.

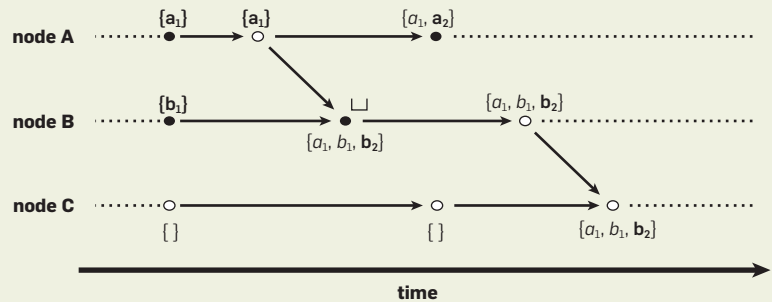
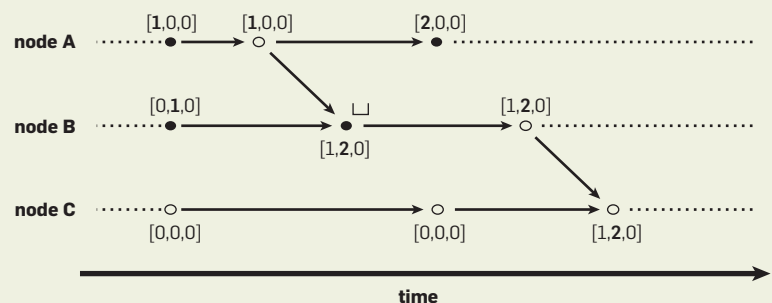


Figure 5. Version vectors with only some relevant events.



register only those events that change replicas. In this case, when thinking about causal histories, you need only to assign a new unique name to these relevant events. Still, you need to propagate the causal histories when messages are propagated from one site to another and the remaining rules for comparing causal histories remain unchanged.

Figure 4 presents the same example as before, but now with events that are not registered for causality tracking denoted with $\circ$. If the run represents the updates to replicas of a data object, then after nodes *A* and *B* are concurrently modified, the state of replica *a* is sent to replica *b* (in a message). When the message is received in node *B*, it is detected two concur-

rent updates have occurred, with histories $\{a_1\}$ and $\{b_1\}$, as neither $a_1 \rightarrow b_1$ nor $b_1 \rightarrow a_1$. In this case, a new version that merges the two updates is created (merge is denoted by the join symbol $\sqcup$), which requires creating a new unique name, leading to $\{a_1, b_1, b_2\}$. When the state of replica *b* is later propagated to replica *c*, as no concurrent update exists in replica *c*, no new version is created.

Again, vectors can compact the representation. The result, known as a *version vector*, was created in 1983,¹² five years before vector clocks. Figure 5 presents the same example as before, represented with version vectors.

In some cases when the state of one replica is propagated to another replica, the two versions are kept by the system as conflicting versions. For example, in Figure 6, when the message from node *A* is received in node *B*, the system keeps each causal history $\{a_1\}$ and $\{b_1\}$ associated with the respective version. The causal history associated with the node containing both versions is $\{a_1, b_1\}$, the union of the causal history of all versions. This approach allows later checking for causality relations between each version and other versions when merging the states of additional nodes. The conflicting versions could also be merged, creating a new unique name, as in the example.

One limitation of causality tracking by vectors is that one entry is needed for each source of concurrency.⁴ You can expect a difference of several orders of magnitude between the number of nodes in a datacenter and the number of clients they handle. Vectors with one entry per client do not scale well when millions of clients are accessing the service.⁷ Again, a look at the foundation of causal histories shows how to overcome this limitation.

The basic requirement in causal histories is each event be assigned a unique identifier. There is no requirement this unique identifier be created locally or immediately. Thus, in systems where nodes can be divided into clients and servers and where clients communicate only with servers, it is possible both to delay the creation of a new unique name until the client communicates with

Figure 6. Causal histories with versions not immediately merged.

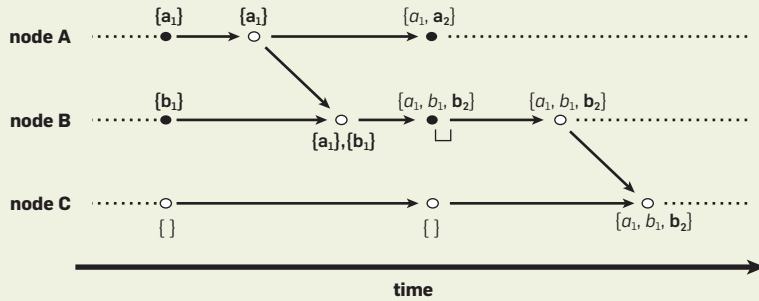


Figure 7. Causal histories in a distributed storage system.

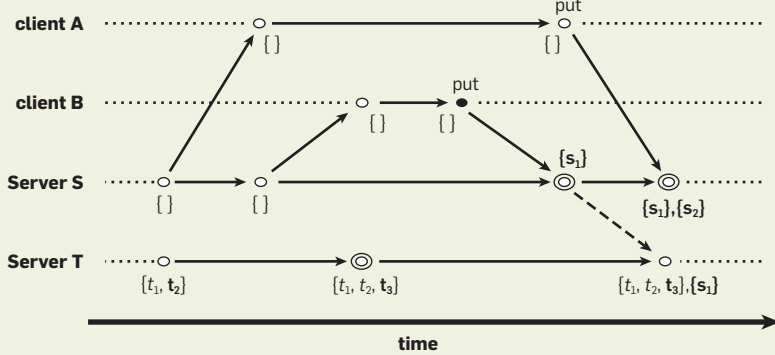
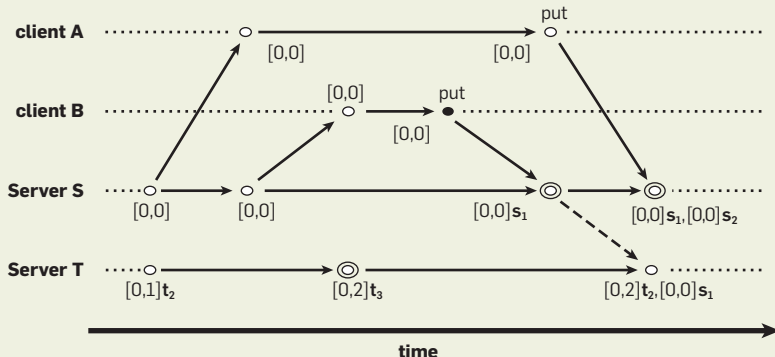


Figure 8. Dotted version vectors in distributed storage system.



the server and to use a unique name generated in the server. The causal history associated with the new version is the union of the causal history of the client and the newly assigned unique name.

Figure 7 shows an example where clients A and B concurrently update server S. When client B first writes its version, a new unique name, s_1 , is created (in the figure this action is denoted by the symbol $\odot$) and merged with the causal history read by the client $\{ \}$, leading to the causal history $\{s_1\}$. When client A later writes its version, the causal history assigned to this version is the causal history at the client, $\{ \}$, merged with the new unique name s_2 , leading to $\{s_2\}$. Using the normal rules for checking for concurrent updates, these two versions are concurrent. In the example, the system keeps both concurrent updates. For simplicity, the interactions of server T with its own clients were omitted, but as shown in the figure, before receiving data from server S, server T had a single version that depicted three updates it managed—causal history $\{t_1, t_2, t_3\}$ —and after that it holds two concurrent versions.

One important observation is that in each node, the union of the causal histories of all versions includes all generated unique names until the last known one: for example, in server S, after both clients send their new versions, all unique names generated in S are known. Thus, the causal past of any update can always be represented using a compact vector representation, as it is the union of all versions known at some server when the client read the object. The combination of the causal past represented as a vector and the last event, kept outside the vector, is known as a *dotted version vector*.^{2,13} Figure 8 shows the previous example using this representation, which, as the system keeps running, eventually becomes much more compact than causal histories.

In the condition expressed before (clients communicate only with servers and a new update overwrites all versions previously read), which is common in key-value stores where multiple clients interact with storage nodes via a *get/put* interface, the dotted version vectors allow causality to

be tracked between the written versions with vectors of the size of the number of servers.

Final Remarks

Tracking causality should not be ignored. It is important in the design of many distributed algorithms. And not respecting causality can lead to strange behaviors for users, as reported by multiple authors.^{1,9}

The mechanisms for tracking causality and the rules used in these mechanisms are often seen as complex,^{6,15} and their presentation is not always intuitive. The most commonly used mechanisms for tracking causality—vector clocks and version vectors—are simply optimized representations of causal histories, which are easy to understand.

By building on the notion of causal histories, you can begin to see the logic behind these mechanisms, to identify how they differ, and even consider possible optimizations. When confronted with an unfamiliar causality-tracking mechanism, or when trying to design a new system that requires it, readers should ask two simple questions: Which events need tracking? How does the mechanism translate back to a simple causal history?

Without a simple mental image for guidance, errors and misconceptions become more common. Sometimes, all you need is the right language.

Acknowledgments

We would like to thank Rodrigo Rodrigues, Marc Shapiro, Russell Brown, Sean Cribbs, and Justin Sheehy for their feedback. This work was partially supported by EU FP7 SyncFree project (609551) and FCT/MCT projects UID/CEC/04516/2013 and UID/EEA/50014/2013. 

Related articles on queue.acm.org

The Inevitability of Reconfigurable Systems
Nick Tredennick, Brion Shimamoto
<http://queue.acm.org/detail.cfm?id=957767>

Abstraction in Hardware System Design
Rishiyur S. Nikhil
<http://queue.acm.org/detail.cfm?id=2020861>

Eventually Consistent: Not What You Were Expecting?
Wojciech Golab, et al.
<http://queue.acm.org/detail.cfm?id=2582994>

References

1. Ajoux, P., Bronson, N., Kumar, S., Lloyd, W., Veeraraghavan, K. Challenges to adopting stronger consistency at scale. In *Proceedings of the 15th Workshop on Hot Topics in Operating Systems*, Kartausa Ittingen, Switzerland. Usenix Association, 2015.
2. Almeida, P.S., Baquero, C., Gonçalves, R., Pregoça, N.M., Fonte, V. Scalable and accurate causality tracking for eventually consistent stores. In *Proceedings of the Distributed Applications and Interoperable Systems*, held as part of the Ninth International Federated Conference on Distributed Computing Techniques (Berlin, Germany, 2014), 67–81.
3. Birman, K.P., Joseph, T.A. Reliable communication in the presence of failures. *ACM Transactions on Computer Systems* 5, 1 (1987), 47–76.
4. Charron-Bost, B. Concerning the size of logical clocks in distributed systems. *Information Processing Letters* 39, 1 (1991), 11–16.
5. Fidge, C.J. Timestamps in message-passing systems that preserve the partial ordering. *Proceedings of the 11th Australian Computer Science Conference* 10, 1 (1988), 56–66.
6. Fink, B. Why vector clocks are easy. Basho Blog, 2010; <http://basho.com/posts/technical/why-vector-clocks-are-easy/>.
7. Hoff, T. How League of Legends scaled chat to 70 million players—it takes lots of minions. High Scalability; <http://highscalability.com/blog/2014/10/13/how-league-of-legends-scaled-chat-to-70-million-players-it-t.html>.
8. Lamport, L. Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM* 21, 7 (1978), 558–565.
9. Lloyd, W., Freedman, M.J., Kaminsky, M., Andersen, D.G. Don't settle for eventual: Scalable causal consistency for wide-area storage with COPS. In *Proceedings of the 23rd ACM Symposium on Operating Systems Principles* (New York, NY, 2011), 401–416.
10. Mattern, F. Virtual time and global states in distributed systems. In *Proceedings of the International Workshop on Parallel and Distributed Algorithms* (Gers, France, 1988), 215–226.
11. Neville-Neil, G. Time is an illusion. *acmqueue* 13, 9 (2015), 57–72.
12. Parker, D.S. et al. Detection of mutual inconsistency in distributed systems. *IEEE Transactions on Software Engineering* 9, 3 (1983), 240–247.
13. Pregoça, N.M., Baquero, C., Almeida, P.S., Fonte, V., Gonçalves, R. Brief announcement: Efficient causality tracking in distributed storage systems with dotted version vectors. In *ACM Symposium on Principles of Distributed Computing*. D. Kowalski and A. Panconesi, Eds. (2012), 335–336.
14. Schwarz, R., Mattern, F. Detecting causal relationships in distributed computations: in search of the Holy Grail. *Distributed Computing* 7, 3 (1994), 149–174.
15. Sheehy, J. Why vector clocks are hard. Basho Blog, 2010; <http://basho.com/posts/technical/why-vector-clocks-are-hard/>.
16. Sheehy, J. There is no now. *acmqueue* 13, 3 (2015), 20–27.

Carlos Baquero (cbm@di.uminho.pt) is assistant professor of computer science and senior researcher at the High-Assurance Software Laboratory, Universidade do Minho and INESC Tec. His research interests are focused on distributed systems, in particular causality tracking, data types for eventual consistency, and distributed data aggregation.

Nuno Pregoça (nuno.pregoça@fct.unl.pt) is associate professor in the Department of Computer Science, Faculty of Science and Technology, Universidade NOVA de Lisboa, and leads the computer systems group at NOVA Laboratory for Computer Science and Informatics. His research interests are focused on the problems of replicated data management and processing of large amounts of information in distributed systems and mobile computing settings.

Copyright held by authors.
Publication rights licensed to ACM. \$15.00

How to lose friends and alienate coworkers.

BY THOMAS A. LIMONCELLI

How SysAdmins Devalue Themselves

Q: DEAR TOM: How can I devalue my work? Lately I've felt like everyone appreciates me, and, in fact, I'm overpaid and underutilized. Could you help me devalue myself at work?

A: Dear Reader: Absolutely! I know what a pain it is to lug home those big paychecks. It's so distracting to have people constantly patting you on the back. Ouch! Plus, popularity leads to dates with famous musicians and movie stars. (Just ask someone like Taylor Swift or Leonardo DiCaprio.) Who wants that kind of distraction when there's a perfectly good video game to be played?

Here are some time-tested techniques that everyone should know.

Work more than 40 hours each week.

This is the simplest, and possibly the most common, technique used by sysadmins. You can easily cut your hourly worth in half by working 80 hours each week. Start by working late one or two nights a week. Then add weekends. Soon you'll be well on your way to a full 80 hours.

Working beyond the hours you are paid is free labor for the company. This reduces your average hourly rate. Why hire more sysadmins when you are willing to take up the slack? Overpaid CEOs say things like, "My pay is commensurate with my responsibilities." Notice they don't relate their pay to how many hours they work. Neither should you.

Mock what you don't like.

If you don't like Microsoft, call it Microsquish. Do not say open source; say "open sores." The more you can sound like a 12-year-old, the better.

If you want to devalue yourself, throw professionalism out the door. Show your disrespect in the most childish way possible. I know one engineer who spells the operating system he doesn't like "Windoze," even in email messages to his co-workers and clients who use it. Way to go!

Interrupt other people.

Nothing says "I don't want to be respected" like not showing respect to other people. That is why it is important not to let people finish their sentences. Once you have heard enough to know the general idea, just start barking out your reply. It shows you don't care about what other people are saying and they will return that lack of respect to you.

Not letting people finish their sentences shows them how smart you are. Your brain is so powerful you have developed ESP. Prove it by answering their questions before they have told you what the problems are.

Respect is a two-way street. It's like a boomerang. You show others your disrespect, and they send it right back at ya.



Don't document or automate your operations.

This point is a little controversial. Some people think they increase their value by refusing to document anything. It makes them unfireable. The truth is employees who keep playbooks and other documentation up to date are highly valued by managers.

Likewise, some sysadmins fear if they write too much automation, it will put them out of a job. The truth is if you automate a task out of existence, there will always be more tasks waiting to be automated. The person who does this is a workforce-multiplier: one person enabling others to do the work of many. That is highly valuable.

Therefore, if you want to devalue yourself, do not document or automate. Assure everyone you will document something later: resist the temptation to update wikis as you perform a task. When asked to automate anything, just look the person in the eye, sigh, and say, "I'm too busy to save time by automating things."

Focus on technology, not business benefits.

That new server you want to buy is awesome, and if the business cannot understand that, yell louder.

Some people disagree. They think every technology purchase should be justified in terms of how it will benefit the business in ways that relate to money or time—for example, a server that will consolidate all sales information, making it possible for salespeople to find the information they need, when they need it. How boring. It is much more

fun to explain it is 20T of SSD-accelerated storage, Intel 5655 CPUs, and triple-redundant power supplies.

If you want to devalue yourself, describe projects in ways that obscure their business value. Use the most detailed technical terms and let people guess the business reason. Act as if the business is there to serve technology, not the other way around.

Only hire people that look like you.

Diversity is about valuing the fact that people with different backgrounds bring different skills to the table. Studies find the addition of a single person with a different background improves a team's productivity.

Productivity? Sounds like the opposite of devaluing yourself. To truly devalue yourself, make sure everyone on your team thinks the same way, has the same skills and similar backgrounds, and makes all the same mistakes.

As I wrote earlier, respect is a two-way street. If you want to devalue yourself, do not value differences.

Be the weird one.

Be "the weird one" in your company. It does not matter that your co-workers do not understand your obscure references to *Dune*, *Animaniacs*, and *LOTR*. The spice must flow so we can make the bologna to put in our slacks before we head to Mount Doom. Pretend you do not notice the confused looks you get. Surely everyone has read *Dune*. Don't explain your cultural references and don't stop making them just because nobody understands them.

Many may consider someone who

is diverse to be weird, but these are two different concepts. Diversity is about valuing differences. Being weird is about being oblivious to other people's reactions. Diversity requires a commitment to educating and being educated. Being weird is the opposite.

Everyone should be free to fly a freak-flag. If you want to devalue yourself, never explain.

Refer to the server room as the Shire. Don't say "happy birthday," say "happy hatchling day." Any time something has a red button, ask if it is candy-like.

Be difficult to find.

You cannot be valuable if you don't exist. If you are difficult to find or are not available when people need you, you aren't providing value to anyone.

Work strange hours. Do not arrive until noon—unless the corporate culture is to arrive at noon; then arrive early.

Either way, make sure your work hours don't overlap well with the people who need you.

Concluding Thoughts

If we all make a concerted effort, then all sysadmins, as a community, can make sure the role of system administrator stays devalued for a very long time. 

Related articles on queue.acm.org

Innovation and Inclusion

Telle Whitney, Elizabeth Ames
<http://dx.doi.org/10.1145/2676861>

Are You Invisible?

Jack Rosenberger
<http://cacm.acm.org/blogs/blog-cacm/94307-are-you-invisible/fulltext>

Automation Should Be Like Iron Man, Not Ultron

Thomas A. Limoncelli
<http://queue.acm.org/detail.cfm?id=2841313>

Thomas A. Limoncelli is a site reliability engineer at Stack Overflow, Inc. in NYC. His books include *The Complete April Fools' Day RFC* (www.rfchumor.com), *The Practice of Cloud Administration* (the-cloud-book.com) and *Time Management for System Administrators* (O'Reilly). He blogs at EverythingSysadmin.com.

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

DOI:10.1145/2818993

Business dashboards that overuse or misuse colors cause cognitive overload for users who then take longer to make decisions.

BY PALASH BERA

How Colors in Business Dashboards Affect Users' Decision Making

BUSINESS DASHBOARDS HELP users visually identify trends, patterns, and anomalies in order to make effective decisions.¹ Dashboards often use a variety of colors to differentiate and identify objects.² Although using colors might improve visualization, overuse or misuse can distract users and adversely affect decision making. This article tests this effect with the help of eye-tracking technology.

The bar charts in Figure 1 reflect sales of office-supply products. The bars in the left-hand chart are uniform in color, and the relative height is the only salient

information source. However, the chart on the right uses a different color for each bar, and the variation in both height and color could be perceived as different information. As a general principle, color variation should reflect value variation.⁹ Use of colors can needlessly attract viewers' attention, causing them to search for meaning that is not there.³

Each dashboard in Figure 2 shows the profits by market size for geographic regions, as well as by products. Although the two dashboards are exactly the same in terms of content, they differ in the way color is used in the bars. The dashboard in the upper panel uses a blue palette that varies from zero saturation (white) to 100% saturation (deep blue), whereas the dashboard in the lower panel uses a palette that starts at 100% saturation (red), decreases to zero saturation in the middle of the scale, and then increases back to 100% saturation but in a green hue.

Contrasting colors attract viewers' attention. If the contrasting colors are not related to a viewer's task, then their use creates distraction; for example, the lower panel in Figure 2 uses two contrasting colors—dark red for less profit, dark green for higher profit. A distraction occurs if the task the viewer is performing does not focus on high or low profit; for example, if the task is to identify what product (such as coffee or tea) has the smallest difference in profit between major and small markets, then the task requires focusing on only the bottom part of the lower panel. However, contrasting colors force the viewer to also look at the contrasting areas, including the top part

» key insights

- Overuse or misuse of colors in business dashboards can distract users and have adverse effects on decision making.
- Studying these effects with eye-tracking technology shows colors do not per se lead to poorer decision performance but rather to longer time to make decisions.
- This research thus suggests dashboard developers avoid the indiscriminate use of colors in business dashboards.



Figure 1. Overuse of colors in bar charts.

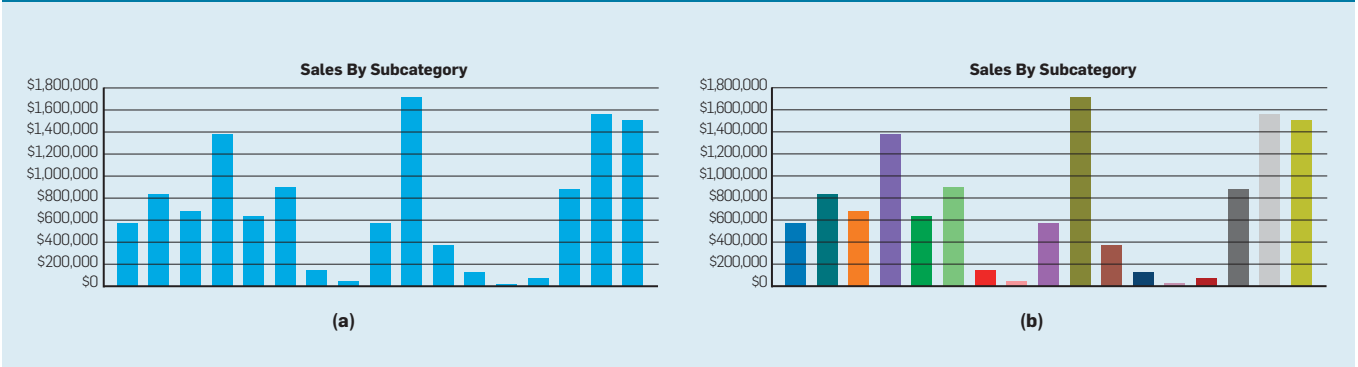
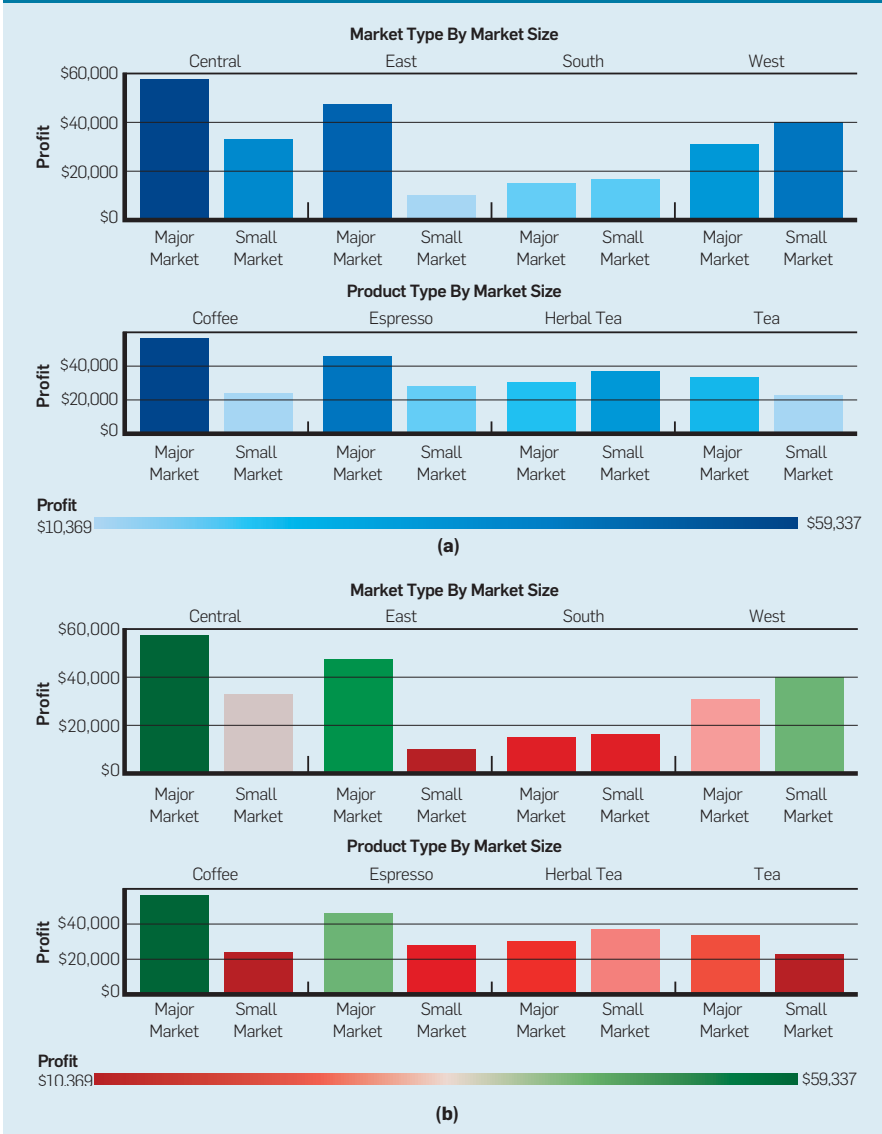


Figure 2. Colors can attract unnecessary attention and viewer distraction.



of the lower panel that includes information on market types (such as East and South); the dashboard in the bottom panel is an example of how colors can be misused.

To perform a decision-making task, viewers need to pay attention to specific

parts of the dashboards. Viewers thus need to isolate and extract the relevant information from a diagram.⁵ A dashboard (related to a task) can be split into two parts—task relevant and task non-relevant.⁵ Using the task example in Figure 2, specific areas of the bottom part of

the lower panel can be termed “task relevant,” and the top part of the lower panel can be termed “task non-relevant.” Misuse of colors forces viewers to look at both areas.

This article investigates how the overuse of colors, as in Figure 1, and misuse of colors, as in Figure 2, in business dashboards affects users’ decision making. It uses eye-tracking technology to provide insight into how individuals read and scan displayed information, identifying how they make decisions with business dashboards.¹³ Eye tracking is particularly relevant in measuring a viewer’s attention and effort on a visual display because it offers a window into how the viewer reads and scans the displayed information.¹³

Eye Tracking

Eye tracking enables researchers to measure a subject’s eye movements while reading text or viewing a picture. The involuntary and voluntary responses of eye movements reflect the internal processing of information.¹³ When reading, our eyes make rapid movements to shift attention from one part of a display to another, then remain almost motionless while the brain interprets the material at that location.¹³ The periods in which the eyes are motionless are called “fixations.”¹⁴ Fixation information can be used to measure the attention individuals pay to the viewing object. Fixation is characterized by three measures:

Fixation count. Total number of fixations on a specific area of display;

Fixation duration. Total fixation time on a specific area of display; and

First fixation time. Start time of the first fixation on the display area.

