

Securing Digital Finances ■ Integrating Business Operations with Cyberdefense ■ Supporting Non-Tech-Savvy Users

IEEE

SECURITY & PRIVACY

BUILDING DEPENDABILITY, RELIABILITY, AND TRUST

GENOME Privacy and Security

September/October 2017
Vol. 15, No. 5

 IEEE

 IEEE
computer
society

 IEEE
Reliability Society

Call for Papers

Special Issue on Security and Privacy Research in Brazil

for *IEEE Security & Privacy* magazine's November/December 2018 issue

Articles due to ScholarOne: 1 March 2018

Author guidelines: www.computer.org/web/peer-review/magazines

The goal of this special issue is to showcase cutting-edge security and privacy research being conducted by the Brazilian community, with topics unique to Brazil. Brazil's cybersecurity capabilities are growing, and a recent joint program between the US's National Science Foundation (NSF) and Brazil's Ministry of Science, Technology, Innovation and Communication (MCTI) program has increased focus on opportunities for international collaboration (www.usbrazilsec.org).

Examples of issues that are unique to Brazil include: relationships of privacy laws, perceptions, and preferences in Brazil versus other countries; case studies of cybersecurity in Brazilian networks and critical infrastructure; legal aspects of cybersecurity in Brazil; similarities and differences in cybersecurity innovation characteristics in Brazil versus other countries; cyberattacks specific to Brazil (for instance, Boletos); and security and privacy problems that are of particular national importance or unique expertise that is specific to Brazil's culture, education system, location, history, and so on.

Considering this focus on unique Brazilian cybersecurity issues, topics for the special issue may include, but are not limited to:

- Malware analysis and detection
- Network security

- Hardware security
- Internet of Things
- Privacy and perceptions
- Cryptography
- Usable security and human factors in cybersecurity
- Interdisciplinary security
- Web security

Submission Guidelines

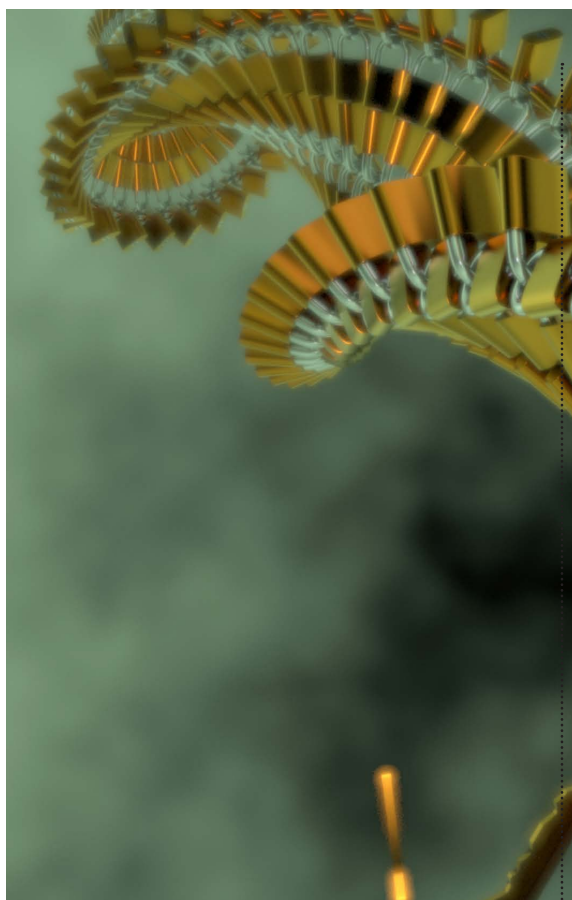
Submissions will be subject to the IEEE Computer Society's peer-review process, and if accepted, to the Computer Society editing process. Articles should be at most 6,000 words, with a maximum of 15 references, and should be understandable to a broad audience of people interested in security, privacy and dependability. The writing style should be down-to-earth, practical, and original. Authors should not assume that the audience will have specialized experience in a particular subfield. All accepted articles will be edited by a staff editor according to the IEEE Computer Society style guide. Submit your papers to ScholarOne at <https://mc.manuscriptcentral.com/cs-ieee>.

Questions?

Contact the guest editors:

- Daniela Oliveira, University of Florida, daniela@ece.ufl.edu
- Jeremy Epstein, National Science Foundation, jepstein@nsf.gov
- Anderson Rocha, University of Campinas, anderson.rocha@ic.unicamp.br

www.computer.org/security/cfp



Cover art by Peter Bollinger, www.shannonassociates.com

Genome Privacy and Security

Genomic data has the potential to be highly sensitive. Thus, privacy and security are considered paramount when collecting or sharing such data. This special issue explores the risks of genomic data storage. Furthermore, it provides an overview of solutions ensuring the integrity and privacy of genomic computation.

10 Guest Editors' Introduction: Genomic Data Privacy and Security: Where We Stand and Where We Are Heading

Jean-Pierre Hubaux, Stefan Katzenbeisser, and Bradley Malin

14 Characterizing the Risks and Harms of Linking Genomic Information to Individuals

Sara Renee Savage

20 Improving the Security and Efficiency of Private Genomic Computation Using Server Aid

Marina Blanton and Fattaneh Bayatbabolghani

29 Inference Attacks against Kin Genomic Privacy

Erman Ayday and Mathias Humbert

38 Genomic Security (Lest We Forget)

Tatiana Bradley, Xuhua Ding, and Gene Tsudik

Also in This Issue

47 Cybersecurity Framework Adoption: Using Capability Levels for Implementation Tiers and Profiles

Adenekan Dedek

55 152 Simple Steps to Stay Safe Online: Security Advice for Non-Tech-Savvy Users

Robert W. Reeder, Iulia Ion, and Sunny Consolvo

65 Assessing Mission Impact of Cyberattacks: Toward a Model-Driven Paradigm

Alexander Kott, Jackson Ludwig, and Mona Lange



76



85

Columns

3 From the Editors

AI Industrial Complex: The Challenge of AI Ethics
Ahmad-Reza Sadeghi

96 Last Word

IoT Security: What's Plan B?
Bruce Schneier

Also in This Issue

28 | IEEE Computer Society
Information

37 | IEEE Reliability Society
Information

Departments

7 Interview

Silver Bullet Talks with Ksenia Dmitrieva-Peguero
Gary McGraw

76 Sociotechnical Security and Privacy

Security and Privacy Experiences and Practices of Survivors of Intimate Partner Abuse
Tara Matthews, Kathleen O'Leary, Anna Turner, Manya Sleeper, Jill Palzkill Woelfer, Martin Shelton, Cori Manthorne, Elizabeth F. Churchill, and Sunny Consolvo

82 Real-World Crypto

Disillusioning Alice and Bob
Rolf Oppliger

85 Systems Attacks and Defenses

FinTechSec: Addressing the Security Challenges of Digital Financial Services
Patrick Traynor, Kevin Butler, Jasmine Bowers, and Bradley Reaves

90 Cybercrime and Forensics

Availability of Required Data to Support Criminal Investigations Involving Large-Scale IP Address-Sharing Technologies
David O'Reilly

94 In Focus

The Open Science Cyber Risk Profile: The Rosetta Stone for Open Science and Cybersecurity
Sean Peisert and Von Welch



Postmaster: Send undelivered copies and address changes to *IEEE Security & Privacy*, Membership Processing Dept., IEEE Service Center, 445 Hoes Lane, Piscataway, NJ 08854-4141. Periodicals postage rate paid at New York, NY, and at additional mailing offices. Canadian GST #125634188. Canada Post Publications Mail Agreement Number 40013885. Return undeliverable Canadian addresses to PO Box 122, Niagara Falls, ON L2E 6S8. Printed in the USA. **Circulation:** *IEEE Security & Privacy* (ISSN 1540-7993) is published bimonthly by the IEEE Computer Society. IEEE Headquarters, Three Park Ave., 17th Floor, New York, NY 10016-5997; IEEE Computer Society Publications Office, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720, phone +1 714 821 8380; IEEE Computer Society Headquarters, 2001 L St., Ste. 700, Washington, D.C. 20036. Subscribe to *IEEE Security & Privacy* by visiting www.computer.org/security. *IEEE Security & Privacy* is copublished by the IEEE Computer and Reliability Societies. For more information on computing topics, visit the IEEE Computer Society Digital Library at www.computer.org/csdl.