Empirical Evidence

This study involved dashboards with

bar charts. Bar charts were used because they are the natural choice for displaying multiple measures³ and the most effective way to compare values across dimensions.¹¹ It recruited 30 information systems graduate students from IS analysis and design courses at Texas A&M International University, Laredo, TX, as subjects. These students also took graduate statistics courses and were thus familiar with the elements of dashboards, including graphs and tables. Small samples are typical in eye-tracking studies due to the limited availability of equipment and the large amount of time required to collect each set of observations.⁶

The subjects were asked to answer questions based on two dashboards: What two subcategories of office supplies have the same sales (to test the overuse of colors in Figure 1)? and For which product type is the difference in profit between the major market and the small market the smallest (to test the misuse of colors in Figure 2)?

Hypotheses and Design

Viewers engage in cognitive processes to perform decision-making tasks. Two such types of processes are “incidental processing” and “essential processing.”¹⁰ The former does not require making sense of the presented material, whereas the latter does. Moreover, they can be related to the concepts of “System 1” and “System 2,” the two basic modes of thought in the human mind.⁸ System 1 is the brain’s fast, automatic, and intuitive approach; System 2 is the mind’s slower, analytical mode, where reason dominates.⁸ System 1 operates involuntarily and impulsively with little effort; System 2 allocates effort to the cognitive activities demanding attention.⁸

Viewers of dashboards with overuse or misuse of colors show evidence of System 1 processing. When contrasting colors are used, our brains attempt to assign meaning to the colors.² Viewers are thus directed spontaneously to the areas where the colors are present. These viewers also show use of System 2 processing because this processing is activated when they deliberately pay attention to the decision-making task. In contrast, viewers of dashboards without overuse or misuse of colors avoid System 1 processing and focus on Sys-



A practical implication is dashboard developers should avoid the indiscriminate use of colors in business dashboards.



tem 2 processing instead, requiring less time to perform the task.

Recording eye fixations can reveal the amount of information processed. A longer fixation duration might indicate difficulty extracting information from the displayed area.⁷ A high fixation count and longer fixation duration are thus indicative of cognitive overload. Accordingly, here is the first hypothesis regarding the overuse of colors.

Hypothesis 1. For a dashboard-related task, viewers using dashboards with overuse or misuse of colors have a higher overall fixation number and longer fixation time than viewers with dashboards with no such overuse or misuse.

Tests can be devised to determine whether viewers of dashboards with misuse of colors engage in System 1 processing first before System 2 processing. If there is evidence of this sequence, it will provide insight into the viewers’ decision-making process. Such evidence can be collected by comparing task-relevant and task non-relevant areas of the dashboard with misuse of colors, as in Figure 1. It can be predicted that viewers will engage in System 1 processing because they would be immediately directed to the task non-relevant areas. Subsequently, to complete the task, the viewers must consciously engage in System 2 processing by referring to the task-relevant areas. This sequence of engagement can be identified through the eye metric known as “first fixation time.”

First fixation time is used as a measure of attention to show how quickly one looks at a certain element on a dashboard.⁶ It is measured as the start time of the first fixation on the display area. Eye-tracking software marks a specific area of the dashboard in order to identify the viewer’s eye movements in that area. If a viewer looks at a task non-relevant area at the start of the viewing time (such as the fifth second in a total viewing time of 30 seconds), then the area indeed attracted the viewer’s immediate attention. Low first fixation time thus indicates the area attracted attention quickly, meaning the following hypothesis can be proposed:

Hypothesis 2. Compared to a dashboard that does not misuse colors, viewers of a dashboard that misuses colors will have a low first fixation time on task non-relevant areas and a high first fixa-

tion time on task-relevant areas.

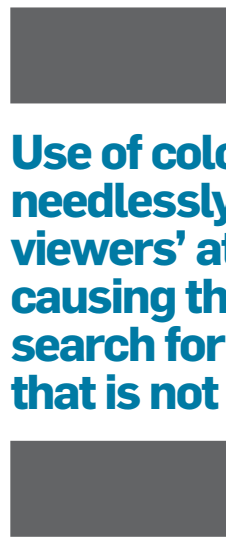
The identification of a System 1, then System 2 activation sequence is easier for dashboards that misuse colors because viewers readily recognize the task-relevant and task non-relevant areas, as in Figure 2. This sequence identification is not possible for dashboards that overuse colors because the areas overlap, as in Figure 1.

This study followed a design in which subjects were randomly assigned to one of the variations—overuse vs. no overuse of colors and misuse vs. no misuse of colors—in dashboards. Each group included an equal number of subjects. One variation was provided to 15 subjects, and the other to the rest. The order of the dashboards was randomized; that is, some subjects received dashboards with or without overuse of colors first and some received dashboards with or without misuse of colors first. The subjects performed two tasks related to the two dashboards as their eye movements were tracked. Prior to tracking, subjects' eyes were calibrated and validated. Following calibration, the subjects were shown a task on a screen and asked to read it carefully. They then saw the dashboard and verbalized an answer. This sequence was used to avoid eye movements associated with writing down answers. The sequence was repeated for each dashboard and eye movements were tracked through EyeLink 1000 software. Verbalizations were also recorded. The tracker recorded a minimum fixation time of four milliseconds.

Analysis of Overuse of Colors

The accuracy of the analysis between the two groups showed no statistical difference. Nearly 92% of the subjects (or 28 of the 30) answered the task correctly in both groups. But total fixation duration and fixation counts for the tasks were compared between the two groups, finding significant differences. Subjects who used the dashboard with overuse of colors took approximately 28 seconds, while those who viewed the dashboard with no such misuse took approximately 22 seconds. The independent sample t test (see Table 1) confirmed the differences in fixation durations; counts were significant in the two groups.

Fixation counts and durations showed the dashboard with overuse of colors induced more cognitive effort



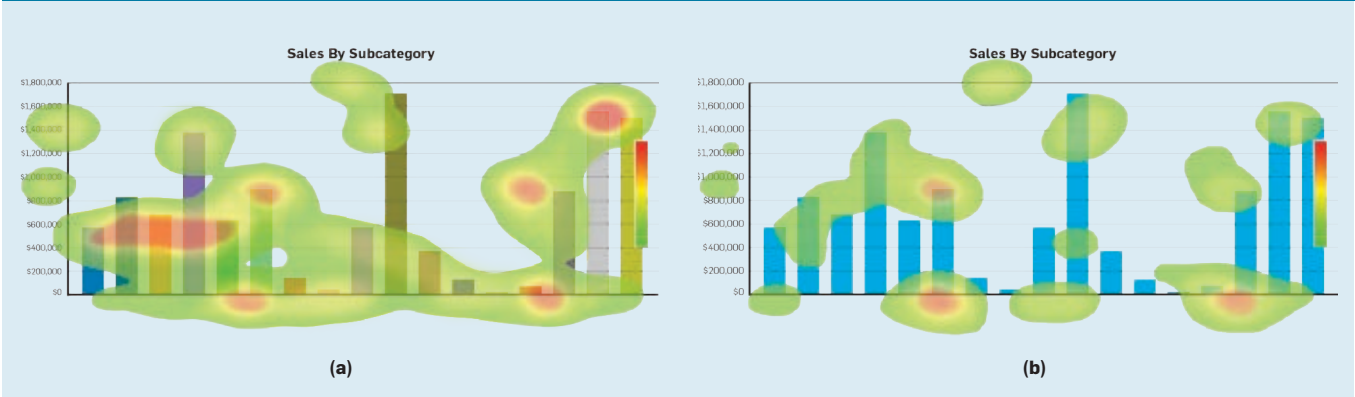
Use of colors can needlessly attract viewers' attention, causing them to search for meaning that is not there.

compared to the dashboard with no overuse of colors. However, this excess effort did not affect performance of the task. The sample heat map of two subjects (see Figure 3) reflects the presence of cognitive overload. A heat map uses different colors—red for the largest number of fixations, green for the fewest number of fixations—to show the number of fixations viewers make in certain areas of an image.⁶ The heat map on the left of the figure indicates the subject spent significant time on all bars with different colors. In contrast, the heat map on the right of the figure shows how another subject spent time on specific bars.

Analysis of Misuse of Colors

Task performance between the two groups showed no statistical difference. Approximately 88% of the subjects (or 26 of the 30) answered the task correctly in both groups. However, a significant difference was found in the overall fixation durations and counts between the two groups. To perform the task, subjects who viewed the dashboard with misuse of colors took approximately 45 seconds, whereas those who viewed the dashboard with no such misuse took approximately 27 seconds; Table 2 shows the results of the independent sample t test, indicating a high cognitive load exists for viewers of the dashboard with misuse of colors.

To determine whether there was a System 1, then System 2 sequence, the study identified task-relevant and task-non-relevant areas. First, a specific task non-relevant area was identified from the dashboard that did not have to be viewed to perform the task. This area is the bar that indicates the small market in the East zone (dark red) for the chart “market type by market size” that appears in the top part of the lower panel of Figure 2. The time of the first fixation was obtained when a viewer would look at this dark red bar. This time was compared with the first fixation time of the same area (light blue) in the top part of the top panel in Figure 2. On average, the subjects using the dashboard with misuse of colors looked at this area within 6.2 seconds of their average viewing time of 45.2 seconds. On the other hand, subjects using the dashboard with no such misuse looked at this specific area within 18.2 seconds of their average viewing time of 26.8 sec-

Figure 3. Heat maps of a subject performing a task with overuse vs. no overuse of colors in dashboards.**Figure 4. Heat maps of two subjects performing tasks with dashboards.**

onds. Three of the subjects in the latter group did not look at this area at all.

The first fixation times of the two groups were compared for the task-relevant areas in the dashboard. The bar chart labeled “Product Type By Market Size” was chosen as a task-relevant area because viewers needed to see this area to complete the task. The results (see Table 3) show viewers using the dashboard with no misuse of colors viewed this area much more quickly than the other group, indicating the contrasted areas indeed distracted the viewers who engaged in System 1 and System 2 processing, respectively.

The sample heat maps of two subjects (see Figure 4) reflect the analysis mentioned in Table 2 and Table 3. The bottom-left heat map in Figure 4 shows areas with contrasting colors attracted viewer’s attention, whereas the bottom-right heat map shows the focus was on the task-relevant areas (such as the chart title).

A fixation-sequence analysis was conducted to determine whether a viewer’s decision-making process involving the dashboard with misuse of colors was different from those viewing the dashboard with no such misuse. To do it, the dash-

Table 1. Analysis of fixation durations and counts for dashboards with overuse of colors.

Group	Fixation duration (ms)	t-stat	p-value	Fixation count	t-stat	p-value
Dashboard with overuse of colors	28313.71	2.12	0.02*	107.78	2.47	0.01*
Dashboard with no overuse of colors	22586.44			78.25		

* significant at 0.05 level; ms = millisecond

Table 2. Analysis of fixation durations and counts for dashboards with misuse of colors.

Group	Fixation duration (ms)	t-stat	p-value	Fixation count	t-stat	p-value
Dashboard with misuse of colors	45176.69	2.37	0.01*	159.43	1.92	0.03*
Dashboard with no misuse of colors	26800.00			102.57		

* significant at 0.05 level; ms = millisecond

Table 3. Analysis of the first fixation times for dashboards that misuse colors; similar results were obtained for the other task-non-relevant and -relevant areas between the two groups.

Group	First fixation time task non-relevant area (ms)	t-stat	p-value	First fixation time-task-relevant area (ms)	t-stat	p-value
Dashboard with misuse of colors	6205.31	7.50	0.00*	13065.63	6.76	0.00*
Dashboard with no misuse of colors	18127.09			7901.45		

* significant at 0.05 level; ms = millisecond

Figure 5. First fixation-time-sequence analysis.

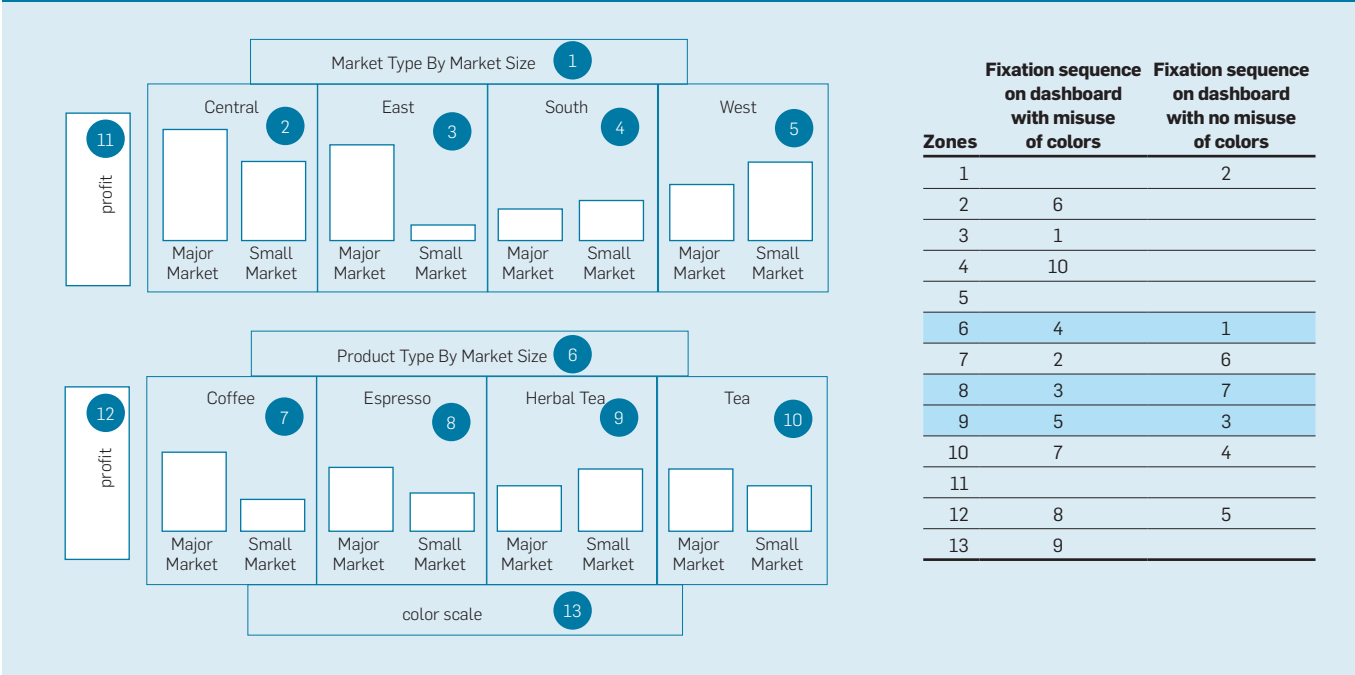
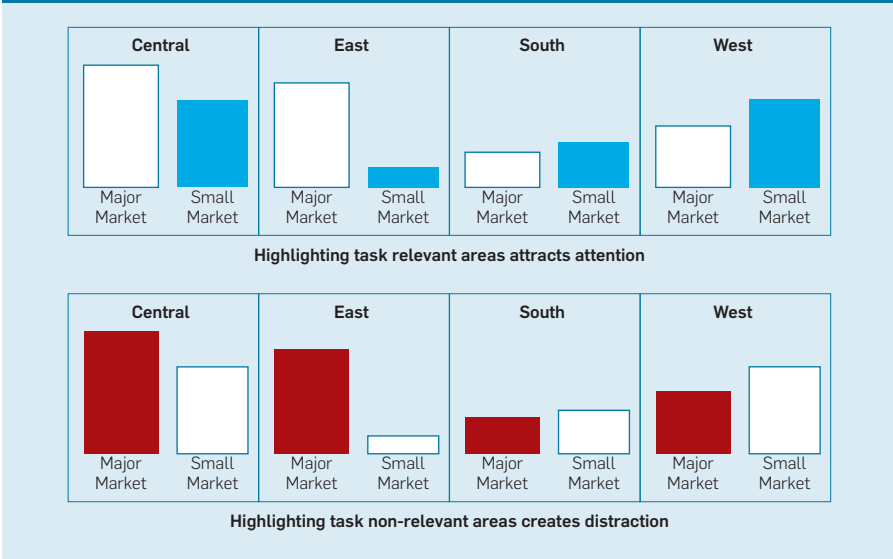


Table 4. Summary of eye-tracking study.

Dashboard overusing colors	Dashboard not overusing colors
Viewers spend time inspecting all the bars and trying to interpret the additional meaning of the colors in each bar, a sequence that takes extra time and effort.	Viewers focus on the bar chart, performing the task quickly.
Dashboard misusing colors	Dashboard not misusing colors
Viewers focus on the areas with contrasting colors.	Viewers focus on the areas related to the task.
The task is not related to the contrasting colors so distracts viewers.	In the absence of distraction, viewers focus on the dashboard, performing the task quickly.
Task performance is not affected, but viewers spend extra time and effort due to the distraction.	

Figure 6. Effect of highlighting task-relevant and non-relevant areas.



board with misuse of colors, as in Figure 2, was divided into 13 zones (see Figure 5), and the eye-fixation sequences of all subjects were mapped with these zones. Zone 9 was the most relevant because it contained the answer to the task. The mapping results were ranked in the order in which the zones were visited first by the subjects, as in Figure 5. The task-relevant areas are highlighted in the table within Figure 5, indicating the viewers of the dashboard with misuse of colors visited the task non-relevant areas (such as Zone 3 and Zone 7) first, followed by the task-relevant areas (such as Zone 9). This sequence demonstrates viewers engaged in System 1 processing first, then System 2 processing. In contrast, viewers of the dashboard with no such misuse visited the task-relevant areas (such as Zone 6) first. The viewers thus engaged System 2 processing directly. Together with the heat maps and the statistical analysis in Table 3, this analysis provides evidence the misuse of colors affects the pattern of a viewers' eye movements and decision-making processes.

Conclusion

This article has reported the effects of overuse and misuse of colors in dashboards on decision making, as summarized in Table 4.

The study made several interesting observations. First, the high fixation counts and durations in Table 1 and

Table 2 indicate overuse and misuse of colors in dashboards create distractions and thus viewers' cognitive overload. Second, the areas affected by overuse and misuse of colors attract viewers' attention and delay performance of a task. Although such distraction increases a viewer's cognitive load, that increase is not great enough to affect task performance. It can be argued viewers engaged in System 2 processing, ensuring task performance is not affected. Third, use of colors affects the decision-making process when using dashboards. The first fixation times (in Table 3) and the fixation sequence analysis (in Figure 5) indicate color variations in dashboards affect viewers' decision-making processes. Finally, the decision performance is not negatively affected in all groups (see the cells in Table 4).

Specific suggestions can thus be made to dashboard developers concerning use of colors in business dashboards. Although cognitive overload does not necessarily affect a decision maker's performance, overload is undesirable. A practical implication is dashboard developers should avoid the indiscriminate use of colors in business dashboards. Using the concepts of task-relevant and task non-relevant areas,⁵ they need to think in advance about how a dashboard will be used. They should first identify the task-relevant and task non-relevant areas of the dashboard for possible decision-making tasks. Note these areas could change based on tasks users intend to perform with the dashboards. Following such identification, dashboard developers should avoid highlighting task non-relevant areas, as doing so causes distraction. Instead, the task-relevant areas should be highlighted to attract viewers' attention. Figure 6 reflects the effect of highlighting task-relevant (blue) and task non-relevant (brown) areas. If a task relates to decision making with small markets, then areas related to small markets are task relevant. This example shows highlighting specific areas of visualization can cause distraction.

This research shows dashboards with misuse and overuse of colors do not lead to poorer decision performance but rather decision makers using such dashboards taking longer to make a decision. One notable practical finding is organi-



One notable practical finding is organizations do not need to redevelop their dashboards unless the cost of redevelopment is less than the cost of the extra decision time.



zations do not need to redevelop their dashboards unless the cost of redevelopment is less than the cost of the extra decision time. It is likely existing dashboards do not need to be altered, though new dashboard development should avoid overuse and misuse of colors.

These results also apply to the use of colors in dashboards. Bar charts are used more frequently in dashboards than in any other aspect of information visualization.¹² Dashboards are designed for users to see how various indicators are performing¹⁵ and are thus used primarily to identify trends and patterns for decision making.³ Here, bar charts within dashboards were used to identify patterns. Future studies can investigate the effect of colors on new generations of complex dashboards (such as those providing interactivity through a drill-down feature) and on ways to measure task performance (such as memory retention). 

References

1. Brath, R. and Peters, M. Dashboard design: Why design is important. *DM Direct Newsletter* (Oct. 2004).
2. Few, S. Dashboard confusion. *InformationWeek* (Mar. 2004).
3. Few, S. *Information Dashboard Design: Displaying Data for At-A-Glance Monitoring*. Analytics Press, Burlingame, CA, 2012.
4. Goldstein, E.B. *Sensation and Perception*. Thomson Wadsworth, 2007.
5. Hegarty, M., Canham, M., and Fabrikant, S. Thinking about the weather: How display salience and knowledge affect performance in a graphic inference task. *Journal of Experimental Psychology* 36, 1 (2010), 37–53.
6. Jacob, R.J.K. and Karn, K.S. Eye tracking in human-computer interaction and usability research: Ready to deliver the promises. Chapter 4 in *The Mind's Eye: Cognitive and Applied Aspects of Eye Movement*, R. Radach, J. Hyona, and H. Deubel, Eds. Elsevier Sciences, Oxford, U.K., 2003, 573–605.
7. Just, M.A. and Carpenter, P.A. Eye fixations and cognitive processes. *Cognitive Psychology* 8, 1 (1976), 441–480.
8. Kahneman, D. *Thinking, Fast and Slow*. Farrar, Straus and Giroux, New York, 2011.
9. Kosslyn, S.M. *Graph Design for the Eye and Mind*. Oxford University Press, 2006.
10. Mayer, R.E. and Moreno, R. Nine ways to reduce cognitive load in multimedia learning. *Educational Psychologist* 38, 1 (2003), 43–52.
11. Murray, D. *Tableau Your Data!: Fast and Easy Visual Analysis with Tableau Software*. John Wiley and Sons, Inc., New York, 2013.
12. Peck, G. *Tableau 8: The Official Guide*. McGraw Hill Education, New York, 2014.
13. Rayner, K. Eye movements in reading and information processing: 20 years of research. *Psychological Bulletin* 124, 3 (1998), 372–422.
14. Sharif, B. and Maletic, J. An eye-tracking study on the effects of layout in understanding the role of design patterns. In *Proceedings of the IEEE International Conference on Software Maintenance* (Timisoara, Romania, Sept. 12–18). IEEE Press, 2010, 41–48.
15. Yigitbasoglu, O. and Velcu, O. A review of dashboards in performance management: Implications for design and research. *International Journal of Accounting Information Systems* 13, 1 (2012), 41–59.

Palash Bera (pbera@slu.edu) is an assistant professor in the John Cook School of Business at Saint Louis University, Saint Louis, MO.

© 2016 ACM 0001-0782/16/04 \$15.00

DOI:10.1145/2818990

Fusing information from multiple biometric traits enhances authentication in mobile devices.

BY MIKHAIL I. GOFMAN AND SINJINI MITRA

Multimodal Biometrics for Enhanced Mobile Device Security

MILLIONS OF MOBILE devices are stolen every year, along with associated credit card numbers, passwords, and other secure and personal information stored therein. Over the years, criminals have learned to crack passwords and fabricate biometric traits and have conquered practically every kind of user-authentication mechanism designed to stop them from accessing device data. Stronger mobile authentication mechanisms are clearly needed.

Here, we show how multimodal biometrics promises untapped potential for protecting consumer mobile devices from unauthorized access, an authentication approach based on multiple physical and behavioral traits like face and voice. Although multimodal biometrics are deployed in homeland

security, military, and law-enforcement applications,^{15,18} they are not yet widely integrated into consumer mobile devices. This can be attributed to implementation challenges and con-

» key insights

- Multimodal biometrics, or identifying people based on multiple physical and behavioral traits, is the next logical step toward more secure and robust biometrics-based authentication in mobile devices.
- The face-and-voice-based biometric system covered here, as implemented on a Samsung Galaxy S5 phone, achieves greater authentication accuracy in uncontrolled conditions, even with poorly lit face images and voice samples, than single-modality face and voice systems.
- Multimodal biometrics on mobile devices can be made user friendly for everyday consumers.





cern that consumers may find the approach inconvenient.

We also show multimodal biometrics can be integrated with mobile devices in a user-friendly manner and significantly improve their security. In 2015, we thus implemented a multimodal biometric system called Proteus at California State University, Fullerton, based on face and voice on an Samsung Galaxy S5 phone, integrating new multimodal biometric authentication algorithms optimized for consumer-level mobile devices and an interface that allows users to readily record multiple biometric traits. Our experiments confirm it achieves considerably greater authentication accuracy than systems based solely on face or voice alone. The next step is to integrate other biometrics

(such as fingerprints and iris scans) into the system. We hope our experience encourages researchers and mobile-device manufacturers to pursue the same line of innovation.

Biometrics

Biometrics-based authentication establishes identity based on physical and behavioral characteristics (such as face and voice), relieving users from having to create and remember secure passwords. At the same time, it challenges attackers to fabricate human traits that, though possible, is difficult in practice.²¹ These advantages continue to spur adoption of biometrics-based authentication in smartphones and tablet computers.

Despite the arguable success of biometric authentication in mobile devices,

several critical issues remain, including, for example, techniques for defeating iPhone TouchID and Samsung Galaxy S5 fingerprint recognition systems.^{2,26} Further, consumers continue to complain that modern mobile biometric systems lack robustness and often fail to recognize authorized users.⁴ To see how multimodal biometrics can help address these issues, we first examine their underlying causes.

The Mobile World

One major problem of biometric authentication in mobile devices is sample quality. A good-quality biometric sample—whether a photograph of a face, a voice recording, or a fingerprint scan—is critical for accurate identification; for example, a low-resolution photograph of a face or

noisy voice recording can lead a biometric algorithm to incorrectly identify an impostor as a legitimate user, or “false acceptance.” Likewise, it can cause the algorithm to declare a legitimate user an impostor, or “false rejection.” Capturing high-quality samples in mobile devices is especially difficult for two main reasons. Mobile users capture biometric samples in a variety of environmental conditions; factors influencing these conditions include insufficient lighting, different poses, varying camera angles, and background noise. And biometric sensors in consumer mobile devices often trade sample quality for portability and lower cost; for example, the dimensions of an Apple iPhone’s TouchID fingerprint scanner prohibit it from capturing the entire finger, making it easier to circumvent.⁴

Another challenge is training the biometric system to recognize the device user. The training process is based on extracting discriminative features from a set of user-supplied biometric samples. Increasing the number and variability of training samples increases identification accuracy. In practice, however, most consumers likely train their systems with few samples of limited variability for reasons of convenience. Multimodal biometrics is the key to addressing these challenges.

Promise of Multimodal Biometrics

Due to the presence of multiple pieces of highly independent identifying information (such as face and voice), multimodal systems can address the

security and robustness challenges confronting today’s mobile unimodal systems^{13,18} that identify people based on a single biometric characteristic. Moreover, deploying multimodal biometrics on existing mobile devices is practical; many of them already support face, voice, and fingerprint recognition. What is needed is a robust user-friendly approach for consolidating these technologies. Multimodal biometrics in consumer mobile devices deliver multiple benefits.

Increased mobile security. Attackers can defeat unimodal biometric systems by spoofing a single biometric modality used by the system. Establishing identity based on multiple modalities challenges attackers to simultaneously spoof multiple independent human traits—a significantly tougher challenge.²¹

More robust mobile authentication. When using multiple biometrics, one biometric modality can be used to compensate for variations and quality deficiencies in the others; for example, Proteus assesses face-image and voice-recording quality and lets the highest-quality sample have greater impact on the identification decision.

Likewise, multimodal biometrics can simplify the device-training process. Rather than provide many training samples from one modality (as they often must do in unimodal systems), users can provide fewer samples from multiple modalities. This identifying information can be consolidated to ensure sufficient training data for reliable identification.