AI Industrial Complex: The Challenge of AI Ethics

Modern societies are increasingly dependent on evolving intelligent IT systems. The physical and virtual worlds are melding into a cyberworld that is highly dynamic and intelligent (that is, context aware, self-organized, and adaptive), unbounded in scale (that is, vastly growing number of devices, users, services, and businesses), unstructured and decentralized, and without fixed perimeters (with fluid borders between the inside and outside of systems).

AI plays an important role in this development and is currently undergoing a major revival. It's being gradually integrated into numerous application domains, such as social networks, the financial sector, autonomous systems, and data analytics. AI is of interest not only to industry but also to nation-states.

Much has been said and written in the recent past about AI's benefits and hazards. While it can certainly benefit our lives in many ways, AI also presents a variety of security, privacy, and safety challenges.

The Concerned

In 2015, prominent (AI and other) researchers signed and published an open letter against autonomous weapons.¹ The letter advocates a ban on weapons that “select and engage targets without human intervention.” It's argued that, although such weapons help reduce our own casualties, they also lower the threshold of going into battle. The signatories fear that development of such weapons would lead to an AI weapons arms race and ultimately result in such weapons leaking into the black market and, subsequently, into the hands of nefarious actors, such as terrorists or dictators. Such actors could utilize AI weapons in “assassinations, destabilizing nations, subduing populations and selectively killing a particular ethnic group.” Development of such weapons could also harm the reputation of other AI technologies that are beneficial to humanity.

A group of concerned AI scientists and engineers including captains of industry, such as Elon Musk, CEO of Tesla and SpaceX, has asked the United Nations (UN) to regulate the use of autonomous weapon systems.² They argue that civilians need to be protected from the misuse of AI-driven weapons, as they could be easily abused by despots or tyrants and would represent a dangerous development in weapons technology. More recently, Musk has spoken out about the dangers of AI, as an AI system designed by OpenAI won all of its one-on-one games against human teams (which included many of the world's best players) in the multiplayer online battle arena game DotA 2. Musk stated that emergence of AI poses “vastly more risk” than North Korea's nuclear capabilities.³

The most dystopian vision is offered by the famed theoretical physicist Stephen Hawking, who warns that development of AI could represent a threat to the very existence of the human race.⁴ He argues that if machines acquire the capability to engineer themselves to be far more intelligent, it might lead to an “intelligence explosion,” which in turn could cause situations where machines would eradicate humans if their goals were not aligned with those of humans. He therefore advocates an approach wherein we “shift the goal of AI from creating pure undirected AI to creating beneficial intelligence.”⁴

These concerns might become reality in the future; however, “beneficial intelligence” seems like a theoretical wish, difficult to materialize in the real world. Most of us benefit from this technology in one way or another. For instance, people use transport systems and vehicles every day. But any intelligent vehicle controlled by malicious (hacked) AI can become a weapon. Today, we're (sadly) becoming accustomed to disgruntled and misguided individuals who terrorize innocent people by driving vehicles into crowds. In the not-too-distant future, these people



Ahmad-Reza Sadeghi
Editor in Chief

might subvert a fleet of autonomous vehicles to achieve a much greater (and much grimmer) impact without their own physical presence at the scene of the crime. Does this mean that the automotive industry, among others Tesla, isn't going to produce autonomous cars?

The Advocates

Some experts are still skeptical that the envisioned dangers of AI would materialize in the foreseeable future. They argue that we're still far from machines being able to achieve the required computational power and develop the necessary algorithms needed to attain intelligence that matches human cognitive capabilities.⁵ Others experts, such as robotics researcher Rodney Brooks, the founding director of MIT's Computer Science and Artificial Intelligence Lab, think that critics (such as Musk) lack insights into, and understanding of, AI.⁶

It seems obvious that service providers (for instance, Facebook, Google, Apple, and Amazon and their business units and start-up acquisitions) will continue to invest in AI. Some industry stalwarts, such as Mark Zuckerberg, find the fear of AI to be exaggerated and believe that limiting or slowing down the use of AI is unjustified due to AI's benefits for humanity.⁵ (This is of course along the business model of many service providers wherein users are more or less the product because they give away their data.)

AI technologies that diagnose diseases and that are used in autonomous vehicle navigation are example areas in which AI is not simply beneficial; it can literally save countless human lives. To this end, some industrial initiatives (for instance, by the alliance of Facebook, Google, and Microsoft) aim to alleviate concerns and fears about AI.

Famed futurist Ray Kurzweil claims that AI can improve humans' positive cognitive capabilities.

He thinks that the so-called singularity—that is, the moment when AI exceeds the intellectual capacity of humans—is merely a decade away. He argues that, by 2029, computers will reach human-level intelligence and people will start using computers to enhance their own cognitive capabilities by physically connecting parts of their brain to this computer support.⁷

However, the question remains: Will our lives, in the near future, be ruled and controlled by a friendly club of creatures such as Alexa, Siri, Cortana, and “Ok Google,” and is this the future we want?

AI Ethics

We can observe a trend toward people trusting AI to do “magic” unattainable by humans. However, there are many open questions to be answered and challenges to be tackled technologically, socially, and judicially, such as:

- Who will own the copyrights for AI algorithms and the data they process—Google, Facebook, pharmaceutical industries, and so on?
- Will AI be used for more intelligent ways to discriminate; surveil and violate privacy; manipulate democratic procedures; carry out war crimes through malfunctioning autonomous robots; generate or launder money for those who used to have accounts in tax havens; or enhance the obfuscation of deals in the financial market and create even more revenue for Wall Street, the City of London, and the like?
- Will AI be deployed to determine a person's potential for conducting criminal or terroristic acts?
- To what extent can AI be compromised or fooled by skilled adversaries?
- What are the design rules for AI when privacy regulations such

as the European General Data Protection Regulation are on the verge of being implemented?

- Will we need new laws and regulations for AI as we have for privacy in some regions of the world?
- How will elected officials and judicial authorities, who aren't technical experts, effectively draft and interpret laws that control such technologies?

In my view, another concern is AI's opaqueness and its invisible impact on our society. When George Orwell wrote his fascinating novel *1984*, he had no idea what social networks and online service providers could do with echo chambers and data analytics to make our private lives completely transparent. Indeed, since the beginning of the last election campaigns in the US, UK, and many other parts of Europe, experts have been debating the possible manipulation of election outcomes through deployment of sophisticated classification and data analytics algorithms. Whether companies such as Cambridge Analytica used their machinery to infer sensitive personal attributes of people or predict their behavior to eventually manipulate them during the election is still not proven. However, the power of intelligent algorithms and AI seems undeniable even by most skeptics.

Recently, I was invited to take part in a panel of “Concerned Scientists” (www.ucsusa.org/nuclear-weapons/international-symposium-science-and-world-affairs). It was an amazing, eye-opening event attended by many accomplished scientists and activists from various disciplines including physics, politics, philosophy, and sociology—some of whom have been fighting for decades to establish regulations and control mechanisms for nuclear disarmament, and to build technologies and procedures to estimate nation-states'

nuclear power potential. We were philosophizing about how to deal with highly intelligent and powerful AI in the cyberworld, a kind of Skynet (from the Hollywood movie *The Terminator*). The essence of the discussion was that the antinuclear and peace movement already has tremendous experience and a legacy that can be partially applied to the cyberworld, particularly to AI and its deployment.

Concerns about AI misuse have led to many initiatives on AI ethics. Both the NSF and the EU are funding research in this area, IEEE has a Global Initiative for Ethical Considerations in AI and Autonomous Systems (standards.ieee.org/develop/indconn/ec/autonomous_systems.html), and the ACM is hosting a panel on Algorithmic Transparency and Accountability (www.acm.org/media-center/2017/august/usacm-ata-panel-media-advisory). Other examples are the ethics of computer science research workshop at Princeton (citp.princeton.edu/event/ethics-conf) and the German Ethics committee's report on automated and connected driving.⁸

Researchers have already been looking into issues such as algorithmic accountability and verifiability to prevent intelligent decision algorithms from discrimination, such as misusing personal data; building privacy-preserving learning algorithms; and mitigating adversarial machine learning. These are actually simpler examples (although still sufficiently complex) compared to what future AI can potentially achieve 100 years from now.