A market ripe with opportunities. De-

spite the recent popularity of biometric authentication in consumer mobile devices, multimodal biometrics have had limited penetration in the mobile consumer market.^{1,15} This can be attributed to the concern users could find it inconvenient to record multiple biometrics. Multimodal systems can also be more difficult to design and implement than unimodal systems.

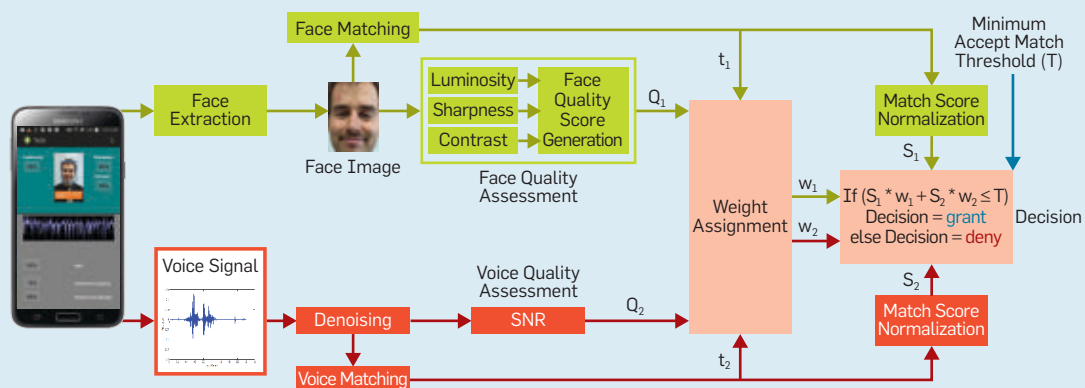
However, as we explain, these problems are solvable. Companies like Apple and Samsung have invested significantly in integrating biometric sensors (such as cameras and fingerprint readers) into their products. They can thus deploy multimodal biometrics without substantially increasing their production costs. In return, they profit from enhanced device sales due to increased security and robustness. In the following sections we discuss how to achieve such profitable security.

Fusing Face and Voice Biometrics

To illustrate the benefits of multimodal biometrics in consumer mobile devices, we implemented Proteus based on face and voice biometrics, choosing these modalities because most mobile devices have cameras and microphones needed for capturing them. Here, we provide an overview of face- and voice-recognition techniques, followed by an exploration of the approaches we used to reconcile them.

Face and voice recognition. We used the face-recognition technique known as FisherFaces³ in Proteus, as it works well in situations where images are captured under varying conditions, as

Figure 1. Schematic diagram illustrating the Proteus quality-based score-level fusion scheme.



expected in the case of face images obtained through mobile devices. FisherFaces uses pixel intensities in the face images as identifying features. In the future, we plan to explore other face-recognition techniques, including Gabor wavelets⁶ and Histogram Oriented Gradients (HOG).⁵

We used two approaches for voice recognition: Hidden Markov Models (HMM) based on the Mel-Frequency Cepstral Coefficients (MFCCs) as voice features,¹⁰ the basis of our score-level fusion scheme; and Linear Discriminant Analysis (LDA),¹⁴ the basis for our feature-level fusion scheme. Both approaches recognize a user's voice independent of phrases spoken.

Assessing face and voice sample quality. Assessing biometric sample quality is important for ensuring the accuracy of any biometric-based authentication system, particularly for mobile devices, as discussed earlier. Proteus thus assesses facial image quality based on luminosity, sharpness, and contrast, while voice-recording quality is based on signal-to-noise ratio (SNR). These classic quality metrics are well documented in the biometrics research literature.^{1,17,24} We plan to explore other promising metrics, including face orientation, in the future.

Proteus computes the average luminosity, sharpness, and contrast of a face image based on the intensity of the constituent pixels using approaches described in Nasrolli and Moeslund.¹⁷ It then normalizes each quality measure using the min-max normalization method to lie between [0, 1], finally computing their average to obtain a single quality score for a face image. One interesting problem here is determining the impact each quality metric has on the final face-quality score; for example, if the face image is too dark, then poor luminosity would have the greatest impact, as the absence of light would be the most significant impediment to recognition. Likewise, in a well-lit image distorted due to motion blur, sharpness would have the greatest impact.

SNR is defined as a ratio of voice signal level to the level of background noise signals. To obtain a voice-quality score, Proteus adapts the probabilistic approach described in Vondrasek and Pollak²⁵ to estimate the voice and noise



To get its algorithm to scale to the constrained resources of the device, Proteus had to be able to shrink the size of face images to prevent the algorithm from exhausting the available device memory.



signals, then normalizes the SNR value to the [0, 1] range using min-max normalization.

Multimodal biometric fusion. In multimodal biometric systems, information from different modalities can be consolidated, or fused, at the following levels:²¹

Feature. Either the data or the feature sets originating from multiple sensors and/or sources are fused;

Match score. The match scores generated from multiple trait-matching algorithms pertaining to the different biometric modalities are combined, and

Decision. The final decisions of multiple matching algorithms are consolidated into a single decision through techniques like majority voting.

Biometric researchers believe integrating information at earlier stages of processing (such as at the feature level) is more effective than having integration take place at a later stage (such as at the score level).²⁰

Multimodal Mobile Biometrics Framework

Proteus fuses face and voice biometrics at either score or feature level. Since decision-level fusion typically produces only limited improvement,²¹ we did not pursue it when developing Proteus.

Proteus does its training and testing processes with videos of people holding a phone camera in front of their faces while speaking a certain phrase. From each video, the face is detected through the Viola-Jones algorithm²⁴ and the system extracts the soundtrack. The system de-noises all sound frames to remove frequencies outside human voice range (85Hz–255Hz) and drops frames without voice activity. It then uses the results as inputs into our fusion schemes.

Score-level fusion scheme. Figure 1 outlines our score-level fusion approach, integrating face and voice biometrics. The contribution of each modality's match score toward the final decision concerning a user's authenticity is determined by the respective sample quality. Proteus works as outlined in the following paragraphs.

Let t_1 and t_2 , respectively, denote the average face- and voice-quality scores of the training samples from the user of the device. Next, from a

test-video sequence, Proteus computes the quality scores Q_1 and Q_2 of the two biometrics, respectively. These four parameters are then passed to the system's weight-assignment module, which computes weights w_1 and w_2 for face and voice modalities, respectively. Each w_i is calculated as $w_i = \frac{v_i}{p_1 + p_2}$, where p_1 and p_2 are percent proximities of Q_1 to t_1 and Q_2 to t_2 , respectively. The system requests users train mostly through good-quality samples, as discussed later, so close proximity of the testing sample quality to that of training samples is a sign of a good-quality testing image. Greater weight is thus assigned to the modality with a higher-quality sample, ensuring effective integration of quality in the system's final authentication process.

The system then computes and normalizes matching scores S_1 and S_2 from the respective face- and voice-recognition algorithms applied to test images through z-score normalization. We chose this particular method because it is a commonly used normalization method, easy to implement, and highly efficient.¹¹ However, we wish to experiment with more robust methods (such as the $\tanh$ and sigmoid functions) in the future. The system then computes the overall match score for the fusion scheme using the weighted sum rule as $M = S_1w_1 + S_2w_2$. If $M \geq T$ (T is the pre-selected threshold), the system will accept the user as authentic; otherwise, it declares the user to be an imposter.

Discussion. The scheme's effectiveness is expected to be greatest when $t_1 = Q_1$ and $t_2 = Q_2$. However, the system must exercise caution here to ensure significant representation of both modalities in the fusion process; for example, if Q_2 differs greatly from t_2 while Q_1 is close to t_1 , the authentication process is dominated by the face modality, thus reducing the process to an almost unimodal scheme based on the face biometric. A mandated benchmark is thus required for each quality score to ensure the fusion-based authentication procedure does not grant access for a user if the benchmark for each score is not met. Without such benchmarks, the whole authentication procedure could be exposed to the risk of potential fraud-



Storing and processing biometric data on the mobile device itself, rather than offloading these tasks to a remote server, eliminates the challenges of securely transmitting the biometric data and authentication decisions across potentially insecure networks.



ulent activity, including deliberate attempts to alter the quality score of a specific biometric modality. The system must thus ensure the weight of each modality does not fall below a certain threshold so the multimodal scheme remains viable.

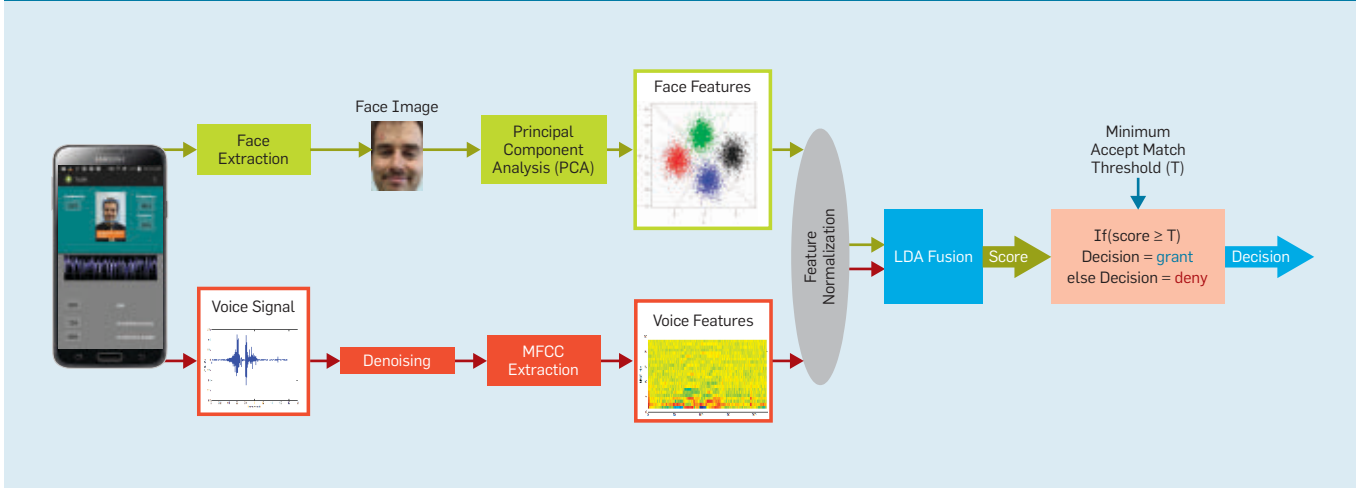
In 2014, researchers at IBM proposed a score-level fusion scheme based on face, voice, and signature biometrics for iPhones and iPads.¹ Their implementation considered only the quality of voice recordings, not face images, and is distinctly different from our approach, which incorporates the quality of both modalities. Further, because their goal was secure sign-in into a remote server, they outsourced the majority of computational tasks to the target server; Proteus performs all computations directly on the mobile device itself. To get its algorithm to scale to the constrained resources of the device, Proteus had to be able to shrink the size of face images to prevent the algorithm from exhausting the available device memory. Finally, Aronowitz et al.¹ used multiple facial features (such as HOG and LBP) that, though arguably more robust than FisherFaces, can be prohibitively slow when executed locally on a mobile device; we plan to investigate using multiple facial features in the future.

Feature-level fusion scheme.

Most multimodal feature-level fusion schemes assume the modalities to be fused are compatible (such as in Kisku et al.¹² and in Ross and Govindarajan²⁰); that is, the features of the modalities are computed in a similar fashion, based on, say, distance. Fusing face and voice modalities at the feature level is challenging because these two biometrics are incompatible: face features are pixel intensities and voice features are MFCCs. Another challenge for feature-level fusion is the curse of dimensionality arising when the fused feature vectors become excessively large. We addressed both challenges through the LDA approach. In addition, we observed LDA required less training data than neural networks and HMMs, with which we have experimented.

The process (see Figure 2) works like this:

Figure 2. Linear discriminant analysis-based feature-level fusion.



Phase 1 (face feature extraction). The Proteus algorithm applies Principal Component Analysis (PCA) to the face feature set to perform feature selection;

Phase 2 (voice feature extraction). It extracts a set of MFCCs from each preprocessed audio frame and represents them in a matrix form where each row is used for each frame and each column for each MFCC index. And to reduce the dimensionality of the MFCC matrix, it uses the column means of the matrix as its voice feature vector;

Phase 3 (fusion of face and voice features). Since the algorithm measures face and voice features using different units, it standardizes them individually through the z-score normalization method, as in score-level fusion. The algorithm then concatenates these normalized features to form one big feature vector. If there are N face features and M voice features, it will have a total of $N + M$ features in the concatenated, or fused, set. The algorithm then uses LDA to perform feature selection from the fused feature set. This helps address the curse of the dimensionality problem by removing irrelevant features from the combined set; and

Phase 4 (authentication). The algorithm uses Euclidean distance to determine the degree of similarity between the fused features sets from the training data and each test sample. If the distance value is less than or equal to a predetermined threshold, it accepts the test subject as a legitimate user. Otherwise, the subject is declared an impostor.

Implementation

We implemented our quality-based score-level and feature-level fusion approaches on a randomly selected Samsung Galaxy S5 phone. User friendliness and execution speed were our guiding principles.

User interface. Our first priority when designing the interface was to ensure users could seamlessly capture face and voice biometrics simultaneously. We thus adopted a solution that asks users to record a short video of their faces while speaking a simple phrase. The prototype of our graphical user interface (GUI) (see Figure 3) gives users real-time feedback on the quality metrics of their face and voice, guiding them to capture the best-quality samples possible; for example, if the luminosity in the video differs significantly from the average luminosity of images in the training database, the user may get a prompt saying, Suggestion: Increase lighting. In addition to being user friendly, the video also facilitates integration of other security features (such as liveness checking⁷) and correlation of lip movement with speech.⁸

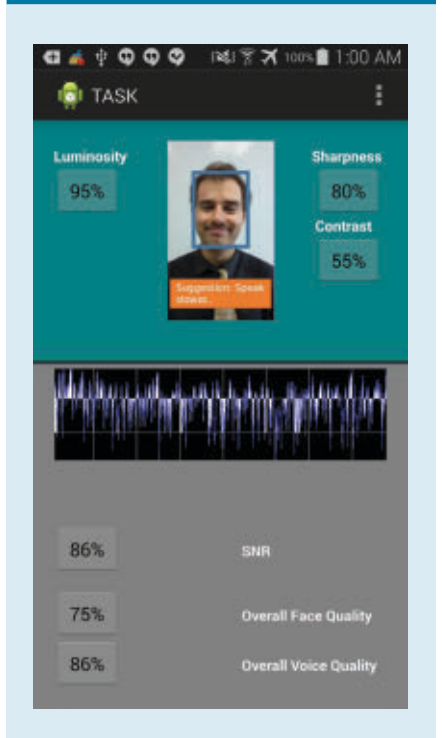
To ensure fast authentication, the Proteus face- and voice-feature extraction algorithms are executed in parallel on different processor cores; the Galaxy S5 has four cores. Proteus also uses similar parallel programming techniques to help ensure the GUI's responsiveness.

Security of biometric data. The greatest risk from storing biometric data on a mobile device (Proteus stores data from multiple biometrics) is the possibility of attackers stealing

and using it to impersonate a legitimate user. It is thus imperative that Proteus stores and processes the biometric data securely.

The current implementation stores only MFCCs and PCA coefficients in the device persistent memory, not raw biometric data, from which deriving useful biometric data is nontrivial.¹⁶ Proteus can enhance security significantly by using cancelable biometric templates¹⁹ and encrypting, storing, and processing biometric data in Trusted Execution Environment tamper-proof hardware highly isolated from the rest of

Figure 3. The GUI used to interact with Proteus.



the device software and hardware; the Galaxy S5 uses this approach to protect fingerprint data.²²

Storing and processing biometric data on the mobile device itself, rather than offloading these tasks to a remote server, eliminates the challenge of securely transmitting the biometric data and authentication decisions across potentially insecure networks. In addition, this approach alleviates consumers' concern regarding the security, privacy, and misuse of their biometric data in transit to and on remote systems.

Performance Evaluation

We compared Proteus recognition accuracy to unimodal systems based on face and voice biometrics. We measured that accuracy using the standard equal error rate (EER) metric, or the value where the false acceptance rate (FAR) and the false rejection rate (FRR) are equal. Mechanisms enabling secure storage and processing of biometric data must therefore be in place.

Database. For our experiments, we created a CSUF-SG5 homegrown multimodal database of face and voice samples collected from University of California, Fullerton, students, employees, and individuals from outside the university using the Galaxy S5 (hence the name). To incorporate various types and levels of variations and distortions in the samples, we collected them in a variety of real-world settings. Given such a diverse database of multimodal biometrics is unavailable, we

plan to make our own one publicly available. The database today includes video recordings of 54 people of different genders and ethnicities holding a phone camera in front of their faces while speaking a certain simple phrase.

The faces in these videos show the following types of variations:

Four expressions. Neutral, happy, sad, angry, and scared;

Three poses. Frontal and sideways (left and right); and

Two illumination conditions. Uniform and partial shadows.

The voice samples show different levels of background noise, from car traffic to music to people chatter, coupled with distortions in the voice itself (such as raspiness). We used 20 different popular phrases, including "Roses are red," "Football," and "13."

Results. In our experiments, we trained the Proteus face, voice, and fusion algorithms using videos from half of the subjects in our database (27 subjects out of a total of 54), while we considered all subjects for testing. We collected most of the training videos in controlled conditions with good lighting and low background noise levels and with the camera held directly in front of the subject's face. For these subjects, we also added a few face and voice samples from videos of less-than-ideal quality (to simulate the limited variation of training samples a typical consumer would be expected to provide) to increase the algorithm's chances of correctly identifying the user in similar conditions. Overall, we used three face frames and five

voice recordings per subject (extracted from video) as training samples. We performed the testing through a randomly selected face-and-voice sample from a subject we selected randomly from among the 54 subjects in the database, leaving out the training samples. Overall, our subjects created and used 480 training and test-set combinations, and we averaged their EERs and testing times. We undertook this statistical cross-validation approach to assess and validate the effectiveness of our proposed approaches based on the available database of 54 potential subjects.

Quality-based score-level fusion. Table 1 lists the average EERs and testing times from the unimodal and multimodal schemes. We explain the high EER of our HMM voice-recognition algorithm by the complex noise signals in many of our samples, including traffic, people chatter, and music, that were difficult to detect and eliminate. Our quality-score-level fusion scheme detected low SNR levels and compensated by adjusting weights in favor of the face images that were of substantially better quality. By adjusting weights in favor of face images, the face biometric thus had a greater impact on the final decision of whether or not a user is legitimate than the voice biometric.

For the contrasting scenario, where voice samples were relatively better quality than face samples, as in Table 1, the EERs were 21.25% and 20.83% for unimodal voice and score-level fusion, respectively.

These results are promising, as they show the quality of the different modalities can vary depending on the circumstances in which mobile users might find themselves. They also show Proteus adapts to different conditions by scaling the quality weights appropriately. With further refinements (such as more robust normalization techniques), the multimodal method can yield even better accuracy.

Feature-level fusion. Table 2 outlines our performance results from the feature-level fusion scheme, showing feature-level fusion yielded significantly greater accuracy in authentication compared to unimodal schemes.

Our experiments clearly reflect the potential of multimodal bio-

Table 1. EER results from score-level fusion.

Modality	EER	Testing Time (sec.)
Face	27.17%	0.065
Voice	41.44%	0.045
Score-level fusion	25.70%	0.108

Table 2. EER results from feature-level fusion.

Modality	EER	Testing Time (sec.)
Face	4.29%	0.13
Voice	34.72%	1.42
Feature-level fusion	2.14%	1.57

metrics to enhance the accuracy of current unimodal biometrics-based authentication on mobile devices; moreover, according to how quickly the system is able to identify a legitimate user, the Proteus approach is scalable to consumer mobile devices. This is the first attempt at implementing two types of fusion schemes on a modern consumer mobile device while tackling the practical issues of user friendliness. It is also just the beginning. We are working on improving the performance and efficiency of both fusion schemes, and the road ahead promises endless opportunity.

Conclusion

Multimodal biometrics is the next logical step in biometric authentication for consumer-level mobile devices. The challenge remains in making multimodal biometrics usable for consumers of mainstream mobile devices, but little work has sought to add multimodal biometrics to them. Our work is the first step in that direction.

Imagine a mobile device you can unlock through combinations of face, voice, fingerprints, ears, irises, and retinas. It reads all these biometrics in one step similar to the iPhone's TouchID fingerprint system. This user-friendly interface utilizes an underlying robust fusion logic based on biometric sample quality, maximizing the device's chance of correctly identifying its owner. Dirty fingers, poorly illuminated or loud settings, and damage to biometric sensors are no longer showstoppers; if one biometric fails, others function as backups. Hackers must now gain access to the many modalities required to unlock the device; because these are biometric modalities, they are possessed only by the legitimate owner of the device. The device also uses cancelable biometric templates, strong encryption, and the Trusted Execution Environment for securely storing and processing all biometric data.

The Proteus multimodal biometrics scheme leverages the existing capabilities of mobile device hardware (such as video recording), but mobile hardware and software are not equipped to handle more so-

phisticated combinations of biometrics; for example, mainstream consumer mobile devices lack sensors capable of reliably acquiring iris and retina biometrics in a consumer-friendly manner. We are thus working on designing and building a device with efficient, user-friendly, inexpensive software and hardware to support such combinations. We plan to integrate new biometrics into our current fusion schemes, develop new, more robust fusion schemes, and design user interfaces allowing the seamless, simultaneous capture of multiple biometrics. Combining a user-friendly interface with robust multimodal fusion algorithms may well mark a new era in consumer mobile device authentication. **□**

References

1. Aronowitz, H., Min L., Toledo-Ronen, O., Harary, S., Geva, A., Ben-David, S., Rendel, A., Hoory, R., Ratha, N., Pankanti, S., and Nahamoo, D. Multimodal biometrics for mobile authentication. In *Proceedings of the 2014 IEEE International Joint Conference on Biometrics* (Clearwater, FL, Sept. 29–Oct. 2). IEEE Computer Society Press, 2014, 1–8.
2. Avila, C.S., Casanova, J.G., Ballesteros, F., Garcia, L.R.T., Gomez, M.F.A., and Sierra, D.S. *State of the Art of Mobile Biometrics, Liveness and Non-Coercion Detection*. Personalized Centralized Authentication System Project. Jan. 31, 2014; <https://www.pcasp-project.eu/images/Deliverables/PCAS-D3.1.pdf>
3. Belhumeur, P.N., Hespanha, J.P., and Kriegman, D. Eigenfaces vector vs. FisherFaces: Recognition using class-specific linear projection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on Pattern Analysis and Machine Intelligence* 19, 7 (July 1997), 711–720.
4. Bonnington, C. The trouble with Apple's Touch ID fingerprint reader. *Wired* (Dec. 3, 2013); <http://www.wired.com/gadgetlab/2013/12/touch-id-issues-and-fixes/>
5. Dalal, N. and Triggs, B. Histograms of oriented gradients for human detection. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (San Diego, CA, June 20–25). IEEE Computer Society Press, 2005, 886–893.
6. Daugman, J.G. Two-dimensional spectral analysis of cortical receptive field profiles. *Vision Research* 20, 10 (Dec. 1980), 847–856.
7. Devine, R. Face Unlock in Jelly Bean gets a 'liveness check'. *AndroidCentral* (June 29, 2012); <http://www.androidcentral.com/face-unlock-jelly-bean-gets-liveness-check>
8. Duchnowski, P., Hunke, M., Busching, D., Meier, U., and Waibel, A. Toward movement-invariant automatic lip-reading and speech recognition. In *Proceedings of the 1995 International Conference on Acoustics, Speech, and Signal Processing* (Detroit, MI, May 9–12). IEEE Computer Society Press, 1995, 109–112.
9. Hansen, J.H.L. Analysis and compensation of speech under stress and noise for environmental robustness in speech recognition. *Speech Communication* 20, 1 (Nov. 1996), 151–173.
10. Hsu, D., Kakade, S.M., and Zhang, T. A spectral algorithm for learning hidden Markov models. *Journal of Computer and System Sciences* 78, 5 (Sept. 2012), 1460–1480.
11. Jain, A.K., Nandakumar, K., and Ross, A. Score normalization in multimodal biometric systems. *Pattern Recognition* 38, 12 (Dec. 2005), 2270–2285.
12. Kisku, D.R., Gupta, P., and Sing, J.K. Feature-level fusion of biometrics cues: Human identification with Doddingtons Caricature. *Security Technology* (2009), 157–164.
13. Kuncheva, L.I., Whitaker, C.J., Shipp, C.A., and Duin, R.P.W. Is independence good for combining classifiers? In *Proceedings of the 15th International Conference on Pattern Recognition* (Barcelona, Spain, Sept. 3–7). IEEE Computer Society Press, 2000, 168–171.
14. Lee, C. Automatic recognition of animal vocalizations using averaged MFCC and linear discriminant analysis. *Pattern Recognition Letters* 27, 2 (Jan. 2006), 93–101.
15. M2SYS Technology. SecuredPass AFIS/ABIS Immigration and Border Control System; <http://www.m2sys.com/automated-fingerprint-identification-system-afis-border-control-and-border-protection/>
16. Milner, B. and Xu, S. Speech reconstruction from mel-frequency cepstral coefficients using a source-filter model. In *Proceedings of the INTERSPEECH Conference* (Denver, CO, Sept. 16–20). International Speech Communication Association, Baixas, France, 2002.
17. Nasrollahi, K. and Moeslund, T.B. Face-quality assessment system in video sequences. In *Proceedings of the Workshop on Biometrics and Identity Management* (Roskilde, Denmark, May 7–9). Springer, 2008, 10–18.
18. Parala, A. UAE Airports get multimodal security. *FindBiometrics Global Identity Management* (Mar. 13, 2015); <http://findbiometrics.com/uae-airports-get-multimodal-security-23132/>
19. Rathgeb, C. and Andreas U. A survey on biometric cryptosystems and cancelable biometrics. *EURASIP Journal on Information Security* (Dec. 2011), 1–25.
20. Ross, A. and Govindarajan, R. Feature-level fusion of hand and face biometrics. In *Proceedings of the Conference on Biometric Technology for Human Identification* (Orlando, FL). International Society for Optics and Photonics, Bellingham, WA, 2005, 196–204.
21. Ross, A. and Jain, A. Multimodal biometrics: An overview. In *Proceedings of the 12th European Signal Processing Conference* (Sept. 6–10). IEEE Computer Society Press, 2004, 1221–1224.
22. Sacco, A. Fingerprint faceoff: Apple TouchID vs. Samsung Finger Scanner. *Chief Information Officer* (July 16, 2014); <http://www.cio.com/article/2454883/consumer-technology/fingerprint-faceoffapple-touch-id-vs-samsung-finger-scanner.html>
23. Tapellini, D.S. Phone thefts rose to 3.1 million last year. *Consumer Reports* finds industry solution falls short, while legislative efforts to curb theft continue. *Consumer Reports* (May 28, 2014); <http://www.consumerreports.org/cro/news/2014/04/smart-phone-thefts-rose-to-3-1-million-last-year/index.htm>
24. Viola, P. and Jones, M. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (Kauai, HI, Dec. 8–14). IEEE Computer Society Press, 2001.
25. Vondrasek, M. and Pollak, P. Methods for speech SNR estimation: Evaluation tool and analysis of VAD dependency. *Radioengineering* 14, 1 (Apr. 2005), 6–11.
26. Zorabedian, J. Samsung Galaxy S5 fingerprint reader hacked—It's the iPhone 5s all over again! *Naked Security* (Apr. 17, 2014); <https://nakedsecurity.sophos.com/2014/04/17/samsung-galaxy-s5-fingerprint-hacked-iphone-5s-all-over-again/>

Mikhail I. Gofman (mgofman@fullerton.edu) is an assistant professor in the Department of Computer Science at California State University, Fullerton, and director of its Center for Cybersecurity.

Sinjini Mitra (smitra@fullerton.edu) is an assistant professor of information systems and decision sciences at California State University, Fullerton.

Tracing the first four decades in the life of suffix trees, their many incarnations, and their applications.

BY ALBERTO APOSTOLICO, MAXIME CROCHEMORE, MARTIN FARACH-COLTON, ZVI GALIL, AND S. MUTHUKRISHNAN

40 Years of Suffix Trees

WHEN WILLIAM LEGRAND finally decrypted the string, it did not seem to make much more sense than it did before.

53†††305))6*,48264†.)4z);806",48†8P60))85;1†
(;†*8†83(88)5*†,46(;88*96*?;8)*†(;485);5*†2:*†
(;4956*2(5*Ñ4)8P8*;4069285);)6†8)4††;1(†9;48081;8:
8†1;4885;4)485†528806*81(ddag9;48;(88;4(†?34;
48)4†;161;:188;†?;

The decoded message read: “A good glass in the bishop’s hostel in the devil’s seat forty-one degrees and thirteen minutes northeast and by north main branch seventh limb east side shoot from the left eye of the death’s-head a bee line from the tree through the shot fifty feet out.” But at least it did sound more like natural language, and eventually guided the main character of Edgar Allan Poe’s “The Gold-Bug”³⁶ to discover the treasure he had been after. Legrand solved a substitution cipher using symbol frequencies.

He first looked for the most frequent symbol and changed it into the most frequent letter of English, then similarly inferred the most frequent word, then punctuation marks, and so on.

Both before and after 1843, the natural impulse when faced with some mysterious message has been to count frequencies of individual tokens or subassemblies in search of a clue. Perhaps one of the most intense and fascinating subjects for this kind of scrutiny have been biosequences. As soon as some such sequences became available, statistical analysts tried to link characters or blocks of characters to relevant biological functions. With the early examples of whole genomes emerging in the mid-1990s, it seemed natural to count the occurrences of all blocks of size 1, 2, and so on, up to any desired length, looking for statistical characterizations of coding regions, promoter regions, among others.

This article is not about cryptography. It is about a data structure and its variants, and the many surprising and useful features it carries. Among these is the fact that, to set up a statistical table of occurrences for all substrings (also called *factors*), of any length, of a text string of n characters, it only takes time and space linear in the length of the text string. While nobody would be so foolish as to solve the problem by first generating all exponentially many possible strings and then counting their occurrences one by one, a text string may still contain $\Theta(n^2)$ distinct substrings, so that tabulating all of them in linear space, never mind linear time, already seems puzzling.

We dedicate this article to our friend and colleague, Alberto Apostolico (1948–2015), who passed away on July 20. He was a major figure in the development of algorithms on strings.



Over the years, such structures have held center stage in text searching, indexing, statistics, and compression as well as in the assembly, alignment, and comparison of bi-

sequences. Their range of scope extends to areas as diverse as detecting plagiarism, finding surprising substrings in a text, testing the unique decipherability of a code, and more.

Their impact on computer science and IT at large cannot be overstated. Text searching and bioinformatics would not be the same without them. In 2013, the Combinatorial Pattern Matching symposium celebrated the 40th anniversary of the appearance of Weiner's invention of the suffix tree⁴¹ with a special session entirely dedicated to that event.

History Bits and Pieces

At the dawn of “stringology,” Donald Knuth conjectured the problem of finding the longest substring common to two long text sequences of total length n required $(n \log n)$ time. An $O(n \log n)$ -time had been provided by Karp, Miller, and Rosenberg.²⁶ That construction was destined to play a role in parallel pattern matching, but Knuth’s conjecture was short lived: in 1973, Peter Weiner showed the problem admitted an elegant linear-time solution,⁴¹ as long as the alphabet of the string was fixed. Such a solution was actually a byproduct of a construction he had originally set up for a different purpose, that is, identifying any substring of a text file without specifying all of them. In doing so, Weiner introduced the notion of a textual inverted index that would elicit refinements, analyses, and applications for 40 years and counting, a feature hardly shared by any other data structure.

Weiner's original construction processed the text file from right to left. As each new character was read in, the structure, which he called a "bi-tree," would be updated to accommodate longer and longer suffixes of the text file. Thus, this was an inherently off-line construction, since the text had to be known in its entirety before the construction could begin. Alternatively, one could say the algorithm would build the structure for the reverse of the text online. About three years later, Ed McCreight provided a left-to-right algorithm and changed the name of the structure to "suffix tree," a name that would stick.³²

Let x be a string of $n - 1$ symbols over some alphabet Σ and $\$$ an extra character not in Σ . The expanded suffix tree T_x associated with x is a digital search tree collecting all suffixes of $x\$$. Specifically, T_x is defined as follows.

Figure 1. The expanded suffix tree of the string $x = abcabcaba$.

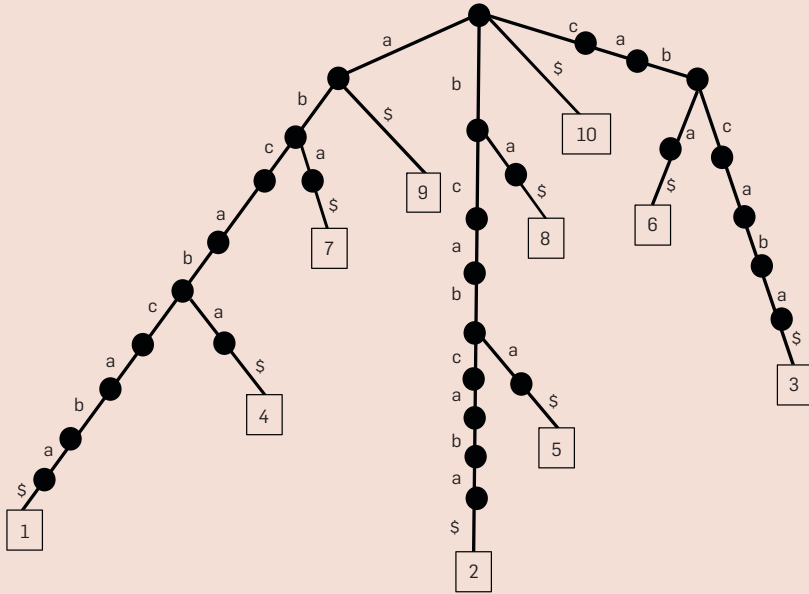
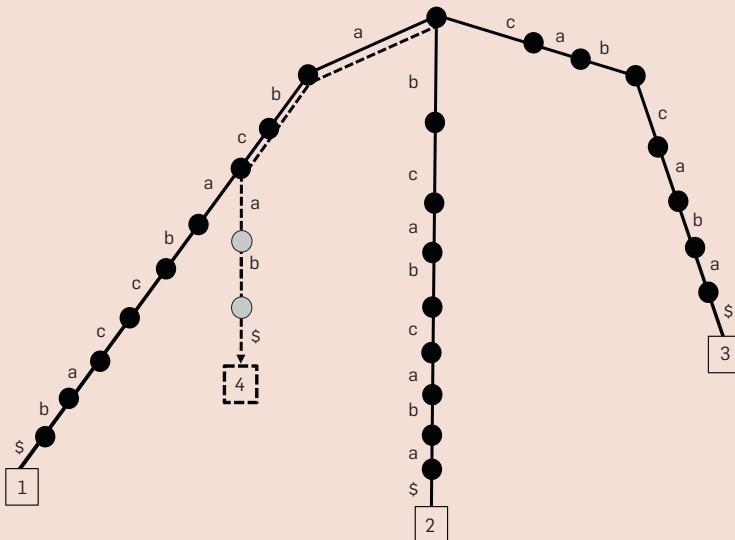


Figure 2. Building an expanded suffix tree by insertion of consecutive suffixes (showing here the insertion of *abcaba\$*).

The insertion of suffix $\text{su}f_i$ ($i = 1, 2, \dots, n$) consists of two phases. In the first phase, we search for $\text{su}f_i$ in T_{i-1} . Note the presence of $\$$ guarantees that every suffix will end in a distinct leaf. Therefore, this search will end with failure sooner or later. At that point, we will have identified the longest prefix of $\text{su}f_i$ that has a locus (that is, a terminal node) in T_{i-1} . Let $\text{head}_i \text{abcab}$ in the example be this prefix and α the locus of head_i . We can write $\text{su}f_i = \text{head}_i \cdot \text{tail}_i$ with tail_i ($a\$$ in the example) nonempty. In the second phase, we need to add to T_{i-1} a path leaving node α and labeled tail_i . This achieves the transformation of T_{i-1} into T_i .



1. T_x has n leaves, labeled from 1 to n .
 2. Each arc is labeled with a symbol of $\Sigma \cup \{\$$. For any i , $1 \leq i \leq n$, the concatenation of the labels on the path from the root of T_x to leaf i is precisely the suffix

$$\text{suf}_i = x_i x_{i+1} \dots x_{n-1} \$.$$

3. For any two suffixes suf_i and suf_j of $x\$$, if w_{ij} is the longest common prefix that suf_i and suf_j have in common, then the path in T_x relative to w_{ij} is the same for suf_i and suf_j .

An example of expanded suffix tree is given in Figure 1.

The tree can be interpreted as the state transition diagram of a deterministic finite automaton where all nodes and leaves are final states, the root is the initial state, and the labeled arcs, which are assumed to point downward, represent part of the state-transition function. The state transitions not specified in the diagram lead to a unique non-final *sink* state. Our automaton recognizes the (finite) language consisting of all substrings of string x . This observation also clarifies how the tree can be used in an online search: letting y be the pattern, we follow the downward path in the tree in response to consecutive symbols of y , one symbol at a time. Clearly, y occurs in x if and only if this process leads to a final state. In terms of T_x , we say the locus of a string y is the node α , if it exists, such that the path from the root of T_x to α is labeled y .

An algorithm for the direct construction of the expanded T_x (often called suffix trie) is readily derived (see Figure 2). We start with an empty tree and add to it the suffixes of $x\$$ one at a time. This procedure takes time $\Theta(n^2)$ and $O(n^2)$ space, however, it is easy to reduce space to $O(n)$ thereby producing a suffix tree in compact form (Figure 3). Once this is done, it becomes possible to aim for an expected non-trivial $O(n)$ time construction.

At the CPM Conference of 2013, McCreight revealed his $O(n)$ time construction was not born as an alternative to Weiner's—he had developed it in an effort to understand Weiner's paper, but when he showed it to Weiner asking him to confirm he had understood that paper the answer was “No, but you have come

up with an entirely different and elegant construction!” In unpublished lecture notes of 1975, Vaughan Pratt displayed the duality of this structure and Weiner's “repetition finder.”³⁷ McCreight's algorithm was still inherently offline, and it immediately triggered a search for an online version. Some partial attempts at an online algorithm were made, but such a variant had to wait almost two decades for Esko Ukkonen's paper in 1995.³⁹ In all these linear-time constructions, linearity was based on the assumption of a finite alphabet and took $\Theta(n \log n)$ time without that assumption. In 1997, Martin Farach introduced an algorithm that abandoned the one suffix-at-time approach prevalent until then; this algorithm gives a linear-time reduction from suffix-tree construction to character sorting, and thus is optimal for all alphabets.¹⁷ In particular, it runs in linear time for a larger class of alphabets, for example, when the alphabet size is polynomial in input length.

Around 1984, Blumer et al.⁹ and Crochemore¹⁴ exposed the surprising result that the smallest finite automaton recognizing all and only the suffixes of

» key insights

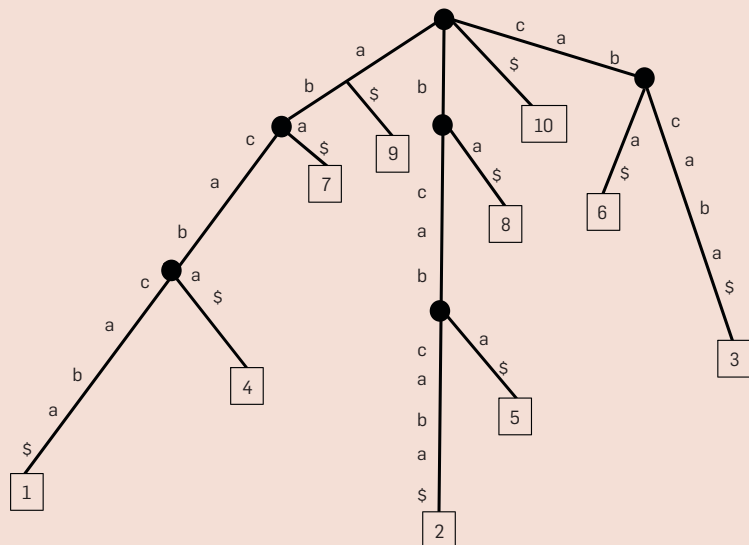
- The suffix tree is the core data structure in string analysis.
- It has a rich history, with connections to compression, matching, automata, data structures and more.
- There are powerful techniques to build suffix trees and use them efficiently in many applications.

a string of n characters has only $O(n)$ states and edges. Initially coined a directed acyclic word graph (DAWG), it can even be further reduced if all states are terminal states.¹⁴ It then accepts all substrings of the string and is called the factor—substring automaton. There is a nice relation between the index data structures when the string has no end-marker and its suffixes are marked with terminal states in the tree.

Then, the suffix tree is the edge-compacted version of the tree and its number of nodes can be minimized like with any automaton thereby providing the compact DAWG of the string. Permuting the two operations, compaction and minimization, leads to the same structure. Apparently Anatoli Slissenko (see the appendix avail-

Figure 3. A suffix tree in compact form.

This is obtained by first collapsing every chain formed by nodes with only one child into a single arc. The resulting compact version of T_x has at most n internal nodes, since there are $n + 1$ leaves in total and every internal node is branching. The labels of the generic arc are now a substring, rather than a symbol of $x\$$. However, arc labels can be expressed by suitable pairs of pointers to a common copy of $x\$$ thus achieving $O(n)$ space bound overall.



able with this article in the ACM Digital Library under Source Material) ended up with a similar structure for his work on the detection of repetitions in strings. These automata provide another more efficient counterexample to Knuth's conjecture when they are used, against the grain, as pattern-matching machines (see Figure 4).

The appearance of suffix trees dovetailed with some interesting and independent developments in information theory. In his famous approach to the notion of information, Kolmogorov equated the information or structure in a string to the length of the shortest program that would be needed to produce that string by a Universal Turing Machine. The unfortunate thing is this measure is not computable and even if it were, most long strings are incompressible (that is, lack a short program producing them), since there are increasingly many long strings and comparatively much fewer short programs (themselves strings).

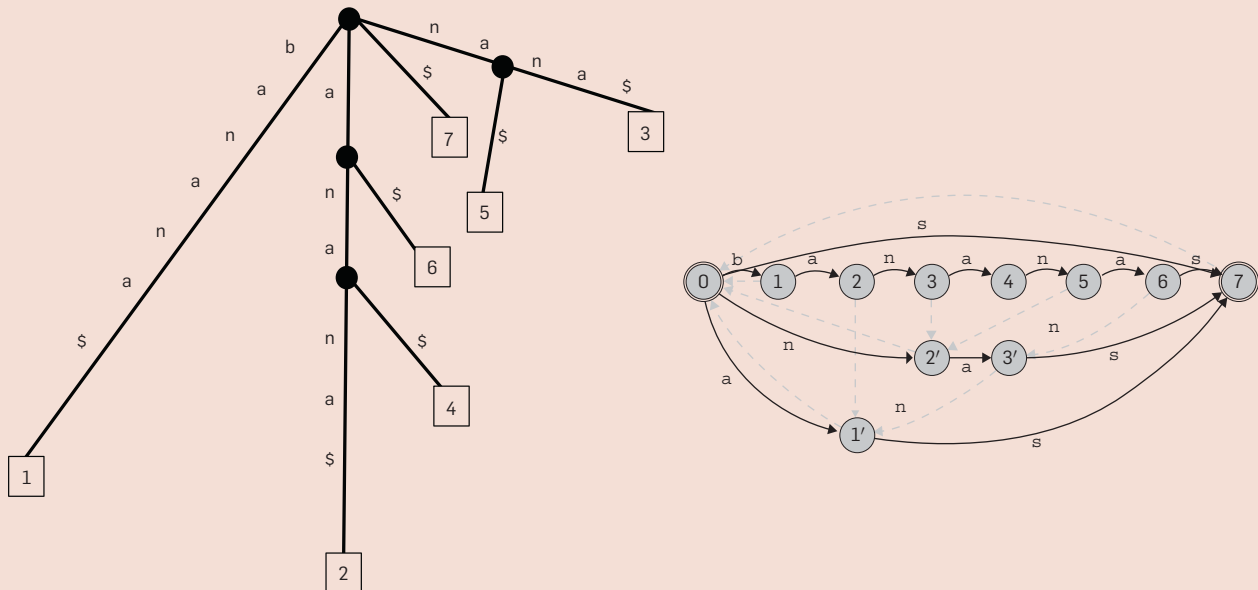
The regularities exploited by Kolmogorov's universal and omniscient machine could be of any conceivable kind, but what if one limited them to the syntactic redundancies affecting a text in the form of repeated substrings? If a string is repeated many times one could profitably encode all occurrences by a pointer to a common copy. This copy could be internal or external to the text. In the former case one could have pointers going in both directions or only in one direction, allow or forbid nesting of pointers, and so on. In his doctoral thesis, Jim Storer showed that virtually all such "macro schemes" are intractable, except one. Not long before that, in a landmark paper entitled "On the Complexity of Finite Sequences,"³⁰ Abraham Lempel and Jacob Ziv had proposed a variable-to-block encoding, based on a simple parsing of the text with the feature that the compression achieved would match, in the limit, that produced by a compressor tailored to the source probabilities.

Thus, by a remarkable alignment of stars, the compression method brought about by Lempel and Ziv was not only optimal in the information theoretic sense, but it found an optimal, linear-time implementation by the suffix tree, as was detailed immediately by Michael Rodeh, Vaughan Pratt, and Shimon Even.³⁸

In his original paper, Weiner listed a few applications of his "bi-tree" including most notably offline string searching: preprocessing a text file to support queries that return the occurrences of a given pattern in time linear in the length of the pattern. And of course, the "bi-tree" addressed Knuth's conjecture, by showing how to find the longest substring common to two files in linear time for a finite alphabet. There followed unpublished notes by Pratt entitled "Improvements and Applications for the Weiner Repetition Finder."³⁷ A decade later, Alberto Apostolico would list more applications in a paper entitled "The Myriad Virtues of Suffix Trees,"³²

Figure 4. The compact suffix tree (left) and the suffix automaton (right) of the string "bananas."

Failure links are represented by the dashed arrows. Despite the fact it is an index on the string, the same automaton can be used as a pattern-matching machine to locate substrings of "bananas" in another text or to compute their longest common substring. The process runs online on the second string. Assume for example "bana" has just been scanned from the second string and the current state of the automaton is state 4. If the next letter is "n," the common substring is "banan" of length 5 and the new state is 5. If the next letter is "s," the failure link is used and from state 3' corresponding to a common substring "ana" of length 3 we get the common substring "ana" with the new state 7. If the next letter is "b," iterating the failure link leads to state 0 and we get the common substring "b" with the new state 1. Finally, any other next letter will produce the empty common substring and state 0.



and two decades later suffix trees and companion structures with their applications gave rise to several chapters in reference books by Crochemore and Rytter, Dan Gusfield, and Crochemore, Hancart, and Lecroq (see the appendix available with this article in the ACM Digital Library).

The space required by suffix trees has been a nuisance in applications where they were needed the most. With genomes on the order of gigabytes, for instance, the space difference between 20 times larger than the source versus, say, only 11 times larger, can be substantial. For a few lustra, Stefan Kurtz and his co-workers devoted their effort to cleverly allocating the tree and some of its companion structures.²⁸ In 2001, David R. Clark and J. Ian Munro proposed one of the best space-saving methods on secondary storage.¹³ Clark and Munro's "succinct suffix tree" sought to preserve as much of the structure of the suffix tree as possible. Udi Manber and Eugene W. Myers took a different approach, however. In 1990, they introduced the "suffix array,"³¹ which eliminated most of the structure of the suffix tree, but was still able to implement many of the same operations, requiring space equal to 2 integers per text character and searching in time $O(|P| + \log n)$ (reducible to 1 by accepting search time $O(|P| + \log n)$). The suffix array stores the suffixes of the input in lexicographic order and can be seen as the sequence of leaves' labels as found in the suffix tree by a preorder traversal that would expand each node according to the lexicographic order.

Although the suffix array seemed at first to be a different data structure than the suffix tree, the distinction has receded. For example, Manber and Myers's original construction of the suffix array took $O(n \log n)$ time for any alphabet, but the suffix array could be constructed in linear time from the suffix tree for any alphabet. In 2001, Toru Kasai et al.²⁷ showed the suffix tree could be constructed in linear time from the suffix array. Therefore, the suffix array was shown to be a succinct representation of the suffix tree. In 2003, three groups presented three different modifications of Farach's algorithm for suffix tree con-



Although the suffix array seemed at first to be a different data structure than the suffix tree, the distinction has receded.



struction to give the first linear-time algorithms for directly constructing the suffix array; that is, the first linear-time algorithms for computing suffix arrays that did not first compute the full suffix tree. Since then, there have been many algorithms for fast construction of suffix arrays, notably by Nong, Zhang, and Chan,³⁵ which is linear time and fast in practice. With fast construction algorithms and small space required, the suffix array is the suffix-tree variant that has gained the most widespread adoption in software systems. A more recent succinct suffix tree and array, which take $O(n)$ bits to represent for a binary alphabet ($O(n \log \sigma)$ bits otherwise), was presented by Grossi and Vitter.²¹

Actually, the histories of suffix trees and compression are tightly intertwined. This should not come as a surprise, since the redundancies that pattern discovery tries to unearth are ideal candidates to be removed for purposes of compression. In 1994, M. Burrows and D.J. Wheeler proposed a breakthrough compression method based on suffix sorting.¹¹ Circa 1995, Amihoud Amir, Gary Benson, and Martin Farach posed the problem of searching in compressed texts.¹ In 2000, Paolo Ferragina and Giovanni Manzini introduced the FM-index, a compressed suffix array based on the Burrows-Wheeler transform.¹⁹ This structure, which may be smaller than the source file, supports searching without decompression. This was extended to compressed tree indexing problems in Ferragina et al.¹⁸ using a modification of the Burrows-Wheeler transform.

Fallout, Extensions, and Challenges

As highlighted out the outset, there has been hardly any application of text processing that did not need these indexes at one point or another. A prominent case has been searching with errors, a problem first efficiently tackled in 1985 by Gad Landau in his Ph.D. thesis.²⁹ In this kind of search, one looks for substrings of the text that differ from the pattern in a limited number of errors such as a single character deletion, insertion or substitution. To efficiently solve this problem, Landau combined suf-

fix trees with a clever solution to the so-called lowest common ancestor (LCA) problem. The LCA problem assumes a rooted tree is given and then it seeks, for any pair of nodes, the lowest node in the tree that is an ancestor of both.²³ It is seen that following a linear-time preprocessing of the tree any LCA query can be answered in constant time. Landau used LCA queries on suffix trees to perform constant-time jumps over segments of the text that would be guaranteed to match the pattern. When k errors are allowed, the search for an occurrence at any given position can be abandoned after k such jumps. This leads to an algorithm that searches for a pattern with k errors in a text of n characters in $O(nk)$ steps.

Among the basic primitives supported by suffix trees and arrays, one finds, of course, the already mentioned search for a pattern in a text in time proportional to the length of the pattern rather than the text. In fact, it is even possible to enumerate occurrences in time proportional to their number and, with trivial preprocessing of the tree, tell the total number of occurrences for any query pattern in time proportional to the pattern size. The problem of finding the longest substring appearing twice in a text or shared between two files has been noted previously: this is probably where it all started. A germane problem is that of detecting squares, repetitions, and maximal periodicities in a text, a problem rooted in work by Axel Thue dated more than a century ago with multiple contemporary applications in compression and DNA analysis. A square is a pattern consisting of two consecutive occurrences of the same string. Suffix trees have been used to detect in optimal $O(n \log n)$ time all squares (or repetitions) in a text, each with its set of starting positions,⁵ and later to find and store all distinct square substrings in a text in linear time. Squares play a role in an augmentation of the suffix tree suitable to report, for any query pattern, the number of its non-overlapping occurrences.^{6,10}

There are multiple uses of suffix trees in setting up some kind of signature for text strings, as well as measures of similarity or difference.



There are multiple uses of suffix trees in setting up some kind of signature for text strings, as well as measures of similarity or difference.



Among the latter, there is the problem of computing the forbidden or absent words of a text, which are minimal strings that do not appear in the text (while all their proper substrings do).^{8,15} Such words lead to, among other things, an original approach to text compression.¹⁶ Once regarded as the succinct representation of the “bag-of-words” of a text, suffix trees can be used to assess the similarity of two text files, thereby supporting clustering, document classification, and even phylogeny.^{4,12,40} Intuitively, this is done by assessing how much the trees for the two input sequences have in common. Suitably enriched with the probability of the substring ending at each node, a tree can be used to detect surprisingly over-represented substrings of any length,³ for example, in the quest of promoter regions in bi-sequences.

The suffix tree of the concatenation of say, $k \geq 2$ text files, supports efficient solutions to problems arising in domains ranging from plagiarism detection to motif discovery in biosequences. The need for k distinct end-markers poses some subtleties in maintaining linear time, for which the reader is referred to Gusfield.²² In its original form, the problem of indexing multiple texts was called the “color problem” and seeks to report, for any given query string and in time linear in the query, how many documents out of the total of k contain at least one occurrence of the query. A simple and elegant solution was given in 1992 by Lucas C.K. Hui.²⁵ Recently, the combined suffix trees of many strings (also known as the generalized suffix tree) was used to solve a variety of document listing problems. Here, a set of text documents is preprocessed as a combined suffix tree. The problem is to return the list of all documents that contain a query pattern in time proportional to the number of such documents, not to the total number of occurrences (occ), which can be significantly larger. This problem was solved in Muthukrishnan³³ by reducing it to *range minimum queries*. This basic document-listing problem has since been extended to many other problems including listing the top- k in various string and information distances. For example, in Hon

et al.,²⁴ the structure of generalized suffix tree is crucially used to design a linear machine-word data structure to return the top- k most frequent documents containing a pattern p in time nearly linear in pattern size.

One surprising variant of the suffix tree was introduced by Brenda Baker for purposes of detection of plagiarism in student reports as well as optimization in software development.⁷ This variant of pattern matching, called “*parameterized matching*,” enables one to find program segments that are identical up to a systematic change of parameters, or substrings that are identical up to a systematic relabeling or permutation of the characters in the alphabet. One obvious extension of the notion of a suffix tree is to more than one dimension, albeit the mechanics of the extension itself are far from obvious.³⁴ Among more distant relatives, one finds “*wavelet trees*.” Originally proposed as a representation of compressed suffix arrays,²⁰ wavelet trees enable one to perform on general alphabets the ranking and selection primitives previously limited to bit vectors, and more.

The list could go on and on, but the scope of this article was not meant to be exhaustive. Actually, after 40 years of unrelenting developments, it is fair to assume the list will continue to grow. Open problems also abound. For instance, many of the observed sequences are expressed in numbers rather than characters, and in both cases are affected by various types of errors. While the outcome of a two-character comparison is just one bit, two numbers can be more or less close, depending on their difference or some other metric. Likewise, two text strings can be more or less similar, depending on the number of elementary steps necessary to change one in the other. The most disruptive aspect of this framework is the loss of the transitivity property that leads to the most efficient exact string matching solutions. And yet indexes capable of supporting fast and elegant approximate pattern queries of the kind just highlighted would be immensely useful. Hopefully, they will come up soon and, in time, have their own 40th-anniversary celebration.