I believe that scientists, practitioners, and decision makers in industry and governments should learn from past public debates and outcomes on ethics in nuclear research, biology, medicine, and the environment. AI will surely impact our lives greatly. However, it also has a high potential for misuse with

very grave consequences for the world as a whole.

IEEE Security & Privacy magazine is continually looking for interesting, exciting, and challenging technology and research topics. The general subject of AI ethics is of great interest to industry, government, and academia, for both advocates and opponents. A particularly vital aspect in this context is that officials, judges, and probably regulators—who are almost certainly not technology experts—might not be able to construct useful and effective legal frameworks.

Our magazine will publish a special issue on AI Ethics: The Privacy Challenge with many interesting interdisciplinary articles by experts. The special issue is also connected to a workshop on the same topic in collaboration with the Future of Privacy Forum (fpf.org/2017/05/09/ai-ethics-privacy-challenge).

I hope our magazine can provide our readers more insight into this fascinating topic. ■

References

1. "Autonomous Weapons: An Open Letter from AI & Robotics Researchers," Future of Life Institute, 28 July 2015; futureoflife.org/open-letter-autonomous-weapons.
2. D. Cooper, "Elon Musk Urges the UN to Limit AI Weapons," Engadget, 21 Aug. 2017; www.engadget.com/2017/08/21/elon-musk-ai-un-letter.
3. S. Gibbs, "Elon Musk: AI 'Vastly More Risky than North Korea,'" *The Guardian*, 14 Aug. 2017; www.theguardian.com/technology/2017/aug/14/elon-musk-ai-vastly-more-risky-north-korea.
4. A. Griffin, "Stephen Hawking: Artificial Intelligence Could Wipe Out Humanity When It Gets Too Clever as Humans Will Be Like Ants," *The Independent*,

- 8 Aug. 2015; www.independent.co.uk/life-style/gadgets-and-tech/news/stephen-hawking-artificial-intelligence-could-wipe-out-humanity-when-it-gets-too-clever-as-humans-a6686496.html.
5. O. Solon, "Killer Robots? Musk and Zuckerberg Escalate Row over Dangers of AI," *The Guardian*, 25 July 2017; www.theguardian.com/technology/2017/jul/25/elon-musk-mark-zuckerberg-artificial-intelligence-facebook-tesla.
6. C. Loizos, "This Famous Robotist Doesn't Think Elon Musk Understands AI," TechCrunch, 19 July 2017; techcrunch.com/2017/07/19/this-famous-robotist-doesnt-think-elon-musk-understands-ai.
7. A. Sulleyman, "The Singularity: AI Will Make Humans Sexier and Funnier, Says Google Expert," *The Independent*, 16 Mar. 2017; www.independent.co.uk/life-style/gadgets-and-tech/news/singularity-artificial-intelligence-humans-sexy-funny-ai-music-art-google-futurist-engineering-ray-a7633481.html.
8. *Ethics Commission: Automated and Connected Driving*, report, Federal Minister of Transport and Digital Infrastructure, June 2017; www.bmvi.de/SharedDocs/EN/Documents/G/ethic-commission-report.pdf?__blob=publicationFile.



EDITOR IN CHIEF

Ahmad-Reza Sadeghi | Technische Universität Darmstadt

ASSOCIATE EDITORS IN CHIEF

N. Asokan | Aalto University
Robin Bloomfield | City University London
Jeremy Epstein | National Science Foundation
Gene Tsudik | University of California, Irvine

EDITORIAL BOARD

George Cybenko* | Dartmouth College
Robert Deng | Singapore Management University
Carrie Gates
Dieter Gollmann | Technical University Hamburg-Harburg
Feng Hao | Newcastle University
Carl E. Landwehr* | George Washington University
Roy Maxion | Carnegie Mellon University
Nasir Memon | Polytechnic University
Rolf Oppliger | eSECURITY Technologies
Sean Peisert | Lawrence Berkeley National Laboratory and University of California, Davis
Anderson Rocha | University of Campinas
Matthew Smith | University of Bonn
Paul Van Oorschot | Carleton University
*EIC Emeritus

DEPARTMENT EDITORS

Building Security In | Jonathan Margulies, Qmulos
Cybercrime and Forensics | Pavel Gladyshev, University College Dublin
Education | Melissa Dark, Purdue University; Jelena Mirkovic, University of Southern California Information Sciences Institute; and Bill Newhouse, NIST
Interview/Silver Bullet | Gary McGraw, Synopsis
Privacy Interests | Katrine Evans, Hayman Lawyers
Real-World Crypto | Peter Gutmann, University of Auckland; David Naccache, École Normale Supérieure; and Charles C. Palmer, IBM
Resilient Security | Mohamed Kaàniche, French National Center for Scientific Research; and Richard Kuhn, NIST
Sociotechnical Security and Privacy | Heather Richter Lipford, University of North Carolina at Charlotte; and Jessica Staddon, Google
Systems Attacks and Defenses | Davide Balzarotti, EURECOM; William Enck, North Carolina State University; Thorsten Holz, Ruhr-University Bochum; and Angelos Stavrou, George Mason University

COLUMNISTS

Last Word | Bruce Schneier, Harvard University; Steven M. Bellovin, Columbia University; and Daniel E. Geer Jr., In-Q-Tel

STAFF

Lead and Content Editor | Christine Anthony
Staff Editor | Rebecca Torres
Contributing Editor | Meghan O'Dell
Publications Coordinator | security@computer.org
Production Editor/Webmaster | Mark J. Bartosik
Production Staff | Erica Hardison, Monette Velasco, and Jennie Zhu-Mai
Graphic Design | Graphic World
Original Illustrations | Robert Stack
Director, Products & Services | Evan Butterfield
Sr. Manager, Editorial Services | Robin Baldwin
Manager, Editorial Content | Carrie Clark
Sr. Advertising Coordinator | Debbie Sims, dsims@computer.org

CS MAGAZINE OPERATIONS COMMITTEE

George K. Thiruvathukal (chair), Gul Agha, Brian Blake, Jim X. Chen, Maria Ebling, Lieven Eeckhout, Miguel Encarnação, Nathan Ensmenger, Sumi Helal, San Murugesan, Yong Rui, Ahmad-Reza Sadeghi, Diomidis Spinellis, VS Subrahmanian, Mazin Yousif

CS PUBLICATIONS BOARD

Greg Byrd (VP for Publications), Alfredo Benso, Irena Bojanova, Robert Dupuis, David S. Ebert, Davide Falessi, Vladimir Getov, José Martínez, Forrest Shull, George K. Thiruvathukal

EDITORIAL OFFICE

IEEE Security & Privacy
c/o IEEE Computer Society Publications Office
10662 Los Vaqueros Circle, Los Alamitos, CA 90720 USA
Phone | +1 714 821-8380; Fax | +1 714 821-4010

PUBLISHING COSPONSORS



TECHNICAL COSPONSORS



Editorial | Unless otherwise stated, bylined articles, as well as product and service descriptions, reflect the author's or firm's opinion. Inclusion in *IEEE Security & Privacy* does not necessarily constitute endorsement by the IEEE or the IEEE Computer Society. All submissions are subject to editing for style, clarity, and length.
Submissions | We welcome submissions about security and privacy topics. For detailed instructions, see the author guidelines (www.computer.org/security/author.htm) or log onto IEEE Security & Privacy's author center at ScholarOne (<https://mc.manuscriptcentral.com/cs-ieee>).
Reuse Rights and Reprint Permissions | Educational or personal use of this material is permitted without fee, provided such use: 1) is not made for profit; 2) includes this notice and a full citation to the original work on the first page of the copy; and 3) does not imply IEEE endorsement of any third-party products or services. Authors and their companies are permitted to post the accepted version of IEEE-copyrighted material on their own Web servers without permission, provided that the IEEE copyright notice and a full citation to the original work appear on the first screen of the posted copy. An accepted manuscript is a version which has been revised by the author to incorporate review suggestions, but not the published version with copyediting, proofreading, and formatting added by IEEE. For more information, please go to www.ieee.org/publications_standards/publications/rights/paperversionpolicy.html. Permission to reprint/republish this material for commercial, advertising, or promotional purposes or for creating new collective works for resale or redistribution must be obtained from IEEE by writing to the IEEE Intellectual Property Rights Office, 445 Hoes Lane, Piscataway, NJ, USA 08854-4141 or pubs-permissions@ieee.org. Copyright © 2017 IEEE. All rights reserved.
Abstracting and Library Use | Abstracting is permitted with credit to the source. Libraries are permitted to photocopy for private use of patrons, provided the per-copy fee indicated in the code at the bottom of the first page is paid through the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923.
IEEE prohibits discrimination, harassment, and bullying: For more information, visit www.ieee.org/web/aboutus/whatis/policies/p9-26.html.

Silver Bullet Talks with Ksenia Dmitrieva-Peguero

Gary McGraw | Synopsis

Hear the full podcast at www.computer.org/silverbullet. Show links, notes, and an online discussion can be found at www.cigital.com/silverbullet.



Ksenia Dmitrieva-Peguero is a principal consultant in Synopsys' Software Integrity Group. She has many years of hands-on experience in software and systems security and is an expert in many practices including penetration testing, static analysis tool design and execution, customization and deployment, and threat modeling. Throughout her career as a consultant, Dmitrieva-Peguero has established and evolved secure coding guidance and best practices for many different firms and has delivered numerous software security

training sessions. She's passionate about cutting-edge web technologies and probing systems security, and she speaks regularly around the world on topics such as HTML5, CSP, and JavaScript.

You've been doing hands-on software security in the real world for many years. How has the field evolved since you started doing consulting seven years ago?

Seven years ago, a lot of people didn't know what we were talking about when we talked about security. We would look for people to hire with programming experience, and we didn't care about security experience. You would learn everything on the job. These days, there are degrees in security, and we expect candidates to know security basics and have some experience. When you talk to clients, they know what software security is. They've definitely heard of penetration testing and static analysis. Today, the awareness is much higher.

Do you think we've made progress as a field in those seven years, beyond raising awareness?

There are still problems, but I think we're making progress. The field has definitely grown. There's more demand and more understanding of the kinds of problems we're trying to solve, of the difference between software security and all the other things that have to do with IT and security in a company. In terms of quality, I'm not sure.

Maybe the quality remained constant?

There's more software and therefore more bugs, and there are more things to be attacked. Now we have software in mobile devices and cars and everywhere else. So, it might look like there are more security issues, but it's probably also society being more aware of the issues. It's hard to say if we've made any qualitative progress.

If you could fix any practice area in software security today, which one would it be?

Probably threat modeling. Architecture review follows because that's the more complicated one, and people have very different descriptions and understanding of what it is. Everybody has their own version of threat modeling and risk analysis.

It might be because we can build a program to go through our code and look for bugs, but we can't build a similar system to go through a design and look for flaws.

Absolutely. There's less tooling because the process is much more involved. Maybe with the

development of artificial intelligence we'll be better able to develop some sort of automation for understanding design problems.

Frameworks like Angular, Node, and Express have been popping up all over the place. In your view, do all frameworks have the same security posture?

Definitely not. All frameworks have the problem of building security in and how much each framework developer or maintainer wants to make security a part of it. Unfortunately, these days that's not the highest priority for the framework creators. Very few even bother with it. Then developers are left to solve all the security problems themselves while they're using the framework. Instead of having the features built in, they have to reinvent the wheel every single time.

On one hand, security should be the responsibility of the framework creators and maintainers. On the other hand, the developers using the framework should demand it. When they choose a framework, they should ask about its security posture. I don't see any questions on forums like Stack Overflow saying, "Hey, which framework is more secure?" Usually you see a question like, "Hey, which framework has more visual components, features, or is faster?"

So, the developers building the frameworks don't really deal with security, and the developers using the frameworks can screw everything up from a security perspective. Right. For validation, developers might write their own routine to validate email addresses, and that will most likely be code copied from Stack Overflow or another GitHub repository where the quality isn't guaranteed. Depending on the developer, the framework, and how well the library is integrated with the framework, they might be using

a third-party library, which is often a better solution—a little more validated, updated, and secure. But in both cases, they have opportunities to screw things up because even if you use a third-party library or a plug-in, it usually requires some configuration. We've seen these examples in Angular where a plug-in comes with a pretty good default security setting, but it doesn't satisfy some features that developers need. So developers turn off security settings; they change them and then the plug-in doesn't do the job it's supposed to do.

You've spent time digging into Angular and thinking about how to automate aspects of security analysis. What have you learned about Angular, and what have you done to get that into automation?

Angular is interesting. On one hand, it's client-side code. We don't trust anything that runs on the client. A lot of things can be bypassed. But on the other hand, the way the templating is done in Angular provides good protection from cross-site scripting even if malicious data is passed through the server-side code. In terms of automatically finding security problems, that's still a big question. There isn't a good tool for JavaScript today. It's possible to find some dataflow and cross-site scripting issues, but the issues that have to do with the configuration of the framework or plug-ins are harder because they are updated so often. We're always chasing the updates that happen every couple of months. When a new version of Angular comes out, not everything changes, but there might be enough changes that our automated tools don't find things anymore.

Have you talked to the developers building Angular? Are they aware of what you're doing?

We haven't communicated directly with them. But from the Angular

developer standpoint, when they introduce new features, they often say, "Hey, this is a breaking build, and we don't really care about the older versions. We don't have the requirements or the desire to support whatever was built before. So, we're just going to go ahead and start a new version."

You have to figure out what they did, and then figure out how to adjust whatever you built to work again every time?

If a developer started using Angular 1.6 before Angular 2 came out, sometimes they'll just stick with the old version because they'd have to rewrite their whole application due to the breaking changes in the framework. Not every company will do that. If developers are still using the older 1.6 version, the tools that were built for 1.6 will still work. But they won't work for a new application.

Do you think version controlling and keeping everything somewhat similar in terms of its security posture are the biggest open problems?

I think one of the problems is keeping up with all the versions and upgrading to the new features and plug-ins. The second problem is keeping up with the variety of frameworks. Last year, Angular was number one, but I think it's fading out and React is stepping in. Now we'll need to build automation tools for React. Six months from now, it'll be something else.

Here's a trick question. What's more important: code review or architecture risk analysis [ARA]?

It depends on the application. If it's a standard web app that has a database, a back end, and a front end, you might not gain as much doing ARA. You could start with a code review and get more bang for your buck that way. But if it's software in a car that isn't standard or

another very complicated application then, yes, we should definitely start with ARA.

How do you think code review and ARA are related?

If you're reviewing a complex system, you'll start with an ARA, which will help you identify potential issues. To find out if these issues actually exist in the system, you would do a code review. You would look at something and say, "Hey, this system is talking to this third-party back end that has its own protocol. How about we review the code of this third-party back end and this protocol and how the communication happens?" In doing code review of this scope, we might find some issues.

You've experienced and lived in different cultures all over the world. Do diverse cultures approach computer security differently, or is it the same?

I don't think there's much difference in the countries I've lived and worked in. I'm from Russia and have worked in Europe and the US. I didn't see any differences in how people understand or relate to computer security. Maybe in other cultures, it's different. I think if you look at other areas like education or psychology or maybe medicine, there might be cultural differences, but if you look at technical stuff—math and computers—I think it's pretty straightforward in any culture.

You give a lot of talks all over the place. What's your favorite conference to attend or speak at?

One of my favorites so far has been AppSec in Europe. There's a great concentration of web technology experts, which is my playground. I really enjoy communicating and interacting with and listening to all these amazingly smart people. In Europe, I think people are very relaxed and friendly and less commercialized than in the



About Ksenia Dmitrieva-Peguero

Ksenia Dmitrieva-Peguero is a principal consultant at Synopsys, where she leads a JavaScript research group that concentrates on common web application vulnerabilities and best practices. Her key areas of expertise include Web 2.0, JavaScript, HTML5, and Content Security Policy. Dmitrieva-Peguero loves studying new technologies, finding their vulnerabilities, and discovering ways to protect them. She presents at conferences frequently, including AppSec Europe, BSides Security London, and RSA Asia Pacific and Japan. Dmitrieva-Peguero received an MS in computer science from George Washington University. Outside of the office, she is a competitive ballroom dancer. She lives in Virginia with her husband and their brand-new baby girl.