Acknowledgments. We are grateful to Ed McCreight, Ronnie Martin, Vaughan Pratt, Peter Weiner, and Jacob Ziv for discussions and help. We are indebted to the referees for their careful scrutiny of an earlier version of this article, which led to many improvements. 

References

- Amir, A., Benson, G. and Farach, M. Let sleeping files lie: Pattern matching in Z-compressed files. In *Proceedings of the 5th ACM-SIAM Annual Symposium on Discrete Algorithms* (Arlington, VA, 1994), 705–714.
- Apostolico, A. The myriad virtues of suffix trees. *Combinatorial Algorithms on Words*, vol. 12 of *NATO Advanced Science Institutes, Series F. A.* Apostolico and Z. Galil, Eds. Springer-Verlag, Berlin, 1985, 85–96.
- Apostolico, A., Bock, M.E. and Lonardi, S. Monotony of surprise and large-scale quest for unusual words. *J. Computational Biology* 10, 3 / 4 (2003), 283–311.
- Apostolico, A., Denas, O. and Dress, A. Efficient tools for comparative substring analysis. *J. Biotechnology* 149, 3 (2010), 120–126.
- Apostolico, A. and Preparata, F.P. Optimal off-line detection of repetitions in a string. *Theor. Comput. Sci.* 22, 3 (1983), 297–315.
- Apostolico, A. and Preparata, F.P. Data structures and algorithms for the strings statistics problem. *Algorithmica* 15, 5 (May 1996), 481–494.
- Baker, B.S. Parameterized duplication in strings: Algorithms and an application to software maintenance. *SIAM J. Comput.* 26, 5 (1997), 1343–1362.
- Béal, M.-P., Mignosi, F. and Restivo, A. Minimal forbidden words and symbolic dynamics. In *Proceedings of the 13th Annual Symposium on Theoretical Aspects of Computer Science*, vol. 1046 of *Lecture Notes in Computer Science* (Grenoble, France, Feb. 22–24, 1996). Springer, 555–566.
- Blumer, A., Blumer, J., Ehrenfeucht, A., Haussler, D., Chen, M.T. and Seiferas, J. The smallest automaton recognizing the subwords of a text. *Theor. Comput. Sci.* 40, 1 (1985), 31–55.
- Brodal, G.S., Lyngsø, R.B., Östlin, A. and Pedersen, C.N.S. Solving the string statistics problem in time $O(n \log n)$. In *Proceedings of the 29th International Colloquium on Automata, Languages and Programming*, vol. 2380 of *Lecture Notes in Computer Science* (Malaga, Spain, July 8–13, 2002). Springer, 728–739.
- Burrows, M. and Wheeler, D.J. A block-sorting lossless data compression algorithm. Technical Report 124, Digital Equipment Corp., May 1994.
- Chairungsee, S. and Crochemore, M. Using minimal absent words to build phylogeny. *Theoretical Computer Science* 450, 1 (2012), 109–116.
- Clark, D.R. and Munro, J.I. Efficient suffix trees on secondary storage. In *Proceedings of the 7th ACM-SIAM Annual Symposium on Discrete Algorithms*, (Atlanta, GA, 1996), 383–391.
- Crochemore, M. Transducers and repetitions. *Theor. Comput. Sci.*, 45, 1 (1986), 63–86.
- Crochemore, M., Mignosi, F. and Restivo, A. Automata and forbidden words. *Information Processing Letters* 67, 3 (1998), 111–117.
- Crochemore, M., Mignosi, F., Restivo, A. and Salemi, S. Data compression using antidictionaries. In *Proceedings of the IEEE: Special Issue Lossless Data Compression* 88, 11 (2000). J. Storer, Ed., 1756–1768.
- Farach, M. Optimal suffix tree construction with large alphabets. In *Proceedings of the 38th IEEE Annual Symposium on Foundations of Computer Science* (Miami Beach, FL, 1997), 137–143.
- Ferragina, P., Luccio, F., Manzini, G. and Muthukrishnan, S. Compressing and indexing labeled trees with applications. *JACM* 57, 1 (2009).
- Ferragina, P. and Manzini, G. Opportunistic data structures with applications. In *FOCS* (2000), 390–398.
- Grossi, R., Gupta, A. and Vitter, J.S. High-order entropy-compressed text indexes. In *SODA* (2003), 841–850.
- Grossi, R. and Vitter, J.S. Compressed suffix arrays and suffix trees with applications to text indexing and string matching. In *Proceedings ACM Symposium on the Theory of Computing* (Portland, OR, 2000). ACM Press, 397–406.
- Gusfield, D. *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology*. Cambridge University Press, Cambridge, U.K., 1997.
- Harel, D. and Tarjan, R.E. Fast algorithms for finding nearest common ancestors. *SIAM J. Comput.* 13, 2 (1984), 338–355.
- Hon, W.-K., Shah, R. and Vitter, J.S. Space-efficient framework for top- k string retrieval problems. In *FOCS*. IEEE Computer Society, 2009, 713–722.
- Hui, L.C.K. Color set size problem with applications to string matching. In *Proceedings of the 3rd Annual Symposium on Combinatorial Pattern Matching*, no. 644 in *Lecture Notes in Computer Science*, (Tucson, AZ, 1992). A. Apostolico, M. Crochemore, Z. Galil, and U. Manber, Eds. Springer-Verlag, Berlin, 230–243.
- Karp, R.M., Miller, R.E., and Rosenberg, A.L. Rapid identification of repeated patterns in strings, trees and arrays. In *Proceedings of the 4th ACM Symposium on the Theory of Computing* (Denver, CO, 1972). ACM Press, 125–133.
- Kasai, T., Lee, G., Arimura, H., Arikawa, S. and Park, K. Linear-time longest-common-prefix computation in suffix arrays and its applications. *CPM*. Springer-Verlag, 2001, 181–192.
- Kurtz, S. Reducing the space requirements of suffix trees. *Softw. Pract. Exp.* 29, 13 (1999), 1149–1171.
- Landau, G.M. String matching in erroneous input. Ph.D. Thesis, Department of Computer Science, Tel-Aviv University, 1986.
- Lempel, A. and Ziv, J. On the complexity of finite sequences. *IEEE Trans. Inf. Theory* 22 (1976), 75–81.
- Manber, U. and Myers, G. Suffix arrays: A new method for on-line string searches. In *Proceedings of the 1st ACM-SIAM Annual Symposium on Discrete Algorithms* (San Francisco, CA, 1990), 319–327.
- McCreight, E.M. A space-economical suffix tree construction algorithm. *J. Algorithms* 23, 2 (1976), 262–272.
- Muthukrishnan, S. Efficient algorithms for document listing problems. In *Proceedings of the 13th ACM-SIAM Annual Symposium on Discrete Algorithms* (2002), 657–666.
- J. C. Na, P. Ferragina, R. Giancarlo, and K. Park. Two-dimensional pattern indexing. In *Encyclopedia of Algorithms*. 2008.
- Nong, G., Zhang, S. and Chan, W.H. Two efficient algorithms for linear time suffix array construction. *IEEE Trans. Comput.* 60, 10 (2011), 1471–1484.
- Poe, E.A. *The Gold-Bug and Other Tales*. Dover Thrift Editions Series. Dover, 1991.
- Pratt, V. Improvements and applications for the Weiner repetition finder. Manuscript, 1975.
- Rodeh, M., Pratt, V. and Even, S. Linear algorithm for data compression via string matching. *J. Assoc. Comput. Mach.* 28, 1 (1981), 16–24.
- Ukkonen, E. On-line construction of suffix trees. *Algorithmica* 14, 3 (1995), 249–260.
- Ulitsky, I., Burstein, D., Tuller, T. and Chor, B. The average common substring approach to phylogenomic reconstruction. *J. Computational Biology* 13, 2 (2006), 336–350.
- Weiner, P. Linear pattern matching algorithms. In *Proceedings of the 14th Annual IEEE Symposium on Switching and Automata Theory*, (Washington, D.C., 1973), 1–11.

Alberto Apostolico held joint appointments with Georgia Tech's School of Computational Science and Engineering School of Interactive computing as a professor and a researcher. He passed away on July 20, 2015.

Maxime Crochemore (maxime.crochemore@kcl.ac.uk) is a professor at King's College London and Université Paris-Est, France.

Martin Farach-Colton (farach@cs.rutgers.edu) is a professor in the Department of Computer Science at Rutgers University, Piscataway, NJ.

Zvi Galil (galil@cc.gatech.edu) is Dean of the College of Computing at Georgia Institute of Technology, Atlanta, GA.

S. Muthukrishnan (muthu@cs.rutgers.edu) is a professor in the Department of Computer Science at Rutgers University, Piscataway, NJ.

research highlights

P. 75

Technical Perspective Fairness and the Coin Flip

By David A. Wagner

P. 76

Secure Multiparty Computations on Bitcoin

By Marcin Andrychowicz, Stefan Dziembowski,
Daniel Malinowski, and Łukasz Mazurek

P. 85

Technical Perspective The State (and Security) of the Bitcoin Economy

By Emin Gün Sirer

P. 86

A Fistful of Bitcoins: Characterizing Payments among Men with No Names

By Sarah Meiklejohn, Marjori Pomarole, Grant Jordan,
Kirill Levchenko, Damon McCoy, Geoffrey M. Voelker, and Stefan Savage

Technical Perspective

Fairness and the Coin Flip

By David Wagner

ALICE AND BOB have a pleasant dinner together, and want to randomly choose who will have to wash the dishes afterward. How can they fairly choose? One time-honored method is for Alice to flip a coin (hiding it from Bob). Bob calls his guess, and then Alice can reveal the coin, revealing who is stuck washing dishes. Both can verify for themselves whether the procedure was fair.

What if Alice and Bob are on opposite sides of the globe, able to communicate only via the Internet? Over three decades ago, cryptographers designed a clever scheme for solving this coin-tossing problem: roughly, Alice flips a coin and sends Bob a cryptographic hash of the outcome; Bob sends Alice his guess; and then Alice can reveal the coin toss outcome, allowing both Alice and Bob to verify who won and who lost. This protocol is useful in distributed settings where multiple parties who do not trust each other want to jointly generate random values that no one can influence or bias.

The following paper introduces an exciting new idea for how to provide fairness: leverage Bitcoin's existing infrastructure for distributed consensus.

Unfortunately, this scheme has one shortcoming. Alice learns the outcome of the coin toss before Bob does. If Alice is dishonest or a poor loser, she can gain an unfair advantage. After Bob sends his guess, Alice knows whether she won or lost; if she won, she can continue to reveal the coin toss outcome and claim her winnings, but if she lost, she can refuse to continue the protocol, break the connection with Bob, and if necessary claim her computer crashed. This way, a dishonest Alice can ensure either she wins or no one does, which is unfair to Bob. This is known as the *fairness problem*.

In some applications, unfairness can be tolerated, for instance, if there is a way to punish cheaters or if the parties must place a deposit with a trusted escrow service before beginning the coin-flip process. In others, though, this is a serious problem. Researchers have explored various methods for providing fairness, but none are fully satisfactory. Moreover, there are negative results: in a general setting where there is no trusted third party for dispute resolution, the fairness problem appears to be unsolvable. The general view seemed to be that this is simply an unavoidable problem.

The following paper introduces an exciting new idea for how to provide fairness: leverage Bitcoin's existing infrastructure for distributed consensus. Bitcoin is a sophisticated distributed system that was designed to resist manipulation even by sophisticated, well-resourced attackers. The authors illustrate how we can build cryptographic protocols whose security rests on the foundation provided by Bitcoin: breaking the cryptographic protocol would require breaking Bitcoin, something that is believed to be difficult to do.

The paper exploits a fascinating feature of Bitcoin technology. Bitcoin

provides an audit log of transactions, and it allows transactions to contain scripts—programs that determine whether the transaction will happen. The authors use this aspect of Bitcoin to achieve fairness: scripts implement the functionality that would otherwise need to be provided by a trusted third-party escrow service.

More broadly, distributed coin flipping is not the only task we might want to perform in a distributed world. Decades ago, cryptographers studied the general problem of multi-party secure computation, where Alice and Bob want to jointly perform some computation on their data, but without revealing their own data to each other. Coin flipping is just one instance of this paradigm. Cryptographers have shown a very strong result: essentially every task of this form can be done securely. However, again these protocols suffer from an unavoidable fairness problem: one party learns the result of the computation before the other, and can terminate the protocol early and prevent the other from learning the output. One especially exciting aspect of this paper is that it suggests a direction for achieving fairness for general multi-party secure computation, if the parties are willing to use Bitcoin. Who would have predicted Bitcoin could have such implications for secure distributed computation? 

David Wagner is a professor of computer science at the University of California, Berkeley.

Copyright held by author.

Secure Multiparty Computations on Bitcoin

By Marcin Andrychowicz, Stefan Dziembowski, Daniel Malinowski, and Łukasz Mazurek

Abstract

Is it possible to design an online protocol for playing a lottery, in a completely decentralized way, that is, without relying on a trusted third party? Or can one construct a fully decentralized protocol for selling secret information, so that neither the seller nor the buyer can cheat in it? Until recently, it seemed that every online protocol that has financial consequences for the participants needs to rely on some sort of a trusted server that ensures that the money is transferred between them. In this work, we propose to use Bitcoin (a digital currency, introduced in 2008) to design such fully decentralized protocols that are secure even if no trusted third party is available. As an instantiation of this idea, we construct protocols for secure multiparty lotteries using the Bitcoin currency, without relying on a trusted authority. Our protocols guarantee fairness for the honest parties no matter how the loser behaves. For example, if one party interrupts the protocol, then her money is transferred to the honest participants. Our protocols are practical (to demonstrate it, we performed their transactions in the actual Bitcoin system) and in principle could be used in real life as a replacement for the online gambling sites.

1. INTRODUCTION

One of the most attractive features of the Internet is its decentralization: the TCP/IP protocol itself, and several other protocols running on top of it do not rely on a single server, and often can be executed between parties that do not need to trust each other, or even do not need to know each other's true identity. Examples of such protocols include: the SMTP and the HTTP protocols, the peer-to-peer content distributions platforms, messaging systems, and many others. A natural question to ask is how far can the "decentralization" of the digital world go? In other words, what are the real-life applications which one can implement on the Internet without the need of a trusted third party? Until recently, one notable example of a task that seemed to always require some sort of a "trusted server" was the online financial transactions (that had to rely on a bank or a credit card company). This situation changed radically in 2009 when the first fully decentralized digital currency, called Bitcoin, was deployed by Nakamoto.^{17, a} The huge success of Bitcoin (its current market capitalization is around \$5 billion) is due precisely to its distributed nature and the lack of a central authority that controls Bitcoin transactions. We describe Bitcoin in more detail in Section 2.

^a This name is widely believed to be a pseudonym.

The fact that Bitcoin money transfers can be done without a trusted server raises another intriguing question, namely, can we "decentralize" the financial system even further, that is, can we implement some more advanced financial instruments in a distributed manner? The Bitcoin specification partly answers this question, by providing the so-called "nonstandard transactions." We describe this feature in more detail in Section 2, but for a moment, let us only say that Bitcoin allows the parties to specify more complex conditions about when the money can be spent. This, in turn, permits them to create the so-called "Bitcoin contracts," which are forms of agreements whose execution is later enforced by the Bitcoin system itself (without the need of a trusted third party). Examples of such contracts include rapidly adjusted micropayments, assurance contracts, and dispute mediation (see <https://en.bitcoin.it/wiki/Contracts> for more on this).

Probably, one of the most advanced types of multiparty protocols that can be performed digitally are the cryptographic "secure multiparty computation (MPC)" protocols, originating from the seminal works of Yao²⁰ and Goldreich et al.¹⁴ Informally, such protocols allow a group of mutually distrusting parties to compute a joint function f on their private inputs. For example, for two parties, Alice and Bob, Alice has an input x , Bob has an input y , and they both want to learn $f(x, y)$, but without Alice learning y or Bob learning x . In this paper, we initiate the study of using Bitcoin to perform MPC protocols.

The coin-tossing protocol. A very simple example of such a protocol is the *coin-tossing problem*,⁶ executed between two parties, Alice and Bob, who want to jointly compute a bit b that is equally likely to be 0 or 1. In other words, they want to compute a randomized function $f_{\text{rnd}} : \{\perp\} \times \{\perp\} \rightarrow \{0, 1\}$ that takes no inputs and outputs as a uniformly random bit. This protocol can be implemented using an idea similar to the rock-paper-scissors game: Alice sends a bit b_A to Bob, and simultaneously Bob sends a bit b_B to Alice. The output b is computed

A longer version of this paper appeared on the IEEE Symposium on Security and Privacy 2014. An extended version of it is also available on the Cryptology Eprint Archive eprint.iacr.org/2013/784. This work was supported by the WELCOME/2010-4/2 grant founded within the framework of the EU Innovative Economy (National Cohesion Strategy) Operational Programme. Łukasz Mazurek is a recipient of the Google Europe Fellowship in Security, and this research is supported in part by this Google Fellowship.

as $b := b_A \oplus b_B$ (where “ $\oplus$ ” denotes the XOR function). Clearly if at least one of the bits b_A and b_B is uniformly random, then b is also uniformly random, and hence each party can be sure that the game is fair, as long as she behaves honestly (i.e., chooses her bit uniformly). When one tries to implement this protocol over the Internet, then of course, the main challenge is to ensure that Alice and Bob send their bits simultaneously. This is because if one party, say Alice, can choose her bit b_A after she learns b_B , then she can make b equal to any value b' she wants by choosing $b_A := b' \oplus b_B$.

The solution proposed in Blum⁶ is to use a tool called a *cryptographic commitment scheme*. Informally, such a scheme is a two-party protocol executed between a *committer* and a *receiver*. At the beginning, the committer knows some value s that is secret to the receiver. The parties first perform the *commitment phase* (*Commit*). After this phase is executed, the receiver still does not know s (this property is called *hiding*). Later, the parties execute the opening phase (*Open*) during which the receiver learns s . The key property of a commitment scheme is that the committer cannot “change his mind” after the commitment phase. More precisely, after the first phase is executed, there exists precisely one value s that can be opened in the second phase. This property is called *binding*. In some sense, the commitment phase is analogous to sending a message s in a locked box, and the opening phase can be thought of as sending the key to the box. Clearly after the box is sent, the committer cannot change its contents, but before getting the key, the receiver does not know what is inside the box.

There exist several secure methods of constructing such commitments. In this paper, we use the ones that are based on the cryptographic hash functions (see Section 3).

It is now easy to see how a commitment scheme can be used to solve the coin-tossing problem: instead of sending her bit b_A directly to Bob, Alice just commits to it (i.e., Alice and Bob execute the commitment scheme with Alice acting as the committer, Bob acting as the receiver, and b_A being the secret). Symmetrically, Bob commits to his bit b_B . After this commitment phase is over, the parties execute the opening phase and learn each other's bits. Then the output is computed as $b = b_A \oplus b_B$. The security of the commitment scheme guarantees that no party can choose her bit depending on the bit of the other party, and hence this procedure produces a uniformly random bit.

Boolean operations. The coin-tossing example above is a particularly simple case of a multiparty protocol since the parties that execute it do not take any inputs. To explain what we mean by a protocol where the parties do take inputs, consider the case when the function that Alice and Bob compute is the conjunction $f \wedge(a, b) = a \wedge b$, where $a, b \in \{0, 1\}$ are Boolean variables denoting the inputs of Alice and Bob, respectively. This is sometimes called the *marriage proposal problem* since one can interpret the input of each party as a declaration if she/he wants to marry the other one. More precisely, suppose $a = 1$ if and only if Alice wants to marry Bob, and $b = 1$ if and only if Bob wants to marry Alice. In this case $f \wedge(a, b) = 1$ if and only if *both* parties want to marry each other, and hence, if for example, $b = 0$, then Bob after learning the output of the function has no information about Alice's input. Therefore, the privacy of Alice is protected.

One can generalize this example and consider the

set-intersection problem. Here Alice and Bob have sets A and B as their inputs and the output is equal to $f_{\cap}(A, B) = A \cap B$. For example, think of A and B as sets of e-mail addresses in Alice's and Bob's contact lists—then the output $f_{\cap}(A, B)$ is the list of the contacts that they have in common. The security here means that: (1) the parties do not learn about each other's input more than they can deduce from their own input and the output, and (2) a malicious party cannot cause the result to be incorrect (e.g., a corrupt Alice cannot falsely make Bob think that some e-mail address is in her contact list). For this example, condition (1) means that for every $a \notin A$, Alice should obtain no information if a is in B (and symmetrically for Bob).

General results and the lack of “fairness.” The above examples can be generalized in several ways. First of all, one can consider protocols executed among groups of parties of size larger than two (hence the name MPCs, as opposed to the *two-party* examples above). For example, a multiparty coin-tossing protocol is specified exactly as the two-party one, except that the number of the participants is larger than two.

Second, one can consider more complicated functions than the ones described above. It was shown in Goldreich et al.¹⁴ that for any efficiently computable function f (including “randomized” functions like the one in the coin-tossing example), there exists an efficient protocol that securely computes it, assuming the existence of trapdoor permutations (which is a well-established assumption, widely believed to hold). If a minority of the parties is malicious (i.e., does not follow the protocol), then the protocol always terminates, and the output is known to each honest participant. However, if more than half of the parties are malicious, then the malicious parties can terminate the protocol after learning the output, preventing the honest parties from learning it. Note that in case of two-player protocols, it makes no sense to assume that the majority of the players is honest, as this would simply mean that none of the players is malicious. This problem is visible in the coin-tossing example above, as each party can refuse to open her commitment after she learned what was the bit of the other party. In some cases, this is not a problem since the parties can agree that refusing to open the commitment is equivalent to losing the game.

However, it turns out⁹ that in general this problem, called the lack of *fairness*, is unavoidable. Hence, two-party protocols in general do not provide complete fairness.

Why are the MPC not widely used over the Internet? Since the introduction of MPCs there has been a significant effort to make these protocols efficient^{4,10,16} and sometimes even to use them in the real-life applications such as the online auctions.⁷ On the other hand, perhaps surprisingly, the MPCs have not been used in many other areas where seemingly they would fit perfectly. One prominent example is Internet gambling: it may be intriguing that currently gambling over the Internet is done almost entirely with the help of websites that play the roles of “trusted parties,” instead of using a cryptographic coin-flipping protocol to eliminate the need for trust. This situation is clearly unsatisfactory from the security point of view, especially since in the past, there were cases when the operators of these sites abused their privileged position for their own financial gain.¹⁸ Hence, it may look like the multiparty techniques that eliminate the need for a trusted party would be a perfect

replacement for the traditional gambling sites. An additional benefit would be a reduced cost of gambling since gambling sites typically charge fees for their service.

In our opinion, there are at least two main reasons why MPCs are not used for online gambling. The first reason is that multiparty protocols do not provide fairness in case there is no honest majority among the participants. Consider, for example, a simple two-party lottery based on the coin-tossing protocol: the parties first compute a random bit b , if $b = 0$, then Alice pays \$1 to Bob, if $b = 1$, then Bob pays \$1 to Alice, and if the protocol did not terminate correctly, then the parties do not pay any money to each other. In this case, a malicious party, say Alice, could prevent Bob from learning the output if it is equal to 0, making 1 the only possible output of a protocol. This means that two-party coin tossing is not secure in practice. More generally, multiparty coin tossing would work only if the majority is honest, which is not a realistic assumption in the fully distributed Internet environment, for instance, *sybil* attacks¹¹ allow one malicious party to create and control several “fake” identities, easily obtaining the “majority” among the participants.

The second reason is even more fundamental, as it comes directly from the inherent limitations of the MPC security definition: such protocols take care only of the security of the computation and are not “responsible” for ensuring that the users provide the “real” input to the protocol and that they respect the output.

Consider, for example, the marriage proposal problem: it is clear that there is no technological way to ensure that the users honestly provide their input to the trusted party. Nothing prevents one party, say Bob, from lying about his feelings and setting $b = 1$ to learn Alice’s input a . Similarly, forcing both parties to respect the outcome of the protocol and indeed marry cannot be guaranteed in a cryptographic way.

This problem is especially important in the gambling applications: even in the simplest “two-party lottery” example described above, there exists no cryptographic method to force the loser to transfer the money to the winner.

One pragmatic solution to this problem, both in the digital and the nondigital world, is to use the concept of “reputation”: a party caught cheating (i.e., providing the wrong input or not respecting the outcome of the game) damages her reputation and next time may have trouble finding another party willing to gamble with her. Reputation systems have been constructed and analyzed in several papers.¹⁹ However, they seem too cumbersome to use in many applications, one reason being that it is unclear how to define the reputation of new users if users are allowed to pick new names whenever they want.¹²

Another option is to exploit the fact that the financial transactions are done electronically. One could try to “incorporate” the final transaction (transferring \$1 from the loser to the winner) into the protocol, in such a way that the parties learn who won the game only when the transaction has already been performed. It is unfortunately not obvious how to do it within the framework of the existing electronic cash systems. Obviously, since the parties do not trust each other, we cannot accept solutions where the winning party learns the credit card number or the account password of the loser. One possible solution would be to design a multiparty protocol that

simulates, in a secure way, a simultaneous access to all the online accounts of the participants and executes a wire transfers in their name. Even if theoretically possible, this solution is very hard to implement in real life, especially since the protocol would need to be adapted to several banks used by the players (and would need to be updated whenever they change).

The main contribution of this paper is the introduction of a new paradigm, which we call “MPC protocols on Bitcoin,” that provides a solution to both of the problems described above: the lack of fairness and the lack of the link between “real life” and the result of the cryptographic computation. We describe our solution in Section 1.1.

1.1. Our contribution

We study how to do “MPCs on Bitcoin.” First of all, we show that the Bitcoin system provides an attractive way to construct a version of “timed commitments,”^{8, 13} where the committer has to reveal his secret within a certain time frame or pay a fine. This, in turn, can be used to obtain fairness in certain multiparty protocols. Hence, it can be viewed as an “application of Bitcoin to MPCs.”

What is probably more interesting is our second idea, which in some sense inverts the previous one by showing an “application of the MPCs to Bitcoin,” namely we introduce a concept of multiparty protocols that work directly on Bitcoin. As explained above, the standard definition of MPCs guarantees only that the protocol performs the computation securely, but ensuring that the inputs are correct and the parties do not interrupt the protocol execution is beyond the scope of the security definition. Our observation is that the Bitcoin system can be used to go beyond this standard definition, by constructing protocols that link the inputs and the outputs with real Bitcoin transactions. This is possible since the Bitcoin lacks a central authority, the list of transactions is public, and its syntax allows more advanced transactions than simply transferring the money.

As an instantiation of this idea, we construct protocols for secure multiparty lottery using the Bitcoin currency, without relying on a trusted authority. By “lottery,” we mean a protocol in which a group of parties initially invests some money, and at the end, one of them, chosen randomly, gets all the invested money (called the *pot*). Our protocol works in purely peer-to-peer environment and can be executed between players who are anonymous and do not trust each other. Our constructions come with a very strong security guarantee: no matter how the dishonest parties behave, the honest parties will never get cheated. More precisely, each honest party can be sure that, once the game starts, it will always terminate and will be fair.

Our main construction is presented in Section 4. Its security is obtained via *deposits*: each user is required to initially put aside a certain amount of money, which will be paid back to her once she completes the protocol honestly. Otherwise, the deposit is given to the other parties and “compensates” them for the fact that the game terminated prematurely. This protocol uses the timed commitment scheme described above. A drawback of this protocol is that the deposits need to be relatively large, especially if the protocol is executed among larger groups of players. More precisely, to achieve security the deposit of each player

needs to be $N(N - 1)$ times the size of the bet, where N is the number of players. For the two-party case, this simply means that the deposit is twice the size of the bet.