US. The conference is more like the exchange of ideas and not the exchange of business cards and promotional materials.

As a hardcore technologist and a woman, what's your view of sexism in the field? Do you think it's harder to gain respect as a technologist if you're a woman?

Yes, unfortunately, it's there. It's harder to gain respect for sure, and I think it's especially hard to gain respect at the middle level. When you're at a higher level—for example, when speaking at a big conference—there are requirements to have men and women represented equally. And a conference might actually tend to select talks from women more often than from men. At the middle level, working with clients and establishing your position and trust as a woman can be very challenging.

You have to sort of prove yourself a little bit more?

Yes. I've been in situations where I was working with a male colleague and the client would interact with him but not with me.

Even though you probably knew more than the other guy?

Sometimes, yes.

Oh, come on, always. One last question. One of the coolest things you do is competitive ballroom dancing. When did you start dancing, and did you ever think about going pro?

It's been about 10 years. I've never thought about going pro because I started dancing in Russia. Most people in Russia start ballroom dancing when they are six or seven years old. If you're starting after high school, there's no way you can become a professional—it's treated only as a hobby. In the US, you can become a pro, but you have to make your life out of it. For me, it's a hobby that I get a lot of enjoyment from.

The Silver Bullet Podcast with Gary McGraw is cosponsored by Cigital (part of Synopsys) and this magazine and is syndicated by SearchSecurity. ■

Gary McGraw is vice president of security technology at Synopsys. He's the author of *Software Security: Building Security In* (Addison-Wesley 2006) and eight other books. McGraw received a BA in philosophy from the University of Virginia and a dual PhD in computer science and cognitive science from Indiana University. Contact him via garymcgraw.com.



Genomic Data Privacy and Security: Where We Stand and Where We Are Heading

Jean-Pierre Hubaux | Ecole Polytechnique Fédérale de Lausanne
Stefan Katzenbeisser | Technische Universität Darmstadt
Bradley Malin | Vanderbilt University

Since the seminal discoveries of James Watson, Francis Crick, Maurice Wilkins, and Rosalind Franklin, it's taken humankind more than 40 years to develop technology that can generate a complete human genome sequence. Over the past 20 years, the price of a whole genome sequencing has plummeted from the tens of millions to a few thousand dollars, and will likely soon reach the hundreds.¹ This ability to read the DNA of each of us has fueled immense hopes in healthcare and, specifically, in personalized medicine initiatives² that range from targeted cancer therapeutics³ to precision drug dosing.⁴ Beyond healthcare, DNA sequencing data is now routinely applied in forensics,⁵ law enforcement activities,⁶ parental testing,⁷ and ancestry mapping.⁸

Genomic data has the potential to be highly sensitive, such that privacy and security are often considered paramount when collecting, applying, or sharing such records. Yet, while privacy and security solutions have been developed for various types of data, genomic sequences have proven to be particularly challenging to protect for a number of reasons that include but aren't limited to the following:

- There are still vast swaths within an individual's DNA sequence that we don't have a concrete understanding about.⁹
- Genomics is only the first instance of the “-omics” revolution (other instances include proteomics, transcriptomics, and microbiomics).
- The information security field is evolving at a fast pace and, sooner or later, will have to address challenges such as quantum computing.
- Several potential correlation attacks within genomic data and with other data (phenotypic, or even that gathered from online social networks) have already

been demonstrated and many more can be anticipated.

- DNA leaks information beyond the person from whom it was collected, as it also involves kinship.
- The emergence of the *quantified self*, with its own rich set of body and metabolic measurements, further complicates the situation.

An additional complexity stems from the number and diversity of the involved stakeholders: healthcare organizations, sequencing facilities, insurance companies, regulators, direct-to-consumer genomic companies, ancestry-related online social networks, patients, and patients' relatives. Addressing the issue of data protection in this field requires a multidisciplinary collaboration of geneticists, bioinformaticians, information security specialists, jurists, ethicists, and patient organizations. From a computing perspective, critical questions that need to be addressed include the following:

- Where is the data stored?
- How is it protected?
- How are access rights managed and by whom?
- Which computations are permitted?
- Under what conditions can data be shared across institutions?

Over the past two decades, researchers have worked on the risks induced by this data's existence and availability as well as on techniques to mitigate these risks. This has led to two Dagstuhl seminars,^{10,11} a community website (genomeprivacy.org), and two workshops—the iDash Privacy and Security Workshop (www.humangenomeprivacy.org) and the International Workshop on Genome Privacy and Security (www.humangenomeprivacy.org). In parallel, the Global Alliance for Genomics and Health has set up a working group on security (genomicsandhealth.org/working-groups/security-working-group).

The articles in this special issue address several key aspects of the privacy and security of genomic data.

The first article, "Characterizing the Risks and Harms of Linking Genomic Information to Individuals," by Sara Renee Savage, provides an overview of the risks people face when they contribute their own DNA to genomic databases. The article further characterizes potentially harmful consequences that must be accounted for when developing services based on genetic data.

The second article, "Improving the Security and Efficiency of Private Genomic Computation Using Server Aid," by Marina Blanton and Fattaneh Bayatbabolghani, proposes a cryptographic solution to support paternity tests and genomic compatibility tests with the help of

a server. The article emphasizes the resilience against malicious behavior, thus preventing an attacker from learning unauthorized information.

The third article, "Inference Attacks against Kin Genomic Privacy," by Erman Ayday and Mathias Humbert, reviews the various approaches for learning genomic information about individuals based on what their relatives disclose. The authors trace the evolution of these methods from basic forensic science to modern computational methods. Beyond inference attacks, this article introduces several data protection methods that could help people mitigate risk when disclosing genomic data.

The final article, "Genomic Security (Lest We Forget)" by Tatiana Bradley, Xuhua Ding, and Gene Tsudik explores an issue that has received little attention in the scientific community. Although numerous authors have considered the privacy aspects of genomic data, solutions that provide security against malicious data modifications are rare. The article surveys the challenges involved when securing large and high-dimensional data such as genomes.

These articles provide crucial insights, mostly from a computing perspective. Yet, much more research is needed, and considering the magnitude and the strategic nature of the challenge, we encourage security and privacy researchers to participate in this effort. The landscape is likely to continue its rapid evolution, fueled, in particular, by sequencing technology itself. It could well be that, next year, you will sequence yourself with your smartphone, as the technology seems to be almost mature.¹² How will you protect this data? ■

References

1. "The Cost of Sequencing a Human Genome," Nat'l Human Genome Research Inst., Nat'l Inst. of Health; www.genome.gov/sequencingcosts.
2. R. Mirnezami, J. Nicholson, and A. Darzi, "Preparing for Precision Medicine," *New England J. Medicine*, vol. 366, no. 6, 2012, pp. 489–491.
3. L.A. Garraway, J. Verweij, and K.V. Ballman, "Precision Oncology: An Overview," *J. Clinical Oncology*, vol. 31, no. 15, 2013, pp. 1803–1805.
4. G.P. Hess et al., "Pharmacogenomic and Pharmacogenetic-Guided Therapy as a Tool in Precision Medicine: Current State and Factors Impacting Acceptance by Stakeholders," *Genetics Research Cambridge*, vol. 97, no. 13, 2015; doi:10.1017/S0016672315000099.
5. M.A. Jobling and P. Gill, "Encoded Evidence: DNA in Forensic Analysis," *Nature Reviews Genetics*, vol. 5, Oct. 2004, pp. 739–751.

6. M. Rothstein and M.K. Talbott, "The Expanding Use of DNA in Law Enforcement: What Role for Privacy?," *J. Law, Medicine, and Ethics*, vol. 34, no. 2, 2006, pp. 153–164.
7. J.M. Butler, "Genetics and Genomics of Core Short Tandem Repeat Loci Used in Human Identity Testing," *J. Forensic Sciences*, vol. 51, no. 2, 2006, pp. 253–265.
8. C.D. Royal et al., "Inferring Genetic Ancestry: Opportunities, Challenges, and Implications," *Am. J. Human Genetics*, vol. 86, no. 5, 2010, pp. 661–673.
9. W.F. Doolittle, "Is Junk DNA Bunk? A Critique of ENCODE," *Proc. Nat'l Academy of Sciences USA*, vol. 110, no. 14, 2013, pp. 5294–5300.
10. K. Hamacher, J.P. Hubaux, and G. Tsudik, "Genomic Privacy (Dagstuhl Seminar 13412)," *Dagstuhl Reports*, vol. 3, no. 10, 2014, pp. 25–35; www.dagstuhl.de/en/program/calendar/semhp/?semnr=13412.
11. J.P. Hubaux et al., "Genomic Privacy (Dagstuhl Seminar 15431)," *Dagstuhl Reports*, vol. 5, no. 10, 2016, pp. 50–65; www.dagstuhl.de/en/program/calendar/semhp/?semnr=15431.
12. L. Harley, "Just a SmidgION: Oxford Nanopore Announces iPhone-Powered Sequencing," *Front Line Genomics*, 27 May 2017; www.frontlinegenomics.com/news/5452/just-a-smidgion-oxford-nanopore-announce-iphone-powered-sequencing.

Jean-Pierre Hubaux is a professor of information security and privacy in the School of Computing and Communication Sciences at the Ecole Polytechnique Fédérale de Lausanne (EPFL). Contact him at jean-pierre.hubaux@epfl.ch.

Stefan Katzenbeisser is a professor of security engineering in the Computer Science Department at Technische Universität Darmstadt. Contact him at katzenbeisser@seceng.informatik.tu-darmstadt.de.

Bradley Malin is a professor of biomedical informatics, biostatistics, and computer science and vice chair in the Department of Biomedical Informatics at Vanderbilt University. Contact him at b.malin@vanderbilt.edu.



IEEE Annals

of the History of Computing

From the analytical engine to the supercomputer, from Pascal to von Neumann, from punched cards to CD-ROMs—*IEEE Annals of the History of Computing* covers the breadth of computer history. The quarterly publication is an active center for the collection and dissemination of information on historical projects and organizations, oral history activities, and international conferences.

www.computer.org/annals



CALL FOR NOMINEES

Education Awards Nominations



Taylor L. Booth Education Award

A bronze medal and US\$5,000 honorarium are awarded for an outstanding record in computer science and engineering education. The individual must meet two or more of the following criteria in the computer science and engineering field:

- Achieving recognition as a teacher of renown.
- Writing an influential text.
- Leading, inspiring or providing significant education content during the creation of a curriculum in the field.
- Inspiring others to a career in computer science and engineering education.

Two endorsements are required for an award nomination.

See the award information at:

www.computer.org/web/awards/booth

Computer Science and Engineering Undergraduate Teaching Award

A plaque, certificate and a stipend of US\$2,000 is awarded to recognize outstanding contributions to undergraduate education through both teaching and service and for helping to maintain interest, increase the visibility of the society, and making a statement about the importance with which we view undergraduate education.

The award nomination requires a **minimum of three endorsements**.

See the award details at:

www.computer.org/web/awards/cse-undergrad-teaching



Deadline: 1 October 2017
Nomination Site: awards.computer.org





Characterizing the Risks and Harms of Linking Genomic Information to Individuals

Sara Renee Savage | Vanderbilt University

Genetic data is a valuable resource for learning about disease risks and progression. However, individuals, their relatives, and institutions storing genetic data are at risk for harm if DNA reidentification occurs. Four classes of harms, along with their risks and outcomes, are discussed.

The Human Genome Project's successful completion in the early 2000s stimulated the expansion of genomics research. Technological advances have now made it possible to sequence a human genome in 24 hours for just over \$1,000. This, in turn, has made it feasible to accumulate and store the genetic information of many individuals in large repositories.

Large genomic repositories are extremely valuable to researchers. This data, combined with limited demographics, can be mined for human migration patterns and information on genetic variation. With the addition of simple medical information, researchers can discover genes' contribution to phenotypes, determine how an individual might respond to a particular drug, and understand how DNA mutations affect protein function. Public or easily accessible databases of genetic information reduce barriers to discovery by decreasing sample collection's cost and time.

The downside to these public databases is that they risk violating individuals' genomic privacy. DNA is unique, barring circumstances such as identical twins or tissue transplants. Therefore, special consideration must be taken to protect contributors' privacy, because linking genetic information to an individual could have negative consequences. With interest increasing in

projects such as the Precision Medicine Initiative (ghr.nlm.nih.gov/primer/precisionmedicine/initiative), it's essential to evaluate the risks involved with contributing genetic information.

Individuals might experience two types of risk when contributing their DNA to a public database: privacy breaches, and the consequences of these breaches. A privacy breach can include individuals' reidentification from their genetic data or associated metadata as well as the use of identified genetic data to learn additional information about individuals.¹ The latter type of attack includes *attribute disclosure attacks*, which use individuals' genetic data to learn new sensitive attributes. It also encompasses *completion attacks*, in which genetic data is used to infer information about a biological relative, or partially identified data is used to impute unknown genetic data.¹ In the following sections, I focus on reidentification, which can lead to both attribute disclosure and completion attacks.

Genomic Identification and Metadata

Individuals submit their DNA to public research repositories with the expectation that their genetic information will remain anonymous. However, they could possibly be reidentified through the actions of data

intruders. Intruders might seek to reidentify genomic information in an effort to amass a portfolio of information about an individual or to experience monetary gain for linking individuals to their DNA. The success of a data intruder's attack depends on the data available.

Multiple databases containing genetic information exist, but they differ in the type of information they store, their ease of access, and the number of individuals they contain. For example, Harvard's Personal Genome Project (www.personalgenomes.org) contains approximately 5,000 participants but only around 700 whole genome sequences. The data is publically available, and many of the samples are linked to personal and health information. Meanwhile, the National Human Genome Research Institute's The Cancer Genome Atlas (TCGA; cancergenome.nih.gov) contains more than 1,000 breast cancer samples with exome data, in addition to samples of numerous other cancer types. Access to this data requires a data use agreement and a principal investigator account in the electronic Research Administration (eRA) Commons.

Differing database characteristics lead to different attack methods. Data intruders can reidentify individuals using the metadata (such as ZIP code, gender, and age) that accompanies genomic information. If that data is unavailable, intruders might try to determine a name from the DNA itself. For example, Melissa Gymrek and her colleagues showed that genomes can be linked to surnames via short tandem repeats on the Y chromosome.²

Reidentification entails three steps:

- accessing the data,
- determining identifying information, and
- linking the data with known identifiers.

Intruders will need both time to perform these steps and money to access the identifiable linking records. Intruders might also face barriers to accessing data in the form of data use agreements. The total cost for these steps would be spread over the number of individuals an intruder identifies.

Case Studies

I present here two examples of the steps intruders would have to take to reidentify individuals in two different databases. I chose these cases based on predicted ease of reidentification, with one being easy and the other difficult. This will provide an idea of the challenges data intruders face in linking data.

Personal Genome Project. The first example is the Personal Genome Project, a public resource for sharing genomic and health information that doesn't guarantee

privacy. Some participants provide structured, identifying information such as full name and birth date. Identifying information can be extracted from other profiles whose filenames include full names. Many of the remaining profiles contain quasi-identifying information such as birth date, ZIP code, and gender.

Latanya Sweeney and her colleagues showed that 42 percent of participants can be reidentified—with 97 percent accuracy—by using a voter registration database and a public records database to link quasi-identifying information to unique full names.³ The cost of accessing the data is minimal, because a hacker can write a web scraper that collects the mostly structured data. Three months of access to the voter registration database Voter Mapping (www.votermapping.com) costs \$1,500. So, assuming the intruder can tolerate the 3 percent error, if 42 percent of the 728 individuals with whole genome data can be reidentified, then the cost per record is just over \$5. With the data in the Personal Genome Project profile, a large amount of information can be learned about an individual for a small cost, which could be reduced if the database contained more individuals.