The only cost that the participants need to pay in our protocols is Bitcoin transaction fees. Most Bitcoin transactions are currently free. However, the participants of our protocols need to make a small number of nonstandard transactions (the so-called “strange transactions,” see Section 2), for which there is usually some small fee (currently around $0.0001 \text{ ₿} \approx \$0.04$).^b To keep the exposition simple, we present our results assuming that the fees are zero. For the sake of simplicity, we also assume that the bets in the lotteries are equal to 1 ₿. It should be straightforward to see how to generalize our protocols to other values of the bets.

Our constructions are based on the coin-tossing protocol explained above. We managed to adapt this protocol to our model, without the need to modify the current Bitcoin system. We do not use any generic methods like MPC or zero-knowledge compilers, and hence our protocols are very efficient. The only cryptographic primitives that we use are commitment schemes, implemented using hash functions (which are standard Bitcoin primitives). Our protocols rely strongly on the advanced features of the Bitcoin (in particular, the so-called “transaction scripts,” and “time-locks”). Because of the lack of space, we only sketch the formal security definitions. We executed our transactions on the real Bitcoin. We provide a description of these transactions and a reference to them in the Bitcoin block chain.^c

1.2. Independent and subsequent work

Usage of Bitcoin to create a secure and fair two-player lottery has been independently proposed by Back and Bentov.³ We provide a detailed comparison between their protocol and ours in the extended version of this paper.

In the subsequent work,^{1,2} we show how to extend the ideas from this paper to construct a fair two-party protocol for any functionality, in such a way that the execution of this protocol has “financial consequences.” More precisely, in the first paper,¹ we show how to solve this problem under the assumption that the Bitcoin transactions are nonmalleable (see Andrychowicz et al.^{1,2} for more on this notion), and in Andrychowicz et al.,² we show how to modify the protocol from Andrychowicz et al.¹ to obtain a protocol that is secure in the current version of Bitcoin. Some alternative ideas for obtaining fairness in the multiparty protocols were developed independently by Bentov and Kumaresan.^{5,15}

1.3. Applications and future work

Although, as argued in the extended version of this paper, it may actually make economic sense to use our protocols in practice, we view gambling mostly as a motivating example for introducing a concept that can be called “MPCs on Bitcoin,” and which will hopefully have other applications. One example of a task that can be implemented using our techniques is

a protocol for selling secret information for Bitcoins. Imagine Alice and Bob know a description of a set X containing some valuable information. For example, X can contain some sensitive data that is hard to find (say: personal data signed by a secret key of some public authority). Alice knows some subset A of X and Bob knows a subset B of X . Their goal is to sell to each other the elements of $A \cup B$ in such a way that they will pay to each other only for the elements they did not know in advance. In other words, Alice will pay to Bob $(|B \setminus A| - |A \setminus B|) \text{ ₿}$ (if this value is negative, then Bob will pay to Alice its negation). Without the MPC techniques, it is not clear how to do it: whenever Alice reveals to Bob some element $a \in A$, Bob can always claim that he already knew a . Moreover, even if MPC techniques are used, Alice has no way to force Bob to pay her the money (and vice-versa). Our tools (developed in the subsequent papers mentioned in Section 1.2) solve this problem: we can design a protocol that transfers exactly the right sum of Bitcoins, and moreover, this happens if and only if both parties really learned the output of the computation!

The above example can be generalized in several different ways. For example, the output can go only to one party (say: Alice), and the condition for the information that Alice is willing to pay for can be much more complicated. For example, Alice can be an intelligence agency that has a special secret function g that specifies what is the value of a given information (for some set of inputs g can even output 0). Then Bob can try to “sell” his information x to Alice setting some minimal value v that it is worth according to him. The protocol would compute $g(x)$ and check if $g(x) \geq v$ —if yes, then Alice would learn x and pay v to Bob, and otherwise Alice would learn nothing (and Bob would earn 0).

Finally, let us remark that our protocols can potentially be used for malicious purposes. For example, consider ransomware that encrypts the hard disk of the victim’s machine and promises to provide a decryption key only if the victim pays a ransom. Currently, such malicious programs have no way to prove that they will really send the right key if the ransom is paid. With our techniques, one can make delivery of this key secure (in the sense that the payment happens only if the key really decrypts the disk). Another potential risk is attacks on online voting schemes: it is well-known that if these schemes are not receipt-free, then the adversary can buy votes. Our techniques can make such attacks easier, as they eliminate the need of the vote seller to trust the vote buyer.

2. A SHORT DESCRIPTION OF BITCOIN

Bitcoin¹⁷ works as a peer-to-peer network in which the participants jointly emulate a central server that controls the correctness of the transactions. In this sense, it is similar to the concept of the MPC protocols. Recall that, as described above, a fundamental problem with the traditional MPCs is that they cannot provide fairness if there is no honest majority among the participants, which is particularly difficult to guarantee in the peer-to-peer networks where the sybil attacks are possible. The Bitcoin system overcomes this problem in the following way: the honest majority is defined in terms of the “majority of computing power.” In other words, in order to break the system, the adversary needs to control machines whose total computing power is comparable with

^b We use “₿”; for the Bitcoin currency symbol.

^c For example the main transaction (*Compute*) of the three-party lottery is available here: blockchain.info/tx/540d816bd57300209754dd36ffcec1d669bd2068641844783451cd3ef32c8aa4.

the combined computing power of all the other participants of the protocol. Hence, for example, the sybil attack does not work, as creating a lot of fake identities in the network does not help the adversary. In a moment we will explain how this is implemented, but let us first describe the functionality of the trusted party that is emulated by the users.

One of the main problems with digital currencies is potential double spending: if coins are just strings of bits, then the owner of a coin can spend it multiple times. Clearly, this risk could be avoided if the users had access to a trusted ledger with the list of all the transactions. In this case, a transaction would be considered valid only if it is posted on the ledger. For example, suppose the transactions are of a form: “user A transfers x Bitcoins to user B .” In this case, each user can verify if A really has x Bitcoins (i.e., she received it in some previous transactions) and she did not spend it yet. The functionality of the trusted party emulated by the Bitcoin network does precisely this: it maintains a full list of the transactions that happened in the system. The format of Bitcoin transactions is in fact more complex than in the example above. Since it is of a special interest for us, we describe it in more detail in Section 2.1. However, for the sake of simplicity, we omit the features of Bitcoin that are not relevant to our work such as transaction fees or how the coins are created.

The Bitcoin ledger is in fact a chain of *blocks* (each block contains transactions) that all the participants are trying to extend. The parameters of the system are chosen in such a way that an extension happens on average once each 10 min. The idea of the block chain is that the longest chain C is accepted as the proper one and appending a new block to the chain takes nontrivial computation. As extending the block chain or creating a new one is very hard, all users will use the same, original block chain. Speaking in more detail, this construction prevents double spending of transactions. If a transaction is contained in a block B_i and there are several new blocks after it, then it is infeasible for an adversary with less than a half of the total computational power of the Bitcoin network to revert it—he would have to mine a new chain C' bifurcating from C at block B_{i-1} (or earlier), and C' would have to be longer than C . The difficulty of that grows exponentially with number of new blocks on top of B_i . In practice, transactions need 10–20 min for reasonably strong confirmation and 60 min (6 blocks) for almost absolute certainty that they are irreversible.

To sum up, when a user wants to pay somebody in Bitcoins, he creates a transaction and broadcasts it to other nodes in the network. They validate this transaction, send it further, and add it to the block they are mining. When some node solves the mining problem, it broadcasts its block to the network. Nodes obtain a new block, validate transactions in it and its hash, and accept it by mining on top of it. The presence of the transaction in the block is a confirmation of this transaction, but some users may choose to wait for several blocks to get more assurance. In our protocols, we assume that there exists a maximum delay $T_{\max}$ between broadcasting the transaction and its confirmation and that every transaction once confirmed is irreversible.

2.1. Bitcoin transactions

In contrast to the classical banking system, Bitcoin is based

on *transactions* instead of *accounts*. A user A has some Bitcoins if in the system there are unredeemed transactions for which he is a recipient. Each transaction has some value (number of Bitcoins which is being transferred) and a recipient *address*. An address is simply a public key pk . Normally, every such key has a corresponding private key sk known only to one user—that user is the owner of the address pk . The private key is used for signing (authorizing) the transactions, and the public key is used for verifying the signatures. Each user of the system needs to know at least one private key of some address, but this is simple to achieve since the pairs (sk, pk) can be easily generated offline. We will frequently denote the key pairs using capital letters (e.g., A) and refer to the private key and the public key of A by $A.sk$ and $A.pk$, respectively.

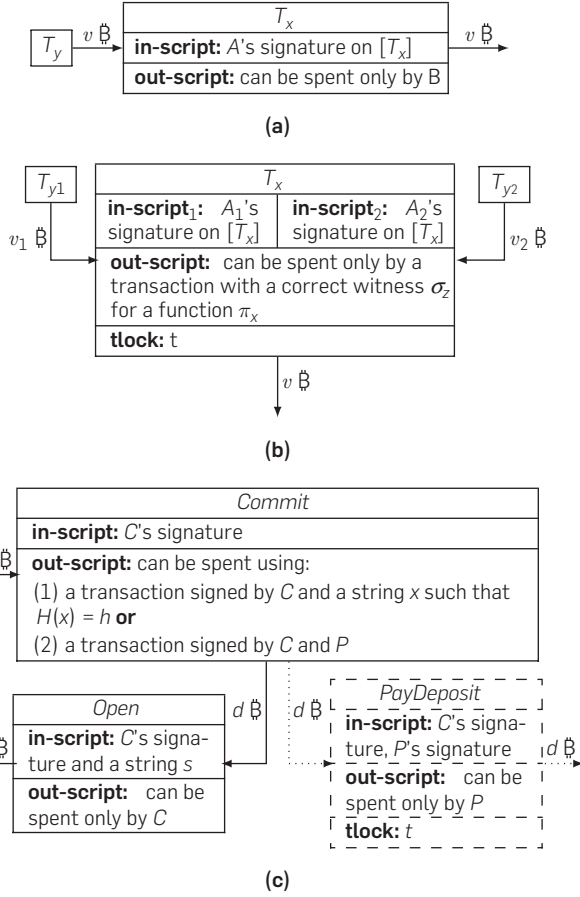
Simplified version. We first describe a simplified version of Bitcoin and then show how to extend it to obtain the description of the real Bitcoin. Let $(A.sk, A.pk)$ and $(B.sk, B.pk)$ be the key pairs belonging to users A and B , respectively. In our simplified view, a transaction describing the fact that an amount v (called the *value* of a transaction) is transferred from an address $A.pk$ to an address $B.pk$ has the form $T_x = (y, v, B.pk, \text{sig})$, where y is an index of a previous transaction T_y , and sig is a signature computed using sender's secret key $A.sk$ on the whole transaction excluding the signature itself (i.e., on $(y, v, B.pk)$). We say that $B.pk$ is the recipient of T_x , and that the transaction T_y is an *input* of the transaction T_x , or that T_y is *redeemed* by T_x . More precisely, the meaning of T_x is that the amount v of money transferred to $A.pk$ in transaction T_y is transferred further to $B.pk$. The transaction T_x is valid only if (1) $A.pk$ was a recipient of the transaction T_y , (2) the value of T_y was equal to v , (3) the transaction T_y has not been redeemed earlier, and (4) the signature of A is correct. All these conditions can be verified publicly.

We will present the transactions as boxes. The redeeming of transactions will be indicated with arrows with the value of the transaction. For example, a transaction $T_x = (y, v, B.pk, \text{sig})$, which transfers v Bitcoins from A to B , is depicted in Figure 1(a).

The first important generalization of this simplified system is that a transaction can have several “inputs” meaning that it can accumulate money from several past transactions $T_{y_1}, \dots, T_{y_\ell}$. Let $A_1, \dots, A_\ell$ be the respective key pairs of the recipients of those transactions. Then a multiple-input transaction has the following form: $T_x = (y_1, \dots, y_\ell, v, B.pk, \text{sig}_1, \dots, \text{sig}_\ell)$, where each sig_i is a signature computed using key $A_i.sk$ on the whole message excluding the signatures. The result of such transaction is that $B.pk$ gets the amount v , provided it is equal to the sum of the values of the transactions $T_{y_1}, \dots, T_{y_\ell}$. This happens only if *none* of these transactions has been redeemed before, and *all* the signatures are valid. Each transaction can also have several outputs, which is a way to divide money between several users or get change, but we do not use this feature in our protocols.

A more detailed version. The real Bitcoin system is significantly more sophisticated than what is described above. First of all, there are some syntactic differences, the most important for us being that each transaction T_x is identified not by its index, but by the hash of the whole transaction, $H(T_x)$. Hence, from now on, we will assume that $x = H(T_x)$. Moreover, each transaction can have a *time-lock* t that tells at what time the

Figure 1. (a) A standard transaction transferring v Bitcoins from A to B, (b) a nonstandard transaction with two inputs and a time-lock, (c) the CS protocol.



transaction becomes valid. In this case, we have: $T_x = (y_1, \dots, y_b, v, B.pk, t, \text{sig}_1, \dots, \text{sig}_l)$. Such a transaction becomes valid only if the time t is reached and all the conditions mentioned earlier are satisfied. Before the time t , the transaction T_x cannot be used (it will not be included into any block before the time t).

The main difference is, however, that in the real Bitcoin, the users have much more flexibility in defining the condition on how the transaction can be redeemed. Consider for a moment the simplest transaction where there is just one input and no time-locks. Recall that in the simplified system described above, in order to redeem a transaction the recipient $A.pk$ had to produce another transaction T_x signed with his private key $A.sk$. In the real Bitcoin, this is generalized as follows: each transaction T_y comes with a description of a function (called *output-script*) π_y whose output is Boolean. The transaction T_x redeeming the transaction T_y is valid if π_y evaluates to true on input T_x . In case of standard transactions, π_y is a function that treats T_x as a pair (a message m_x , a signature σ_x) and checks if σ_x is a valid signature on m_x with respect to the public key $A.pk$. However, much more general functions π_y are possible. Going further into details, a transaction looks as follows: $T_x = (y, \pi_x, v, \sigma_x)$, where $[T_x] = (y, \pi_x, v)$ is called the *body* of T_x and σ_x is an *input-script*—a witness that is used to make the script π_y evaluate to true on T_x (in standard transactions σ_x

is a signature of a sender on $[T_x]$). The scripts are written in the Bitcoin scripting language, which is a stack based, not Turing-complete language (there are no loops in it). It provides basic arithmetical operations on numbers, operations on stack, if-then-else statements, and some cryptographic functions like calculating a hash function or verifying a signature. The generalization to multiple-input transactions with time-locks is straightforward: a transaction has the form $T_x = (y_1, \dots, y_b, \pi_x, v, t, \sigma_1, \dots, \sigma_l)$, where the body $[T_x]$ is equal to $(y_1, \dots, y_b, \pi_x, v, t)$, and it is valid if (1) time t is reached, (2) every $\pi_i([T_x], \sigma_i)$ evaluates to true, where each π_i is the output script of the transaction T_{y_i} , (3) none of these transactions has been redeemed before, and (4) the sum of values of transactions T_{y_i} is equal to v .

A box representation of a general transaction with two inputs, $T_x = (y_1, y_2, \pi_x, v, t, \sigma_1, \sigma_2)$, is depicted in Figure 1(b).

The most common type of transactions is transactions without time-locks or any special script: the input script is a signature, and the output script is a signature verification algorithm. We will call them *standard transactions*, and the address against which the verification is done will be called the *recipient* of a transaction. Currently, some miners accept only standard transactions (although the nonstandard transactions are also correct according to the Bitcoin description). We believe that in the future accepting the nonstandard transactions will become common. This is important for our applications since our protocols rely heavily on nonstandard transactions.

3. BITCOIN-BASED TIMED COMMITMENT SCHEME

We start with constructing a Bitcoin-based timed commitment scheme. Commitment schemes were already described in Section 1. A simple way to implement a commitment is to use a cryptographic hash function H . To commit to a secret $s \in \{0, 1\}^*$, the committer chooses a random string $r \in \{0, 1\}^{128}$ and sends to the receiver $c = H(s||r)$ (where “||” denotes concatenation). To open the commitment, the committer sends (s, x) and the receiver verifies that $H(s||x) = c$.

Although incredibly useful in many applications, standard commitment schemes suffer from the following problem (already described in the introduction): there is no way to force the committer to reveal his secret s , and, in particular, if he aborts before the *Open* phase starts, then s remains secret. Bitcoin offers an attractive way to deal with this problem. Namely, using the Bitcoin system, one can force the committer to back his commitment with some money, called the *deposit*, that will be given to the recipient if he refuses to open the commitment within some time t agreed by both parties. More precisely, during the commitment phase, the committer makes a deposit in Bitcoins. He will get this deposit back if he opens the commitment before the time t . Otherwise, this deposit will be automatically given to the recipient.

3.1. Construction

Our construction of the Bitcoin-Based Timed Commitment Scheme (CS) will be based on the simple commitment scheme described earlier. The hash function used in Bitcoin is SHA256 and in our protocols we also use it because it can be used in the Bitcoin scripting language. But for clarity, we will still denote it by H in the descriptions of the protocols. Additionally,

we assume that the secret is already padded with random bits so we do not add them or strip them off in our description. In fact, we will later use the CS protocol to commit to long random strings so in that case padding is not necessary.

The basic idea of our protocol is as follows. In the commitment phase, the committer creates a transaction *Commit* with some agreed value d , which serves as the deposit. The only way to redeem the deposit is to post another transaction *Open*, which reveals the secret s . The transaction *Commit* is constructed in such a way that the *Open* transaction has to open the commitment, that is, reveal the secret value s . This means that the money of the committer is “frozen” until he reveals s . To allow the recipient to claim the deposit if the committer does not open the commitment within a certain time period, we also require the committer to send to the recipient a transaction *PayDeposit* that can redeem *Commit* if time t passes.

Technically, it is done by constructing the output script of the transaction *Commit* in such a way that the redeeming transaction has to provide either C’s signature and the secret s (which will therefore become publicly known as all transactions are publicly visible) or signatures from both C and R. After broadcasting the transaction *Commit*, the committer creates the transaction *PayDeposit*, which sends the deposit to the recipient and has a time-lock t . The committer signs it and sends it to the recipient. After receiving *PayDeposit*, the recipient checks if it is correct and adds his own signature to it. After that he can be sure that either the committer will open his commitment by the time t or he will be able to use the transaction *PayDeposit* to claim the d deposit.

The graph of transactions in this protocol is depicted in Figure 1(c). The full description of the protocol can be found in the extended version of this paper.

4. THE LOTTERY PROTOCOL

As discussed in Section 1, as an example of an application of the “MPCs on Bitcoin” concept, we construct a protocol for a lottery executed among two parties: Alice (A) and Bob (B). We say that a protocol is a *fair lottery protocol* if it is *correct* and *secure*.

To define correctness assume that both parties are following the protocol and the communication channel between them is secure (i.e., it reliably transmits the messages between the parties without delay). We assume also that before the protocol starts, the parties have enough funds to play the lottery, including both their stakes (for simplicity we assume that the stakes are equal 1 β) and the money for deposits, because in the protocol we will use the commitment scheme from Section 3. If these assumptions hold, a correct protocol must ensure that at the end of the protocol one party, chosen with uniform probability, has to get the whole pot consisting of both stakes and the other party loses her stake. Additionally, both parties have to get their deposits back.

To define security, look at the execution of the protocol from the point of view of one party, say A (the case of the other party is symmetric) assuming that she is honest. Obviously, A has no guarantee that the protocol will terminate successfully, as the other party can leave the protocol before it is completed. What is important is that A should be sure that she will not

lose money because of this termination, for example, the other party should not be allowed to terminate the protocol after he learned that A won. This is formalized as follows: we define the *payoff* of A in the execution of the protocol to be equal to the difference between the money that A invested and the money that she has after the execution of the protocol. We say that the protocol is *secure* if for any strategy of an adversary that controls the network and corrupts one party, the expected payoff of the other, honest party is not negative. We also note that, of course, a dishonest participant can always terminate at a very early stage when she does not know who is the winner—it does not change the payoff of the honest party.

4.1. The protocol

Our protocol is built on top of the classical coin-tossing protocol of Blum⁶ described in Section 1. As already mentioned, this protocol does not directly work for our application, so we need to adapt it to Bitcoin. In particular, in our solution creating and opening the commitments are done by the transactions’ scripts using (double) SHA-256 hashing. After choosing a random bit b_p , the party $P \in \{A, B\}$ chooses a string s_p sampled uniformly random from $\{0, 1\}^{128+b_p}$, that is, the set of strings of length 128 or 129 bits, according to the value of b_p . Party P then commits to s_p using a timed commitment. The winner is determined by the *winner choosing function* f , defined as follows: $f(s_A, s_B) = A$ if $|s_A| = |s_B|$ and B, otherwise, where s_A and s_B are the secret strings chosen by the parties and $|s_p|$ is the length of s_p in bits. It is easy to see that as long as one of the parties draws their bit b_p uniformly, then the output of $f(s_A, s_B)$ is also uniformly random (provided the parties can only choose the strings s_A and s_B to be of length 128 or 129).

First attempt. We start with presenting a naive and insecure construction of the protocol, and then show how it can be modified to obtain a secure scheme. Both parties announce their public keys to each other. Alice and Bob also draw at random their secret strings s_A and s_B (respectively) as mentioned earlier and they exchange the hashes $h_A = H(s_A)$ and $h_B = H(s_B)$. If $h_A = h_B$, then the players abort the protocol.^d Both parties broadcast their input transactions and send to the other party the links to their appearance in the block chain. If at any point later a party $P \in \{A, B\}$ realizes that the other party is cheating, then the first thing P will do is to “take the money and run,” that is, post a transaction that redeems the input transaction. We will call it “halting the execution.” This can clearly be done as long as the input transaction has not been redeemed by some other transaction. In the next step, one of the parties constructs a transaction *Compute* defined as follows:

Compute	
in-script₁ : A’s signature	in-script₂ : B’s signature
out-script : can be spent using: (1) strings x_A and x_B of length 128 or 129 s.t. $H(x_A) = h_A$, $H(x_B) = h_B$ and (2) X ’s signature, where X is the winner (i.e., $X = f(x_A, x_B)$)	

^d We would like to thank Iddo Bentov and Ranjit Kumaresan, and independently David Wagner, for pointing out to us that this step is needed. It protects from the *copy attack*: A waits until B commits with his hash h_B and then she commits with the same hash. During the opening phase, A again waits until B reveals his secret s_B and then she reveals the same secret. By doing this A always wins since $f(s_A, s_B) = A$.

Note that the body of *Compute* can be computed from the publicly available information. Hence, this construction can be implemented as follows: first one of the players, say, Bob computes the body of *Compute* and sends his signature on it to Alice. Alice computes the body, adds both signatures to it, and broadcasts the entire transaction *Compute*.

The output script of *Compute* is tricky. To make it evaluate to true on *body*, one needs to provide as “witnesses” the signature of a party P and strings x_A, x_B , where x_A and x_B are the preimages of h_A and h_B (with respect to H). The collision-resistance of H implies that x_A and x_B have to be equal to s_A and s_B (resp.). Hence, it can be satisfied only if the winner choosing function f evaluates to P on input (s_A, s_B) . Since only party P knows her private key, only she can later provide a signature that would make the output script evaluate to true.

Before *Compute* appears on the block chain, each party P can “change her mind” and redeem her input transaction, which would make the transaction *Compute* invalid. As we said before, it is ok for us if one party interrupts the coin-tossing procedure as long as she had to decide about doing it *before* she learned that she lost. Hence, Alice and Bob wait until the transaction *Compute* becomes confirmed before they proceed to the step in which the winner is determined. This final step is simple: Alice and Bob just broadcast s_A and s_B , respectively. Now: if $f(s_A, s_B) = A$, then Alice can redeem the transaction *Compute* in a transaction *ClaimMoney_A* constructed as:

<i>ClaimMoney_A</i>	
in-script:	strings s_A and s_B and A's signature
out-script:	can be spent only by A

On the other hand, Bob cannot redeem *Compute*, as the condition $f(s_A, s_B) = B$ evaluates to false. Symmetrically: if $f(s_A, s_B) = B$, then only Bob can redeem *Compute* by an analogous transaction *ClaimMoney_B*.

This protocol is obviously correct. It may also look secure, as it is essentially identical to Blum's protocol described before (with a hash function used as the commitment scheme). Unfortunately, it suffers from the following problem: there is no way to guarantee that the parties always reveal s_A and s_B . In particular: one party, say, Bob, can refuse to send s_B *after* he learned that he lost (i.e., that $f(s_A, s_B) = A$). As his money is already “gone” (his input transaction has already been redeemed in transaction *Compute*), he cannot gain anything, but he might do it just because of sheer nastiness. Unfortunately, in a purely peer-to-peer environment, with no concept of a “reputation,” such behavior can happen, and there is no way to punish it. This is exactly why we need to use the Bitcoin-based commitment scheme from Section 3.

The secure version of the scheme. The general idea behind the SecureLottery protocol is that each party first commits to her inputs, using the Bitcoin-based timed commitment scheme, instead of the standard commitment scheme. Recall that the CS protocol can be opened by sending a value s , and this opening is verified by checking that s has required length (either 128 or 129) and hashes to a value h sent by the committer in the commitment phase. So, Alice executes the CS protocol acting as the committer and Bob as a receiver. Let s_A and h_A be the variables s and h created this way. Symmetrically, Bob

executes the CS protocol acting as the committer, and Alice being the receiver, and the corresponding variables are s_B and h_B . Once both commitment phases are executed successfully (recall that this includes receiving by each party the signed *PayDeposit* transaction), the parties proceed to the next steps, which are exactly as before: first, each of them broadcasts an input transaction. Once these transactions are confirmed, they create the *Compute* transaction in the same way as before, and once it appears on the block chain, they open the commitments. The only difference is that, since they used the CS commitment scheme, they can now “punish” the other party if she did not open her commitment by the time t and claim their deposit. On the other hand, each honest party is always guaranteed to get her deposit back, hence she does not risk anything by investing this money at the beginning of the protocol. The graph of transactions in this protocol is presented in Figure 2.

We also need to comment about the choice of the parameters: t : the time when deposit become available to the receiver and d : the value of the deposit. Our protocol consists of four “rounds” of transactions—in each round, parties wait for the confirmation of all the transactions from this round before proceeding to the next round. Thus, the correct execution of the protocol always terminates within time $4 \cdot T_{\max}$, where $T_{\max}$ is the maximal time needed for a transaction to be confirmed. Because of that we can safely set t to be the start time of the protocol plus $5 \cdot T_{\max}$.

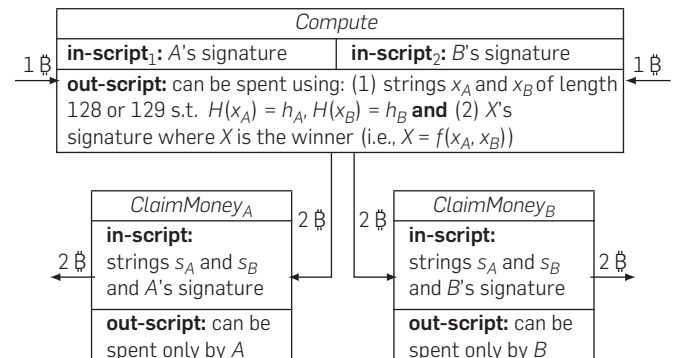
The parameter d should be chosen in such a way that it will fully compensate to each party the fact that the other player aborted. That means that for a two-player lottery, each player should make a deposit equal to two stakes. This way if one party aborts the protocol, then the other party may lose her stake worth 1 ฿ , but she gets a deposit of value 2 ฿ , so as a result of the protocol executions she earns 1 ฿ , what is never worse for her than executing the protocol to the very end.