I selected a random individual from the Personal Genome Project and performed a five-minute Google search to determine the amount and type of information an intruder could collect on an individual. My name and ZIP code search revealed one individual in the appropriate age range. Links to the individual's social media accounts and job website revealed a vast array of information, including marital status, number of children, likely political affiliation, high school, preferred charities, and current photographs. The person's full genome was also available.

Sequence Read Archive. On the other end of the spectrum is the Sequence Read Archive hosted by the National Center for Biotechnology Information (NCBI; www.ncbi.nlm.nih.gov/sra). This database contains at least 100,000 whole human genome samples, and more than 70,000 RNA-sequencing samples. The number of unique individuals these numbers correspond to is unknown. Because this data is publically available, I asked whether a data intruder could reidentify individuals using genomic information when there was no other identifying information. Although some researchers have shown that DNA itself can be identifiable, such reidentification attacks' feasibility hasn't been well studied. In Gymrek and her colleagues' work, reidentification required three to seven hours per individual, with only 12 percent of individuals being reidentified.²

Processing the raw data in the Sequence Read Archive requires significant time and knowledge of bioinformatics tools. Analyzing one sample using a

common tool such as TopHat requires approximately 4.5 hours for downloading the data, aligning to a reference genome, processing, and variant calling and filtering. The data produced from these steps is simply single nucleotide variations from a reference genome. Extracting identifiable phenotypes from these genotypes currently requires commercial software. One website, Promethease (www.promethease.com), will provide this information for \$10 per sample. The resulting phenotypes include information such as probable eye color, hair color, and disease risks. Although these characteristics might help identify an individual if other data is available, they can't be used alone for reidentification.¹

Preventing Privacy Loss

Extensive work has explored ways to mitigate privacy breaches, including methods to provide legal protections, add noise to the data, or share summary statistics instead of full genetic data.⁴ The legal protection of data use agreements and significant punishment for misusing data could deter data intruders, and suppressing data or adding noise could reduce the risk of reidentifying an individual.¹ In addition, a recent article described a game theory approach to determining the best data-sharing policy for preventing attacks.⁴

These strategies reduce the risk of reidentification of genetic data, but the possibility of a privacy breach is never completely abolished. The remaining sections describe a taxonomy of harms that certain groups might experience if reidentification of their genomic data occurred. There is also the risk of some of these harms when individuals willingly provide their full identifying information with their DNA.

Taxonomy of Harms

To generate a set of harms, I performed a preliminary literature review. Between 1 January 2015 and 20 April 2016, PubMed contained 111 articles with the phrase “genomic privacy.” I manually reviewed all accessible articles for mentions of harms. I then separated the identified harms into four groups:

- *harms to individuals*—those that target specific people,
- *harms to relatives*—those experienced by biological relatives of the reidentified individual, and
- *harms to populations*—those that are more general and affect a large number of individuals,
- *harms to institutions*—those experienced by entities that didn't have their own genetic privacy breached.

Harms have both a risk and an outcome. The risk is a combination of the harm's magnitude and probability. As a starting point, I set as low risk those harms that aren't currently feasible or are only experienced by a few

individuals. A moderate-risk harm has temporary negative effects and a medium probability of occurrence. A high-risk harm affects more than 50 percent of individuals who are reidentified or has long-lasting effects. My work extends that of Catherine Heeney and her colleagues, who noted that the amount and type of data available, along with the context of the data, have a large effect on risk.⁵ Table 1 outlines the types of harms and their risks and outcomes.

Harms to Individuals

With reidentification or voluntary disclosure of identifying data, individuals have a high risk of learning their disease risks because of the large number of known genotype–phenotype interactions. This might cause anxiety or other psychological outcomes if the individuals have a high risk for a severe disease, such as breast cancer.⁶ Individuals might also learn family secrets, such as misattributed parentage, which happens anywhere between 1 and 20 percent of the time.⁷ Furthermore, individuals could risk being linked to a crime that could send them to jail or require a fee payment. Law enforcement has asked 23andMe for information on individuals four different times.⁸ In addition, DNA could be recreated and planted at crime scenes or used for cloning.⁹ However, this last harm is unlikely, with no known occurrences.

Discrimination is a commonly cited risk in genomic privacy research. The Genetic Information Nondiscrimination Act of 2008 (GINA) prohibits genetic discrimination for jobs and health insurance. However, it doesn't extend to the US military. It also excludes other types of insurance, such as life, disability, and long-term care. As an anecdote, FastCompany reported one healthy woman was denied life insurance because her genomic data revealed mutation of a breast cancer gene. This case is concerning because the woman might have been seriously injured or died by a completely different cause, in which case life insurance would have benefited her family. In addition, having this mutation doesn't guarantee the development of breast cancer.

Discrimination anecdotes besides denial of insurance coverage have also been reported. In one case, *Chadam v. Palo Alto Unified School District*, a student was prevented from attending a school based on his DNA, which indicated that he had genetic markers associated with cystic fibrosis.¹⁰ When the family of a student with cystic fibrosis attending the same school discovered this, they requested that the child be transferred. Cystic fibrosis is rare, so by extension, multiple children with cystic fibrosis attending the same school is also rare. When such situations do occur, the Cystic Fibrosis Foundation recommends placing the children in different classrooms, or keeping them at least two

Table 1. Characterization of harms caused by violation of genomic privacy.

Category	Harm	Risk	Outcome
Individual	Learn disease risk	High	Psychological
	Secrets	Medium	Psychological Social
	Found for a crime	Medium	Prison Monetary
	Discrimination	Low	Monetary Denied service Denied insurance
	Adverse use of biological data	Low	Prison Psychological
Biological relatives	Learn disease risk	Medium	Psychological
	Secrets	Medium	Psychological Monetary
	Found for a crime	Medium	Prison Monetary
Population	Scorn or embarrassment	Low	Psychological Social
	Use of samples for profit	High	Monetary
Institution	Damage to reputation	High	Impact research Loss of trust Monetary

feet apart, to prevent cross-infection of bacteria that can cause lung infections. However, in *Chadam v. Palo Alto Unified School District*, the child was transferred to a different school solely based on his genetic data, despite not showing any cystic fibrosis symptoms.

In terms of preventing job and health insurance discrimination, GINA appears to work well. Only 333 alleged GINA violations were reported in 2013 compared to 90,000 reports of other forms of job discrimination.¹¹ Most of these violations were from an employer asking for family history, rather than actual genetic information.

Harms to Relatives

Biological relatives of a targeted individual can experience many of the same harms as the reidentified individual, but their risk decreases depending on the amount of DNA shared with the targeted individual. For example, law enforcement searched databases for DNA close enough to be a familial match in a car robbery. This produced several matches, one of whom was the suspect's brother.¹¹ In this case, society might prefer criminals not to have the right to privacy. Furthermore,

an individual's DNA can be used to expose his or her biological relatives' secrets. For example, parents who secretly gave up a baby for adoption can be discovered years later.

Harms to Populations

Harms to populations are classified as those that harm groups of people with the same characteristic. Experiencing societal scorn or embarrassment is one example. People are often scorned for choices regarding their or their child's health (such as smoking or not breastfeeding), so experiencing shame for choices related to genetic information is not outside the realm of possibility. For example, if your child's DNA suggests a high risk for skin cancer, people might criticize your choice to live in an area with high sun exposure. However, no reports of this harm have been recorded.

Individuals might also lose the chance to make money on their own DNA. 23andMe has been reported to make money by selling genetic data.¹³ Notably, DNA is worth more in aggregate, so individuals would likely be paid a minimal sum for their own data.

Harms to Institutions

Entities responsible for the data in a hacked database might also be at high risk for negative outcomes. For example, people might lose trust in institutions, which could impact their research, and these institutions could be required to pay fines for privacy violations.

Discussion

Whether through reidentification or voluntary contributions, individuals might experience the harms outlined in the taxonomy when someone accesses the genetic data linked to their name. If a database is attacked, the database manager might either notify individuals personally if their identity is known, or publish a public notice. However, individuals might not be aware that their data is held in a particular database and thus disregard any notices. Individuals might only become aware of a violation when they experience harm.