The complete description of this protocol can be found in the extended version of this paper, where we also show how to generalize it to N parties. In our multiparty solution, the total amount of money invested in the deposit by each player has to be equal to $N(N-1)$ ฿ . In real-life this would be ok probably for small groups $N = 2, 3$, but not for the larger ones.

Acknowledgments

We would like to thank Iddo Bentov and Ranjit Kumaresan for fruitful discussions and for pointing out an error in a

Figure 2. The SecureLottery protocol.



previous version of our lottery. We are also very grateful to David Wagner for carefully reading our paper and for several useful remarks. 

References

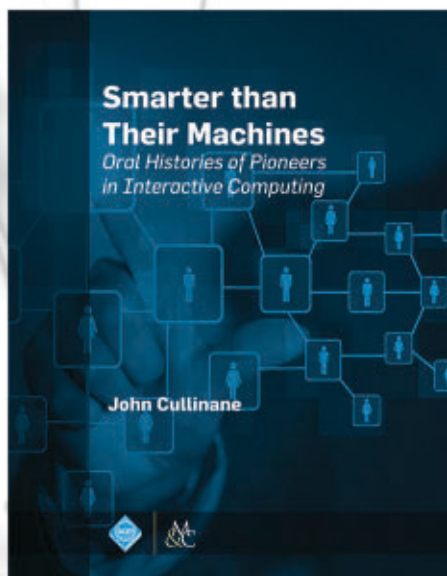
- Andrychowicz, M., Dziembowski, S., Malinowski, D., Mazurek, Ł. Fair two-party computations via bitcoin deposits. In *1st Workshop on Bitcoin Research* (Christ Church, Barbados, March 7, 2014), Springer, Berlin, Germany, 105–121.
- Andrychowicz, M., Dziembowski, S., Malinowski, D., Mazurek, Ł. On the malleability of bitcoin transactions. In *2nd Workshop on Bitcoin Research* (San Juan, Puerto Rico, January 30, 2015), Springer, Berlin, Germany.
- Back, A., Bentov, I. Note on fair coin toss via bitcoin, 2013. <http://www.cs.technion.ac.il/~iddo/cointossBitcoin.pdf>.
- Ben-David, A., Nisan, N., Pinkas, B. FairplayMP: A system for secure multi-party computation. In *ACM CCS 08: 15th Conference on Computer and Communications Security* (Alexandria, VA, October 27–31, 2008), ACM, NY, 257–266.
- Bentov, I., Kumaresan, R. How to use bitcoin to design fair protocols. In *Advances in Cryptology – CRYPTO, 2014. Part II* (Santa Barbara, CA, August 17–21, 2014), Springer, Berlin, Germany, 421–439.
- Blum, M. Coin flipping by telephone. In *Advances in Cryptology – CRYPTO 81* (Santa Barbara, CA, 1981), U.C. Santa Barbara, Department of Electrical and Computer Engineering, 11–15.
- Bogetoft, P., et al. Secure multiparty computation goes live. In *FC 2009: 13th International Conference on Financial Cryptography and Data Security* (Accra Beach, Barbados, February 23–26, 2009), Springer, Berlin, Germany, 325–343.
- Boneh, D., Naor, M. Timed commitments. In *Advances in Cryptology – CRYPTO 2000* (Santa Barbara, CA, August 20–24, 2000), Springer, Berlin, Germany, 236–254.
- Cleve, R. Limits on the security of coin flips when half the processors are faulty. In *Proceedings of the 18th Annual ACM Symposium on Theory of Computing*, STOC '86 (Berkeley, CA, May 28–30, 1986), ACM, NY, 364–369.
- Damgård, I., et al. Practical covertly secure MPC for dishonest majority — Or: Breaking the SPDZ limits. In *ESORICS 2013: 18th European Symposium on Research in Computer Security* (Egham, UK, September 9–13, 2013), Springer, Berlin, Germany, 1–18.
- Douceur, J.R. The sybil attack. In *First International Workshop on Peer-to-Peer Systems*, IPTPS '01, 2002.
- Friedman, E.J., Resnick, P. The social cost of cheap pseudonyms. *J. Econ. Manage. Strat.* 10 (2000), 173–199.
- Garay, J.A., Jakobsson, M. Timed release of standard digital signatures. In *FC 2002: 6th International Conference on Financial Cryptography* (Southampton, Bermuda, March 11–14, 2003), Springer, Berlin, Germany, 168–182.
- Goldreich, O., Micali, S., Wigderson, A. How to play any mental game or A completeness theorem for protocols with honest majority. In *19th Annual ACM Symposium on Theory of Computing* (New York City, NY, May 25–27, 1987), ACM, NY, 218–229.
- Kumaresan, R., Bentov, I. How to use bitcoin to incentivize correct computations. In *ACM CCS 2014* (Scottsdale, AZ, November 3–7, 2014), ACM, NY, 30–41.
- Malkhi, D., Nisan, N., Pinkas, B., Sella, Y. Fairplay – A secure two-party computation system. In *13th Conference on USENIX Security Symposium*, SSYM'04 (San Diego, CA, August 9–13, 2004), USENIX Association, 287–302.
- Nakamoto, S. Bitcoin: A peer-to-peer electronic cash system. The Cryptography Mailing List, 2008.
- Post, T.W. Cheating scandals raise new questions about honesty, security of internet gambling. *The Washington Post* November 30, 2008.
- Resnick, P., Kuwabara, K., Zeckhauser, R., Friedman, E. Reputation systems. *Commun. ACM* 43, 12 (Dec. 2000) 45–48.
- Yao, A.C.-C. How to generate and exchange secrets (extended abstract). In *27th Annual Symposium on Foundations of Computer Science* (Toronto, ON, Canada, October 27–29, 1986), IEEE Computer Society Press, 162–167.

Marcin Andrychowicz, Stefan Dziembowski, Daniel Malinowski, and Łukasz Mazurek
([marcin.andrychowicz, stefan.dziembowski, daniel.malinowski, lukasz.mazurek]@crypto.edu.pl), Institute of Informatics, University of Warsaw, Warsaw, Poland.

Copyright held by authors. Publication rights licensed to ACM. \$15.00.



Watch the author discuss his work in this exclusive *Communications* video. <http://cacm.acm.org/videos/secure-multiparty-computations-on-bitcoin>



A personal walk down the computer industry road. BY AN EYEWITNESS.

Smarter Than Their Machines: Oral Histories of the Pioneers of Interactive Computing

is based on oral histories archived at the Charles Babbage Institute, University of Minnesota. These oral histories contain important messages for our leaders of today, at all levels, including that government, industry, and academia can accomplish great things when working together in an effective way.



ISBN: 978-1-62705-550-5 DOI: 110.1145/2663015
<http://books.acm.org>
<http://www.morganclaypoolpublishers.com/acm>

Technical Perspective

The State (and Security) of the Bitcoin Economy

By Emin Gün Sirer

SUPPOSE WE HAD a complete record of every single financial transaction that took place worldwide over a span of five years. What could we learn by studying the patterns encoded in this complex ledger?

The following paper examines this question in the context of Bitcoin, an emerging cryptocurrency that boasts a large and diverse economy, including many legitimate and some notable less-than-legal users.

Central to Bitcoin's design is the notion of a blockchain that records every single transaction between Bitcoin wallets. While the transactions recorded in the blockchain are public, the identity of the users and services behind the Bitcoin wallets are not. In stark contrast with the banking system, where creating a bank account requires identification, any Bitcoin user can create a new wallet, that is, a set of addresses that hold coins, without having to register with any authority. This provides a modicum of pseudonymity to Bitcoin users, but the question is, precisely how much anonymity does it provide, and how much can we learn about the actual entities behind the pseudonymous addresses on the blockchain.

To address this question, the authors use the graph of transactions encoded in the blockchain to identify patterns

that might link together the identities of Bitcoin users, corroborate that data with known addresses for Bitcoin services, and therefore identify the interaction patterns between users and services. To do this effectively, they examine features of Bitcoin wallets that link addresses together, and build a set of heuristics to cluster addresses that are likely to belong to a single user.

Having built a model for the various entities in the Bitcoin economy, the paper goes on to examine critical questions for any large-scale economy: How big and frequent are transactions? How is wealth distributed? Are the Bitcoin-rich like you and me, or do they exhibit qualitatively different spending patterns? Given the Bitcoin economy combines short-term inflation with long-term deflation, for how long do people hold onto their coins? How big are the various service sectors of the Bitcoin economy? And since we can examine the life cycle of any given coin, can we trace flows of interest, especially those involved in well-publicized heists? And most critically, just how anonymous can one be on the blockchain?

Practitioners of the dismal science often complain about having insufficient data about the economy. The real economy, indeed, makes it difficult to trace many kinds of transactions. This is why the emerging cryptocurrency space provides a unique and fascinating opportunity to gain insight into both the legitimate and underground uses of a currency. The paper provides a comprehensive view of the state of the Bitcoin economy at a particular point in time, one that will undoubtedly be important in building economic models and in serving as a point of comparison as the system evolves. 

Emin Gün Sirer (egs@cs.cornell.edu) is an associate professor in the computer science department at Cornell University, Ithaca, NY.

Copyright held by author.



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:
librarians.acm.org/pod**

**The emerging
cryptocurrency space
provides a unique and
fascinating opportunity
to gain insight into
both the legitimate
and underground
uses of a currency.**

A Fistful of Bitcoins: Characterizing Payments among Men with No Names

By Sarah Meiklejohn,* Marjori Pomarole, Grant Jordan, Kirill Levchenko, Damon McCoy, Geoffrey M. Voelker, and Stefan Savage

Abstract

Bitcoin is a purely online virtual currency, unbacked by either physical commodities or sovereign obligation; instead, it relies on a combination of cryptographic protection and a peer-to-peer protocol for witnessing settlements. Consequently, Bitcoin has the unintuitive property that while the ownership of money is implicitly anonymous, its flow is globally visible. In this paper we explore this unique characteristic further, using heuristic clustering to group Bitcoin wallets based on evidence of shared authority, and then using re-identification attacks (i.e., empirical purchasing of goods and services) to classify the operators of those clusters. From this analysis, we consider the challenges for those seeking to use Bitcoin for criminal or fraudulent purposes at scale.

1. INTRODUCTION

Demand for low friction e-commerce of various kinds has driven a proliferation in online payment systems over the last decade. Thus, in addition to established payment card networks (e.g., Visa and Mastercard), a broad range of the so-called “alternative payments” has emerged including eWallets (e.g., Paypal, Google Checkout, and WebMoney), direct debit systems (typically via ACH, such as eBillMe), money transfer systems (e.g., Moneygram), and so on. However, virtually all of these systems have the property that they are denominated in existing fiat currencies (e.g., dollars), explicitly identify the payer in transactions, and are centrally or quasi-centrally administered. (In particular, there is a central controlling authority who has the technical and legal capacity to tie a transaction back to a pair of individuals.)

By far the most intriguing exception to this rule is Bitcoin. First deployed in 2009, Bitcoin is an independent online monetary system that combines some of the features of cash and existing online payment methods. Like cash, Bitcoin transactions do not explicitly identify the payer or the payee: a transaction is a cryptographically signed transfer of funds from one public key to another. Moreover, like cash, Bitcoin transactions are irreversible (in particular, there is no *charge-back* risk as with credit cards). However, unlike cash, Bitcoin requires third-party mediation: a global peer-to-peer network of participants validates and certifies all transactions. Such

decentralized accounting requires each network participant to maintain the entire transaction history of the system, which even in 2012 amounted to over 3GB of compressed data. Bitcoin identities are thus *pseudo-anonymous*: while not explicitly tied to real-world individuals or organizations, all transactions are completely transparent.

This unusual combination of features has given rise to considerable confusion about the nature and consequences of the anonymity that Bitcoin provides. In particular, there is concern that the combination of scalable, irrevocable, anonymous payments would prove highly attractive for criminals engaged in fraud or money laundering. In a widely leaked 2012 Intelligence Assessment, FBI analysts make just this case and conclude that a key “advantage” of Bitcoin for criminals is that “law enforcement faces difficulties detecting suspicious activity, identifying users, and obtaining transaction records.”⁵ Similarly, in a late 2012 report on Virtual Currency Schemes, the European Central Bank opines that the lack of regulation and due diligence might enable “criminals, terrorists, fraudsters, and money laundering” and that “the extent to which any money flows can be traced back to a particular user is unknown.”⁴ Indeed, there is at least some anecdotal evidence that this statement is true, with the widely publicized Silk Road service using Bitcoin to trade in a range of illegal goods (e.g., restricted drugs and firearms). Finally, adding to this urgency is Bitcoin’s considerable growth, both quantitatively—a merchant servicer, Bitpay, announced that it had signed up over 1000 merchants in 2012 to accept bitcoins, and in November 2013 the exchange rate soared to a peak of 1000 USD per bitcoin—and qualitatively via integration with existing payment mechanisms and the increasing attention of world financial institutions. In 2012 alone, Bitinstant offered to tie users’ Bitcoin wallets to Mastercard accounts,³ Bitcoin Central partnered with the French bank Crédit Mutuel Arkéa to gateway Bitcoin into the banking system,⁸ Canada decided to tax Bitcoin transactions,² and FinCEN issued regulations on virtual currencies.⁶ Despite this background of intense interest, Bitcoin’s pseudo-anonymity has limited how much is known about how the currency is used and how Bitcoin’s use has evolved over time.

The original version of this paper is entitled “A Fistful of Bitcoins: Characterizing Payments among Men with No Names” and was published in the *Proceedings of the Internet Measurement Conference*, 2013, ACM.

* Work done while a graduate student at UC San Diego.

In this context, our work seeks to better understand the traceability of Bitcoin flows. Importantly, our goal is not to generally de-anonymize all Bitcoin users—as the abstract protocol design itself dictates that this should be impossible—but rather to identify certain *idioms of use* present in concrete Bitcoin network implementations that erode the anonymity of the users who engage in them. We stress that our work was done at a specific point in the evolution of Bitcoin, and that as idioms of use change, the techniques we develop may need to adapt as well.

Our approach is based on the availability of the Bitcoin *block chain*: a replicated graph data structure that encodes all Bitcoin activity, past and present, in terms of the public digital signing keys party to each transaction. However, since each of these keys carries no explicit information about ownership, our analysis depends on imposing additional structure on the transaction graph.

Our methodology has two phases. First, in Section 3, we describe a re-identification attack wherein we open accounts and make purchases from a broad range of known Bitcoin merchants and service providers. Since one endpoint of the transaction is known (i.e., we know which public key we used), we are able to positively label the public key on the other end as belonging to the service; we augment this attack by crawling Bitcoin forums for “self-labeled” public keys (e.g., where an individual or organization explicitly advertizes a key as their own). Next, in Section 4, we build on past efforts^{1, 9, 10, 12} to cluster public keys based on evidence of shared spending authority. This clustering allows us to amplify the results of our re-identification attack: if we labeled one public key as belonging to a particular service, we can now transitively taint the entire cluster containing this public key as belonging to that service as well. The result is a condensed graph, in which nodes represent entire users and services rather than individual public keys.

From this data, we examine the suitability of Bitcoin for hiding large-scale illicit transactions. Using the dissolution of a large Silk Road wallet and notable Bitcoin thefts as case studies, we argue that an agency with subpoena power would be well placed to identify who is paying money to whom. Indeed, we argue that the increasing dominance of a small number of Bitcoin institutions (most notably services that perform currency exchange), coupled with the public nature of transactions and our ability to label monetary flows to major institutions, ultimately makes Bitcoin unattractive for high-volume illicit use such as money laundering.

2. BITCOIN BACKGROUND

The heuristics that we use to cluster pseudonyms depend on the structure of the Bitcoin protocol, so we first describe it here, and briefly mention the anonymity that it is intended to provide. Additionally, much of our analysis discusses the “major players” and different categories of Bitcoin-based services, so we also present a more high-level overview of Bitcoin participation.

2.1. Bitcoin protocol description

Bitcoin is a decentralized electronic currency, introduced by (the pseudonymous) Satoshi Nakamoto in 2008⁷ and

deployed on January 3, 2009. Briefly, a bitcoin can be thought of as a chain of *transactions* from one owner to the next, where owners are identified by a *public key*—from here on out, an *address*—that serves as a pseudonym; that is, users can use any number of addresses and their activity using one set of addresses is not inherently tied to their activity using another set, or to their real-world identity. In each transaction, the previous owner signs—using the secret signing key corresponding to his address—a hash of the transaction in which he received the bitcoins and the address of the next owner. (In fact, transactions can have many input and output addresses, a fact that we exploit in our clustering heuristics in Section 4, but for simplicity we restrict ourselves here to the case of a single input and output.) This signature (i.e., transaction) can then be added to the set of transactions that constitutes the bitcoin; because each of these transactions references the previous transaction (i.e., in sending bitcoins, the current owner must specify where they came from), the transactions form a chain. To verify the validity of a bitcoin, a user can check the validity of each of the signatures in this chain.

To prevent double spending, it is necessary for each user in the system to be aware of all such transactions. Double spending can then be identified when a user attempts to transfer a bitcoin after he has already done so. To determine which transaction came first, transactions are grouped into *blocks*, which serve to timestamp the transactions they contain and vouch for their validity. Blocks are themselves formed into a chain, with each block referencing the previous one (and thus further reinforcing the validity of all previous transactions). This process yields a *block chain*, which is then publicly available to every user within the system.

This process describes how to transfer bitcoins and broadcast transactions to all users of the system. Because Bitcoin is decentralized and there is thus no central authority minting bitcoins, we must also consider how bitcoins are generated in the first place. In fact, this happens in the process of forming a block: each accepted block (i.e., each block incorporated into the block chain) is required to be such that, when all the data inside the block is hashed, the hash begins with a certain number of zeroes. To allow users to find this particular collection of data, blocks contain, in addition to a list of transactions, a *nonce*. (We simplify the description slightly to ease presentation.) Once someone finds a nonce that allows the block to have the correctly formatted hash, the block is then broadcast in the same peer-to-peer manner as transactions. The system is designed to generate only 21 million bitcoins in total. Finding a block currently comes with an attached reward of 25 BTC; this rate was 50 BTC until November 28, 2012 (block height 210,000), and is expected to halve again in 2016, and eventually drop to 0 in 2140.

The dissemination of information within the Bitcoin network is summarized in Figure 1.

2.2. Participants in the Bitcoin network

In practice, the way in which Bitcoin can be used is much simpler than the above description might indicate. First,

generating a block is so computationally difficult that very few individual users attempt it on their own. Instead, users may join a *mining pool*, in which they contribute “shares” to narrow down the search space, and earn a small amount of bitcoins in exchange for each share.

Users may also avoid coin generation entirely, and simply purchase bitcoins through one of the many *exchanges*. They may then keep the bitcoins in a wallet stored on their computer or, to make matters even easier, use a *wallet service* (although many wallet services have suffered thefts and been shut down).

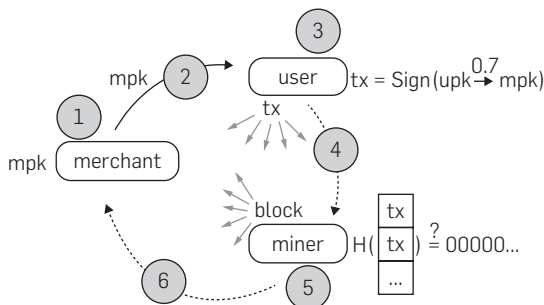
Finally, to actually spend their bitcoins, users could gamble with one of the popular dice games such as Satoshi Dice. They could also buy items from various online vendors. Finally, users wishing to go beyond basic currency speculation can invest their bitcoins with firms such as Bitcoinica (shut down after a series of thefts) or Bitcoin Savings & Trust (later revealed as a major Ponzi scheme).

3. DATA COLLECTION

To identify addresses belonging to the types of services mentioned in Section 2.2, we sought to “tag” as many addresses as possible; that is, label an address as being definitively controlled by some known real-world user. As we will see in Section 4.1, by clustering addresses based on evidence of shared control, we can bootstrap off the minimal ground truth data this provides to tag entire clusters of addresses as also belonging to that user.

Our predominant method for tagging users was simply transacting with them (e.g., depositing into and withdrawing bitcoins from Mt. Gox) and then observing the addresses they used. We additionally collected known (or assumed) addresses that we found in various forums and other Web sites, although we regarded this latter kind of tagging as less reliable than our own observed data.

Figure 1. How a Bitcoin transaction works; in this example, a user wants to send 0.7 bitcoins as payment to a merchant. In (1), the merchant generates or picks an address *mpk*, and in (2) it sends this address to the user. In (3), the user forms the transaction *tx* to transfer the 0.7 BTC from *upk* to *mpk*. In (4), the user broadcasts this transaction to his peers, which (if the transaction is valid) allows it to flood the network. In this way, a miner learns about his transaction. In (5), the miner works to incorporate this and other transactions into a block by checking if their hash is within some target range. In (6), the miner broadcasts this block to her peers, which (if the block is valid) allows it to flood the network. In this way, the merchant learns that the transaction has been accepted into the global block chain, and thus receives the user's payment.



3.1. From our own transactions

We engaged in 344 transactions with a wide variety of services, listed in Table 1, including mining pools, wallet services, bank exchanges, non-bank exchanges, vendors, gambling sites, and miscellaneous services.

Mining pools. We mined bitcoins using an AMD Radeon HD 7970, capable of approximately 530 million SHA-256 computations per second, which allowed us to trigger a payout of at least 0.1 BTC with 11 different pools, anywhere from 1 to 25 times. For each payout transaction, we then labeled the input addresses as belonging to the pool.

Wallets. We kept money with most of the major wallet services (10 in total), and made multiple deposit and withdrawal transactions for each.

Bank exchanges. Most of the real-time trading exchanges (i.e., in which the exchange rate is not fixed) also function as banks. As such, we tagged these services just as we did the wallets: by depositing into and withdrawing from our accounts. We kept accounts with 18 such exchanges in total.

Table 1. The various services we interacted with, grouped by (approximate) type.

Mining		
50 BTC	BTC Guild	Itzod
ABC Pool	Deepbit	Ozcoin
Bitclockers	EclipseMC	Slush
Bitminter	Eligius	
Wallets		
Bitcoin Faucet	Easywallet	Strongcoin
My Wallet	Flexcoin	WalletBit
Coinbase	Instawallet	
Easycoin	Paytunia	
Exchanges		
Bitcoin 24	BTC-e	Aurum Xchange
Bitcoin Central	CampBX	BitInstant
Bitcoin.de	CA VirtEx	Bitcoin Nordic
Bitcurex	ICBit	BTC Quick
Bitfloor	Mercado Bitcoin	FastCash4Bitcoins
Bitmarket	Mt Gox	Lilion Transfer
Bitme	The Rock	Nanaimo Gold
Bitstamp	Vircurex	OKPay
BTC China	Virwox	
Vendors		
ABU Games	BTC Buy	HealthRX
Bitbrew	BTC Gadgets	JJ Games
Bitdomain	Casascius	NZBs R Us
Bitmit	Coinabul	Silk Road
Bitpay	CoinDL	WalletBit
Bit Usenet	Etsy	Yoku
Gambling		
Bit Elfin	BitZino	Gold Game Land
Bitcoin 24/7	BTC Griffin	Satoshi Dice
Bitcoin Darts	BTC Lucky	Seals with Clubs
Bitcoin Kamikaze	BTC on Tilt	
Bitcoin Minefield	Clone Dice	
Miscellaneous		
Bit Visitor	Bitfog	CoinAd
Bitcoin Advertizers	Bitlaundry	Coinapult
Bitcoin Laundry	BitMix	Wikileaks

Non-bank exchanges. In contrast, most of the fixed-rate exchanges did not function as banks, and are instead intended for one-time conversions. We therefore were able to participate in fewer transactions with these exchanges, although we again tried to transact with most of the major ones at least once (eight in total).

Vendors. We purchased goods, both physical and digital, from a wide variety of vendors. Many of the vendors we interacted with did not use an independent method for accepting bitcoins, but relied instead on the BitPay payment gateway (and one used WalletBit as a payment gateway). We also kept a wallet with Silk Road, which allowed us to tag their addresses without making any purchases.

Gambling. We kept accounts with five poker sites, and transacted with eight sites offering mini-games and/or lotteries.

Miscellaneous. Four of the additional services we interacted with were *mix* or *laundry* services: when provided with an output address, they promised to send to that address coins that had no association with the ones sent to them; the more sophisticated ones offered to spread the coins out over various transactions and over time. One of these, BitMix, simply stole our money, while Bitcoin Laundry twice sent us our own coins back, indicating we were possibly their only customer at that time. We also interacted with Bit Visitor, a site that paid users to visit certain sites; Bitcoin Advertisers, which provided online advertizing; CoinAd, which gave out free bitcoins; Coinapult, which forwarded bitcoins to an email address, where they could then be redeemed; and finally, Wikileaks, with whom we donated to both their public donation address and two one-time addresses generated for us via their IRC channel.

3.2. From other sources

In addition to our own transactions, many users publicly claim their own addresses; for example, charities providing donation addresses, or LulzSec claiming their address on Twitter. While we did not attempt to collect all such instances, many of these tags are conveniently collected at blockchain.info/tags, including both addresses provided in users' signatures for Bitcoin forums, as well as self-submitted tags. We collected all of these tags—over 5000 in total—keeping in mind that the ones that were not self-submitted (and even the ones that were) could be regarded as less reliable than the ones we collected ourselves.

Finally, we searched through the Bitcoin forums (in particular, bitcointalk.org) looking for addresses associated with major thefts, or now-defunct services such as Tradehill and GLBSE. Again, these sources are less reliable, so we consequently labeled users only for addresses for which we could gain some confidence through manual due diligence.

4. ADDRESS CLUSTERING

In this section, we present two heuristics for linking addresses controlled by the same user, with the goal of collapsing the many addresses seen in the block chain into larger entities. The first heuristic, in which we treat different addresses

used as inputs to a transaction as being controlled by the same user, has already been used and explored in previous work, and exploits an inherent property of the Bitcoin protocol. The second is new and based on the so-called *change addresses*; in contrast to the first, it exploits a current *idiom of use* in the Bitcoin network rather than an inherent property. As such, it is less robust in the face of changing patterns within the network, but—as we especially see in Section 5—it can provide insight into the Bitcoin network that the first heuristic does not.

4.1. Our heuristics

Heuristic 1. The first heuristic, in which we link together addresses used as input to the same transaction, has already been used many times in previous work.^{1, 9, 10, 12} For completeness, we nevertheless present it here as Heuristic 1: if two (or more) addresses are used as inputs to the same transaction, then they are controlled by the same user.

Using this heuristic, we partitioned the network into 5.5 million clusters of users. By naming these clusters—using the data collection described in Section 3—we observed that some of them corresponded to the same user; for example, there were 20 clusters that we tagged as being controlled by Mt. Gox. (This is not surprising, as many big services appear to spread their funds across a number of distinct addresses to minimize the risk in case anyone gets compromised.) Factoring in “sink” addresses that have to date never sent any bitcoins (and thus did not get clustered using this heuristic) yields at most 6,595,564 distinct users, although we consider this number a quite large upper bound.

Heuristic 2. Although Heuristic 1 already yields a useful clustering of users, restricting ourselves to only this heuristic does not tell the whole story. To further collapse users, our second heuristic focuses on the role of change addresses within the Bitcoin system. A similar heuristic was explored by Androulaki et al.¹ (who called them “shadow” addresses), although there are a number of important differences. In particular, their definition of shadow addresses relied upon assumptions that may have held at the time of their work, but no longer hold at present. For example, they assumed that users rarely issue transactions to two different users, which is a frequent occurrence today (e.g., payouts from mining pools, or bets on gambling sites).

One of the defining features of the Bitcoin protocol is the way that bitcoins must be spent. When the bitcoins redeemed as the output of a transaction are spent, they must be spent all at once: the only way to divide them is through the use of a *change address*, in which the excess from the input address is sent back to the sender. In one idiom of use, the change address is created internally by the Bitcoin client and never re-used; as such, a user is unlikely to give out this change address to other users (e.g., for accepting payments), and in fact might not even know the address unless he inspects the block chain. If we can identify change addresses, we can therefore potentially cluster not only the input addresses for a

transaction (according to Heuristic 1) but also the change address and the input user.

Because our heuristic takes advantage of this idiom of use, rather than an inherent property of the Bitcoin protocol, it does lack robustness in the face of changing (or adversarial) patterns in the network. Furthermore, it has one very negative potential consequence: falsely linking even a small number of change addresses might collapse the entire graph into large “super-clusters” that are not actually controlled by a single user (in fact, we see this exact problem occur in Section 4.2). We therefore focused on designing the safest heuristic possible, even at the expense of losing some utility by having a high false negative rate, and acknowledge that such a heuristic might have to be redesigned or ultimately discarded if habitual uses of the Bitcoin protocol change significantly.

Working off the assumption that a change address has only one input (again, as it is potentially unknown to its owner and is not re-used by the client), we first looked at the outputs of every transaction. If only one of the outputs met this pattern, then we identified that output as the change address. If, however, multiple outputs had only one input and thus the change address was ambiguous, we did not label any change address for that transaction. We also avoided certain transactions; for example, in a coin generation, none of the outputs are change addresses.

In addition, in custom usages of the Bitcoin protocol it is possible to specify the change address for a given transaction. Thus far, one common usage of this setting that we have observed has been to provide a change address that is in fact the same as the input address. (This usage is quite common: 23% of all transactions in the first half of 2013 used self-change addresses.) We thus avoid such “self-change” transactions as well.

To bring all of these behaviors together, we say that an address is a *one-time change address* for a transaction if the following four conditions are met: (1) the address has not appeared in any previous transaction; (2) the transaction is not a coin generation; (3) there is no self-change address; and (4) all the other output addresses in the transaction have appeared in previous transactions. Heuristic 2 then says that the one-time change address—if one exists—is controlled by the same user as the input addresses.

4.2. Refining Heuristic 2

Although effective, Heuristic 2 is more challenging and significantly less safe than Heuristic 1. In our first attempt, when we used it as defined above, we identified over 4 million change addresses. Due to our concern over its safety, we sought to approximate the false positive rate. To do this even in the absence of significant ground truth data, we used the fact that we could observe the behavior of addresses over time: if an address looked like a one-time change address at one point in time (where time was measured by block height), and then at a later time the address was used again, we considered this a false positive. Stepping through time in this manner allowed us to identify 555,348 false positives, or 13% of all labeled change addresses.

We then considered ways of making the heuristic more conservative. First, however, a manual inspection of some of these false positives revealed an interesting pattern: many of them were associated with transactions to and from Satoshi Dice and other dice games. By looking further into the payout structure of these games, it became clear that these were not truly false positives, as when coins are sent to Satoshi Dice, the payout is sent back to the same address. If a user therefore spent the contents of a one-time change address with Satoshi Dice, the address would receive another input back from Satoshi Dice, which would appear to invalidate the “one-timeness” of the address. We therefore chose to ignore this case, believing that addresses that received later inputs solely from Satoshi Dice could still be one-time change addresses. By doing so the false positive rate reduces to only 1%. We next considered waiting to label an address as a change address; that is, waiting to see if it received another input. Waiting a day drove the false positive rate down to 0.28%; waiting a week drove it down to 0.17%, or only 7382 false positives total.

Despite all these precautions, when we clustered users using this modified heuristic, we still ended up with a giant super-cluster containing the addresses of Mt. Gox, Instawallet, BitPay, and Silk Road, among others; in total, this super-cluster contained 1.6 million addresses. After a manual inspection of some of the links that led to this super-cluster, we discovered two problematic patterns. First, especially within a short window of time, the same change address was sometimes used twice. Second, certain addresses were occasionally used as “self-change” addresses, and then later used as separate change addresses. We thus further refined our heuristic by ignoring transactions involved with either of these types of behavior. For transactions in which an output address had already received only one input, or for transactions in which an output address had been previously used in a self-change transaction, we chose to not tag anything as the change address. Doing so, and manually removing a handful of other false positives (with no discernible pattern), we identified 3,540,831 change addresses.

Using this refined Heuristic 2 produces 3,384,179 clusters, which we were able to again collapse slightly (using our tags) to 3,383,904 distinct clusters. Of these clusters, we were able to name 2197 of them (accounting for over 1.8 million addresses). Although this might seem like a small fraction, recall that by participating in 344 transactions we hand-tagged only 1070 addresses, and thus Heuristic 2 allowed us to name 1600 times more addresses than our own manual observation provided. Furthermore, as we will argue in Section 5, the users we were able to name capture an important and active slice of the Bitcoin network.

Having finally convinced ourselves of the safety of Heuristic 2, by refining it substantially, and its effectiveness, we use Heuristic 2 exclusively for the results in the next section.

5. ANALYSIS OF ILLICIT ACTIVITY

Exchanges have essentially become chokepoints in the Bitcoin economy, in the sense that it is unavoidable to buy into or cash out of Bitcoin at scale without using an

exchange. While sites like localbitcoins.com and bitcoinary.com do allow users to avoid exchanges (for the former, by pairing buyers directly with sellers in their geographic area), the current and historical volume on these sites does not seem to be high enough to support cashing out at scale.

In this section, we argue that this centrality presents a unique problem for criminals: if a thief steals thousands of bitcoins, this theft is unavoidably visible within the Bitcoin network, and thus the initial address of the thief is known and (as most exchanges try to maintain some air of reputability) he cannot simply transfer the bitcoins directly from the theft to a known exchange. While he might attempt to use a mix service to hide the source of the money, we again argue that these services do not currently have the volume to launder thousands of bitcoins. As such, we explore in this section various alternative strategies that thieves have developed for hiding the source of stolen bitcoins. In particular, we focus on the effectiveness of Heuristic 2 in de-anonymizing these flows, and thus in tracking illicitly obtained bitcoins to exchanges (and thus, e.g., providing an agency with subpoena power the opportunity to learn whose account was deposited into, and in turn potentially the identity of the thief). For this to work, we do not need to (and cannot) account for each and every stolen bitcoin, but rather need to demonstrate only some flow of bitcoins directly from the theft to an exchange or other known institution.

To demonstrate the effectiveness of Heuristic 2 in this endeavor, we focus on an idiom of use that we call a “peeling chain.” The usage of this pattern extends well beyond criminal activity, and is seen (for example) in the withdrawals for many banks and exchanges, as well as in the payouts for some of the larger mining pools. In a peeling chain, a single address begins with a relatively large amount of bitcoins (e.g., for mining pools it starts with the 25 BTC reward). A smaller amount is then “peeled” off this larger amount, creating a transaction in which a small amount is sent to one address and the remainder is sent to a one-time change address. This process is repeated—potentially for hundreds or thousands of hops—until the larger amount is pared down. By using Heuristic 2, we are able to track flows of money by following these change links systematically: at each hop, we look at the two output addresses in the transaction. If one of these output addresses is a change address, we can follow the chain to the next hop by following the change address (i.e., the next hop is the transaction in which this change address spends its bitcoins), and can identify the meaningful recipient in the transaction as the other output address (the “peel”).

Silk road and Bitcoin savings & trust. One of the most well-known and heavily scrutinized addresses in Bitcoin’s history is 1DkyBEkt—full address: 1Dky-BEkt5S2Gdt-v7aQw6rQepAvnsRyHoYM—which is believed to be associated with Silk Road and was active between January and September 2012. Starting in January, the address began to receive large aggregate sums of bitcoins; in the first of these, the funds of 128 addresses were combined to deposit 10,000 BTC into the 1DkyBEkt address, and many trans-

actions of this type followed. All together, the address received 613,326 BTC in a period of eight months, receiving its last aggregate deposit on August 16, 2012.

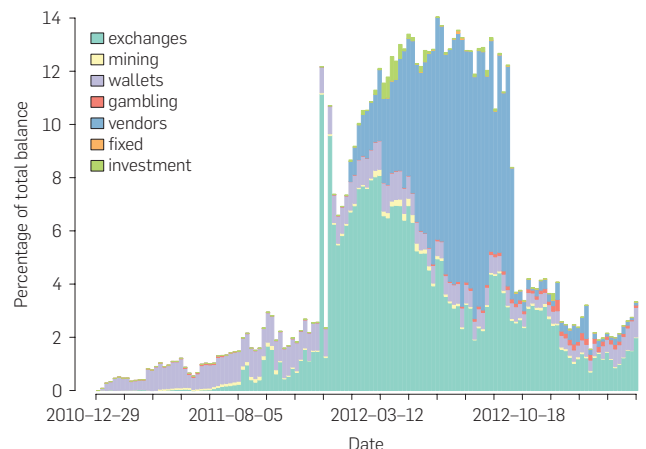
Then, starting in August 2012, bitcoins were aggregated and withdrawn from 1DkyBEkt: first, amounts of 20,000, 19,000, and 60,000 BTC were sent to separate addresses; later, 100,000 BTC each was sent to two distinct addresses, 150,000 BTC to a third, and 158,336 BTC to a fourth, effectively emptying the 1DkyBEkt address of all of its funds.

Due to its large balance (at its height, it contained 5% of all generated bitcoins), as well as the curious nature of its rapidly accumulated wealth and later dissolution, this address has naturally been the subject of heavy scrutiny by the Bitcoin community. While it is largely agreed that the address is associated with Silk Road (and indeed our clustering heuristic did tag this address as being controlled by Silk Road), some have theorized that it was the “hot” (i.e., active) wallet for Silk Road, and that its dissipation represents a changing storage structure for the service. Others, meanwhile, have argued that it was the address belonging to the user pirate@40, who was responsible for carrying out the largest Ponzi scheme in Bitcoin history (the investment scheme Bitcoin Savings & Trust, which is now the subject of a lawsuit brought by the SEC¹¹).

To see where the funds from this address went, and if they ended up with any known services, we first plotted the balance of each of the major categories of services, as seen in Figure 2. Looking at this figure, it is clear that when the address was dissipated, the resulting funds were not sent en masse to any major services, as the balances of the other categories do not change significantly. To nevertheless attempt to find out where the funds did go, we turn to the traffic analysis described above.

In particular, we focus on the last activity of the 1DkyBEkt address, when it deposited 158,336 BTC into a single address. This address then peeled off 50,000 BTC each to two separate addresses, leaving 58,336 BTC for a third address; each of these addresses then began a peeling chain, which we

Figure 2. The balance of each major category, represented as a percentage of total active bitcoins; that is, the bitcoins that are not held in sink addresses.



followed using the methodology described above (i.e., at each hop we continued along the chain by following the change address, and considered the other output address to be a meaningful recipient of the money). After following 100 hops along each chain, we observed peels to the services listed in Table 2.

In this table, we see that, although a longitudinal look at the balances of major services did not reveal where the money went, following these chains revealed that bitcoins were in fact sent to a variety of services. The overall balance was not highly affected, however, as the amounts sent were relatively small and spread out over a handful of transactions. Furthermore, while our analysis does not itself reveal the owner of 1DkyBEKt, the flow of bitcoins from this address to known services demonstrates the prevalence of these services (54 out of 300 peels went to exchanges alone) and provides the potential for further de-anonymization: the evidence that the deposited bitcoins were the direct result of either a Ponzi scheme or the sale of drugs might motivate Mt. Gox or any exchange (e.g., in response to a subpoena) to reveal the account owner corresponding to the deposit address in the peel, and thus provide information to link the address to a real-world user.

Tracking thefts. To ensure that our analysis could be applied more generally, we turned finally to a broader class of criminal activity in the Bitcoin network: thefts. Thefts are in fact quite common within Bitcoin: almost every major service has been hacked and had bitcoins (or, in the case of exchanges, other currencies) stolen, and some have shut down as a result.

To begin, we used a list of major Bitcoin thefts found at <https://bitcointalk.org/index.php?topic=83794>. Some of the thefts did not have public transactions (i.e., ones we

could identify and study in the block chain), so we limited our attention to the ones that did. For each theft, we first found the specific set of transactions that represented the theft; that is, the set of transactions in which the sender was the service and the recipient was the thief. Starting with these transactions, we did a preliminary manual inspection of the transactions that followed to determine their approximate type: we considered aggregations, in which bitcoins were moved from several addresses into a single one; folding, in which some of the aggregated addresses were not clearly associated with the theft; splits, in which a large amount of bitcoins was split among two or more addresses; and finally peeling chains, in which smaller amounts were peeled off from a succession of one-time change addresses. Our results are summarized in Table 3.

Briefly, the movement of the stolen money ranged from quite sophisticated layering and mixing to simple and easy to follow. Examining thefts therefore provides another demonstration of the potential for anonymity provided by Bitcoin, and the ways in which current usage falls short of this potential. For the thieves who used the more complex strategies, we saw little opportunity to track the flow of bitcoins (or at least do so with any confidence that ownership was staying the same), but for the thieves that did not there seemed to be ample opportunity to track the stolen money directly to an exchange.

One of the easiest thefts to track was from Betcoin, an early gambling site that was shut down after its server was hacked on April 11, 2012 and 3171 BTC were stolen. The stolen bitcoins then sat in the thief's address until March 15, 2013 (when the bitcoin exchange rate began soaring), when they were aggregated with other small addresses into one large address that then began a peeling chain. After 10 hops, we saw a peel go to Bitcoin-24, and in another 10 hops we saw a peel go to Mt. Gox; in total, we saw 374.49 BTC go to known exchanges, all directly off the main peeling chain, which originated directly from the addresses known to belong to the thief.

In contrast, some of the other thieves used more sophisticated strategies to attempt to hide the flow of money; for example, for the Bitfloor theft, we observed that large peels off several initial peeling chains were then aggregated,

Table 2. Tracking bitcoins from 1DkyBEKt.

Service	First		Second		Third	
	Peels	BTC	Peels	BTC	Peels	BTC
Bitcoin-24			1	2	3	124
Bitcoin Central					2	2
Bitcoin.de					1	4
Bitmarket					1	1
Bitstamp			5	97	1	1
BTC-e					1	250
CA VirtEx	1	3	1	10	3	22
Mercado Bitcoin					1	9
Mt. Gox	11	492	14	70	5	35
OKPay	2	151			1	125
Instawallet	7	39	5	135	2	43
WalletBit	1	1				
Bitzino					2	1
Seals with Clubs	1	8				
Coinabul			1	29		
Medsforbitcoin	3	10				
Silk Road	4	28			5	102

Along the first 100 hops of the first, second, and third peeling chains resulting from the withdrawal of 158,336 BTC, we consider the number of peels seen to each service, as well as the total number of bitcoins (rounded to the nearest integer value) sent in these peels. The services are separated into the categories of exchanges, wallets, gambling, and vendors.

Table 3. Tracking thefts.

Theft	BTC	Date	Movement	Exchanges?
MyBitcoin	4019	Jun 2011	A/P/S	Yes
Linode	46,648	Mar 2012	A/P/F	Yes
Betcoin	3171	Mar 2012	F/A/P	Yes
Bitcoinica	18,547	May 2012	P/A	Yes
Bitcoinica	40,000	Jul 2012	P/A/S	Yes
Bitfloor	24,078	Sep 2012	P/A/P	Yes
Trojan	3257	Oct 2012	F/A	No

For each theft, we list (approximately) how many bitcoins were stolen, when the theft occurred, how the money moved after it was stolen, and whether we saw any bitcoins sent to known exchanges. For the movement, we use A to mean aggregation, P to mean a peeling chain, S to mean a split, and F to mean folding, and list the various movements in the order they occurred.

and the peeling process was repeated. Nevertheless, by manually following this peel-and-aggregate process to the point that the later peeling chains began, we systematically followed these later chains and again observed peels to multiple known exchanges: the third peel off one such chain was 191.09 BTC to Mt. Gox, and in total we saw 661.12 BTC sent to three popular exchanges (Mt. Gox, BTC-e, and Bitstamp).

Even the thief we had the most difficulty tracking, who stole bitcoins by installing a trojan on the computers of individual users, seemed to realize the difficulty of cashing out at scale. Although we were unable to confidently track the flow of the stolen money that moved, most of the stolen money did not in fact move at all: of the 3257 BTC stolen to date, 2857 BTC was still sitting in the thief's address, and has been since November 2012.

With these thefts, our ability to track the stolen money provides evidence that even the most motivated Bitcoin users (i.e., criminals) are engaging in idioms of use that allow us to erode their anonymity. While one might argue that thieves could easily thwart our analysis, our observation is that—at least at the time we performed our analysis—none of the criminals we studied seem to have taken such precautions. We further argue that the fairly direct flow of bitcoins from the point of theft to the deposit with an exchange provides some evidence that using exchanges to cash out at scale is inevitable, as otherwise thieves presumably would have avoided this less anonymous method of cashing out. Thus, Bitcoin does not—again, at the time we performed our analysis—seem to provide a particularly easy or effective way to transact large volumes of illicitly obtained money.

6. CONCLUSION

In this study, we presented a longitudinal characterization of the Bitcoin network, focusing on the growing gap—due to certain idioms of use—between the potential anonymity available in the Bitcoin protocol design and the actual anonymity that is currently achieved by users. To accomplish this task, we developed a new clustering heuristic based on change addresses, allowing us to cluster addresses belonging to the same user. Then, using a small number of transactions labeled through our own empirical interactions with various services, we identify major institutions. Even our relatively small experiment demonstrates that this approach can shed considerable light on the structure of the Bitcoin economy, how it is used, and those organizations who are party to it.

Although our work examines the current gap between actual and potential anonymity, one might naturally wonder—given that our new clustering heuristic is not fully robust in the face of changing behavior—how this gap will evolve over time, and what users can do to achieve stronger anonymity guarantees. We posit that to completely thwart our heuristics would require a significant effort on the part of the user, and that this loss of usability is unlikely to appeal to all but the most motivated users (such as criminals). Nevertheless, we leave a quantitative analysis of this hypothesis as an interesting open problem. 

References

1. Androulaki, E., Karame, G., Roeschlin, M., Scherer, T., Capkun, S. Evaluating user privacy in Bitcoin. In *Proceedings of Financial Cryptography 2013* (2013).
2. CBC News. Revenue Canada says BitCoins aren't tax exempt, Apr. 2013. www.cbc.ca/news/canada/story/2013/04/26/business-bitcoin-tax.html.
3. Eha, B.P. Get ready for a Bitcoin debit card. CNNMoney, Apr. 2012. money.cnn.com/2012/08/22/technology/startups/bitcoin-debit-card/index.html.
4. European Central Bank. Virtual Currency Schemes. ECB Report, Oct. 2012. www.ecb.europa.eu/pub/pdf/other/virtualcurrencyschemes201210en.pdf.
5. Federal Bureau of Investigation. (U) Bitcoin virtual currency unique features present distinct challenges for deterring illicit activity. Intelligence assessment, cyber intelligence and criminal intelligence section, Apr. 2012. cryptome.org/2012/05/fbi-bitcoin.pdf.
6. FinCEN. Application of FinCEN's regulations to persons administering, exchanging, or using virtual currencies, Mar. 2013. www.fincen.gov/statutes_regs/guidance/pdf/FIN-2013-G001.pdf.
7. Nakamoto, S. Bitcoin: A peer-to-peer electronic cash system, 2008. bitcoin.org/bitcoin.pdf.
8. Peck, M. Bitcoin-central is now the World's First Bitcoin Bank...kind of. IEEE Spectrum: Tech talk, Dec. 2012. spectrum.ieee.org/tech-talk/telecom/internet/bitcoincentral-is-now-the-worlds-first-bitcoin-bankkind.
9. Reid, F., Harrigan, M. An analysis of anonymity in the Bitcoin system. In *Security and Privacy in Social Networks*. Y. Altshuler, Y. Elovici, A.B. Cremers, N. Aharony, and A. Pentland, eds. Springer, New York, 2013, 197–223.
10. Ron, D., Shamir, A. Quantitative analysis of the full Bitcoin transaction graph. In *Proceedings of Financial Cryptography 2013* (2013).
11. Securities and Exchange Commission. SEC charges Texas man With running Bitcoin-denominated Ponzi scheme, July 2013. www.sec.gov/News/PressRelease/Detail/PressRelease/1370539730583.
12. znort987. [blockparser](https://github.com/znort987/blockparser). github.com/znort987/blockparser.

Sarah Meiklejohn (s.meiklejohn@ucl.ac.uk), University College London, London, U.K.

Marjori Pomarole (marjoripomarole@gmail.com), Facebook, London, U.K.

Grant Jordan, Kirill Levchenko, Geoffrey M. Voelker, and Stefan Savage ({ljordan, klevchen, voelker, savage}@cs.ucsd.edu), UC San Diego, La Jolla, CA.

Damon McCoy (damon.mccoy@gmail.com), ICSI, Berkeley, CA.

Copyright held by authors. Publication rights licensed to ACM. \$15.00.



Watch the author discuss his work in this exclusive *Communications* video. <http://cacm.acm.org/videos/a-fistful-of-bitcoins>

CAREERS

The University of California, Berkeley Full Professorship

THE UNIVERSITY OF CALIFORNIA, BERKELEY invites applications for an approved tenured **FULL PROFESSORSHIP** commencing with a five-year 100% appointment as **DIRECTOR** of the **Simons Institute for the Theory of Computing** with an expected start date of July 1, 2017. For more information about the position, including required qualifications and application materials, go to: <http://www.eecs.berkeley.edu/Faculty-Jobs/>. The deadline to apply is April 15, 2016. For questions please contact the Search Committee Chair at eeecs-faculty-recruiting@eecs.berkeley.edu. UC Berkeley is an AA/EEO employer.



Have a question
about advertising
opportunities?

CONTACT US
212 • 626 • 0686
acmm mediasales@acm.org



Association for
Computing Machinery

ACM LEARNING CENTER

RESOURCES
FOR LIFELONG LEARNING

learning.acm.org



Online Courses from Skillsoft

Online Books from Safari, Books24x7,
Morgan Kaufmann and Syngress

Webinars on today's hottest topics
in computing



Association for
Computing Machinery



Faculty positions in Electrical and Computer Engineering in Africa

The College of Engineering at Carnegie Mellon University, a world leader in information and communication technology, has extended its global reach into Africa. In 2012 we became the first U.S.-based research university offering on-site master's degrees in Africa at our base in Kigali, Rwanda.

Carnegie Mellon University in Rwanda is educating future leaders who will use their hands-on, experiential learning to advance technology innovation and grow the businesses that will transform Africa.

We are seeking highly qualified candidates to join our world-class faculty, who share in our vision of developing creative and technically strong engineers that will impact society. Faculty members are expected to collaborate with industry and deliver innovative, interdisciplinary graduate teaching and research programs.

Carnegie Mellon is seeking exceptional candidates who can deliver innovative, interdisciplinary graduate programs in these areas:

- Software engineering
- Mobile and cloud computing
- Communications and wireless networking
- Cybersecurity and privacy
- Embedded systems
- Energy systems
- Image and signal processing
- Data analytics
- Applications in healthcare, agriculture, finance and infrastructure
- Innovation and technology management

Candidates should possess a Ph.D. in a related discipline and an outstanding record in research, teaching and leadership.

-
- Please contact us at info@rwanda.cmu.edu for full application requirements.
 - Further information about CMU in Rwanda can be found at www.cmu.edu/rwanda.
 - Applications should be submitted by email to director@rwanda.cmu.edu.

**Carnegie Mellon University
Rwanda**



DOI:10.1145/2892635

Dennis Shasha

Upstart Puzzles

Sleep No More

WE BEGIN SIMPLY, with a 60-minute clock that counts only minutes, from 0 to 59. The alarm can also be set from 0 to 59 and will go off when the clock reaches the same value. Say you want the alarm to go off in m ($m < 60$) minutes, the time value now is x , and the alarm value is y . You want to move the time value or alarm value forward as little as possible so the alarm goes off m minutes from now.

Warm-up. The time is at 20 minutes, and the alarm is at 5 minutes. You want the alarm to go off in 35 minutes. One option is to move the alarm forward (the only allowable direction) to 55. Another is to move the time value ahead to 30. The second is less expensive, requiring only 10 pushes, so you prefer that.

In general, for this 60-minute clock, if T is the time value and A is the alarm value and you want to wake up in m (< 60) minutes, which value do you move and at what cost in terms of number of minutes ahead you must push that value?

Solution. Recall $(y-x) \bmod 60 = y - x$ if $x < y$ or $(y+60) - x$, otherwise; for example, $(14-4) \bmod 60 = 10$, but $(4-14) \bmod 60 = 50$; $(y-x) \bmod 60$ is thus the number of minutes on the 60-minute clock value y is ahead of x .

Let $L2$ be the minimum non-negative value having the property $m = (A - (T+L2)) \bmod 60$. $L2$ is the number of minutes we would have to advance the time to achieve our goal of waking up in m minutes. We call it `timeadvance` and solve for it as follows:

$$\text{timeadvance}(A, T, m) = (A - (T+m)) \bmod 60.$$

If $\text{timeadvance}(A, T, m) \leq 30$, then advance the time by that



As with every alarm clock, this one can hardly wait to ring, and you must figure out how to set it to wake you when your nap is over, making as few button pushes as possible.

amount, else advance the alarm by $60 - \text{timeadvance}(A, T, m)$. End of solution.

Now imagine you have a 24-hour clock for both alarm and time. You are faced with the same problem. You want the alarm to go off in m minutes, where m can now be any number up to $(24 \times 60) - 1$ minutes.

You can move the hour value (between 0 and 23) for both time and alarm separately from the minute value (between 0 and 59). Each move forward by one of any hour or minute counter costs one unit of effort. (Moving the minute value past 59 to 0 does not affect the hour value.)

Is it ever an advantage to move both a time value and an alarm value, as opposed to just one?

Solution. Yes. Suppose the time is set at 15:18 and the alarm is 14:50.

You want the alarm to go off in 30 minutes. The best thing is to move the alarm forward by one hour and the time to 15:20, costing a total of three units of effort.

Can you now find an elegant, cost-minimizing algorithm for the problem? The alarm clock will still have the pleasure of waking you up, but you will have the satisfaction of knowing the clock will never know what time it really is. 

All are invited to submit their solutions to upstartpuzzles@cacm.acm.org; solutions to upstarts and discussion will be posted at <http://cs.nyu.edu/cs/faculty/shasha/papers/cacmpuzzles.html>

Dennis Shasha (dennisshasha@yahoo.com) is a professor of computer science in the Computer Science Department of the Courant Institute at New York University, New York, as well as the chronicler of his good friend the omniheurist Dr. Ecco.

Copyright held by author.



fuse **DIS 2016**

Designing Interactive Systems

June 4 – 8

Brisbane Australia

 dis2016.org

 bit.ly/dis16

 [@DIS2016](https://twitter.com/DIS2016)



Association for
Computing Machinery



Queensland University of Technology
Brisbane Australia



Computing Reviews

20th ANNUAL BEST OF COMPUTING

Best Reviews Notable Books & Articles

The 20th Annual Best of Computing presents Computing Reviews' best reviews and the most interesting books and articles of the year.

The Best Reviews highlight the excellent contributions of the reviewers. Based on the recommendations from our reviewers, category editors, editors in chief of journals, and others in the computing community, the Notable Books & Articles list brings together influential items published in computing.

Available Online & in Print



Association for
Computing Machinery

ThinkLoud

www.computingreviews.com/best20