The primary hypothesized outcome of genomic privacy violations to individuals is psychological. However, the actual outcome will vary by individual and situation. One person might be unconcerned about discovering their disease risks, while another could experience anxiety over those same risks. In general, research suggests that people don't experience psychological distress over genomic testing results, even in the case of learning their risk of developing Alzheimer's disease.¹⁴ Although discrimination is frequently cited as a possible harm, there are few actual reported cases. There have been cases of individuals being linked to crimes through their DNA in a public database. This is limited to already identified genomic data. Some databases, such as 23andMe, contain only single nucleotide polymorphism data, which isn't compatible with the processed genetic data collected from crime scenes. In addition, 23andMe vows to protect data unless legally compelled to provide it. Therefore, if individuals are comfortable with learning their disease risks, accept that they might learn family secrets, and don't commit crimes, then there seems to be little real risk involved with contributing DNA for scientific research. Note, however, that these risks could change over time. Discrimination could become more prevalent in the future—a child might not be chosen for adoption based on DNA, the care an individual receives from a physician might be negatively impacted by genetic information, and so on.

The situation is murkier for the biological relatives of people who've contributed DNA or for people who were genotyped without their consent (such as children). These individuals could have secrets revealed or learn disease risks that they didn't want to know. Notably, genetic data often doesn't fully predict whether individuals will develop a disease. For example, a pair of monozygotic twins had the same risk for Huntington's

disease, but only one twin developed the disease.¹⁵ This suggests that environmental factors play a significant role in disease development and progression. However, a relative might still experience anxiety over possible disease risks. Whereas individuals might have consented to be in a study, their biological relatives had no say in the publication of their genetic information.

In terms of cost, reidentification is generally more successful and cost effective with structured data and a large amount of identifying information. For example, in the Personal Genome Project, the cost per reidentification is quickly reduced with more records because the primary cost is a single fee for the database used to link quasi-identifying data with an individual's full name. Using this information, intruders can quickly amass a portfolio on those individuals, including current health information, disease risks, family information, personal preferences, and demographics.

Reidentifying individuals from DNA or RNA without additional information is difficult. Data intruders must have knowledge of bioinformatics tools to process raw data and be willing to accept the computational time required for processing. In addition, although phenotypes and disease risks can be inferred, currently there's no simple way to link genomic data back to an individual. However, DNA could be used to link an individual who participated in multiple studies. For example, one database might have a genome connected to demographic information, and a second database might have the same genome linked to psychiatry records. In the genomic repository Sequence Read Archive, however, the primary purpose of most studies is basic biology, which contributes very little useful information about a specific person. In this case, the only information that can be learned about individuals is the way their retinal cells respond to a particular drug.

Data use agreements are one possible way to prohibit data intruders from collecting genomic information that contains demographics. The TCGA, for example, requires two people with eRA Commons accounts to sign a data access request that includes a summary of the proposed research. This likely discourages intruders without significantly impeding legitimate researchers.

Limitations

There are multiple limitations to the work discussed here. First, neither the characterization of harms nor the characterization of the risk of reidentification were systematic reviews. In this project, I performed a preliminary characterization of harms, which could be further assessed with an in-depth and systematic literature review. Next, the cost for reidentification might differ for individual hackers. For some, intensive work or long computational times might not be a barrier. In

addition, more samples could be accumulated in databases over time and better technology could reduce computational time. Furthermore, I ignored sequencing technology errors, which can be a barrier to reidentification. For example, 23andMe has an error rate of one in 7,000 nucleotides. These errors could lead to predictions of incorrect phenotypes, such as an incorrect report of high disease risk when the true risk is actually low. Finally, many genes are pleiotropic and not well understood, so predicted phenotypes might not match actual phenotypes.

Future Directions

This project could extend in several directions. In particular, this work is a precursor to a recently established center at Vanderbilt University. The GetPreCiSe (Genetic Privacy and Identity in Community Settings) Center aims to understand threats to privacy in genomic data, elaborate on efforts to protect privacy, and quantify the probability of reidentification and harm. A further possible direction is to study phenotype prediction from RNA sequencing data. Although this study and others have shown feasibility, determining whether certain samples (treated versus untreated) or cell types provide better variant coverage would be interesting. In addition, this study focused on harms from genomic privacy violations by hackers. The work could be extended to examine other harms, such as individuals in a study not being given access to their data, or physicians not sharing genomic data with patients' biological relatives. Finally, barriers to reidentification, such as data use agreements, could be evaluated to determine their effect on reidentification risk.

Individuals, their relatives, and institutions holding genetic data are at risk for harms if DNA reidentification occurs. Preliminary examination of reidentification risk suggests that data intruders can easily reidentify individuals if the database holding their genetic information also contains metadata. Therefore, institutions might want to focus on providing extra protections for this information rather than for anonymous DNA. However, because we don't know what the future holds and DNA is unique and unchanging, further research should examine the probabilities of future harms. ■

References

1. Y. Erlich and A. Narayanan, "Routes for Breaching and Protecting Genetic Privacy," *Nature Rev. Genetics*, vol. 15, no. 8, 2014, pp. 409–421.
2. M. Gymrek et al., "Identifying Personal Genomes by Surname Inference," *Science*, vol. 339, no. 6117, 2013, pp. 321–324.
3. L. Sweeney, A. Abu, and J. Winn, "Identifying Participants in the Personal Genome Project by Name," arXiv.org, 2013; arxiv.org/abs/1304.7605.
4. Z. Wan et al., "Expanding Access to Large-Scale Genomic Data While Promoting Privacy: A Game Theoretic Approach," *Am. J. Human Genetics*, vol. 100, no. 2, 2017, pp. 316–322.
5. C. Heeney et al., "Assessing the Privacy Risks of Data Sharing in Genomics," *Public Health Genomics*, vol. 14, 2011, pp. 17–25.
6. J.G. Hamilton, M. Lobel, and A. Moyer, "Emotional Distress following Genetic Testing for Hereditary Breast and Ovarian Cancer: A Meta-analytic Review," *Health Psychology*, vol. 28, no. 4, 2009, pp. 510–518.
7. K.G. Anderson, "How Well Does Paternity Confidence Match Actual Paternity? Evidence from Worldwide Non-paternity Rates," *Current Anthropology*, vol. 47, no. 3, 2006, pp. 513–520.
8. L. DeFrancesco, "23andMe Protects Consumers' Data," *Nature Biotechnology*, vol. 33, 2015, p. 1220.
9. D. Frumkin et al., "Authentication of Forensic DNA Samples," *Forensic Science Int'l: Genetics*, vol. 4, no. 2, 2010, pp. 95–103.
10. J.K. Wagner, "Update on Chadam v. Palo Alto Unified School District," *Genomics Law Report*, 24 Jan. 2017; www.genomicslawreport.com/index.php/2017/01/24/update-on-chadam-v-palo-alto-unified-school-district.
11. R.C. Green, D. Lautenbach, and A.L. McGuire, "GINA, Genetic Discrimination, and Genomic Medicine," *New England J. Medicine*, vol. 372, 2015, pp. 397–399.
12. H.T. Greely et al., "Family Ties: The Use of DNA Offender Databases to Catch Offenders' Kin," *J. Law, Medicine, and Ethics*, vol. 34, no. 2, 2006, pp. 248–262.
13. A. Mullard, "23andMe Sets Sights on UK/Canada, Signs up Genentech," *Nature Biotechnology*, vol. 33, 2015, p. 119.
14. R.C. Green et al., "Disclosure of APOE Genotype for Risk of Alzheimer's Disease," *New England J. Medicine*, vol. 361, 2009, pp. 245–254.
15. M.E. Ketelaar, E.M.W. Hofstra, and M.R. Hayden, "What Monozygotic Twins Discordant for Phenotype Illustrate about Mechanisms Influencing Genetic Forms of Neurodegeneration," *Clinical Genetics*, vol. 81, no. 4, 2012, pp. 325–333.

Sara Renee Savage is a postdoctoral trainee in the Department of Biomedical Informatics at Vanderbilt University. Her research interests include all aspects of genetic research, including how genomic and proteomic data can help to develop individualized treatment for patients, and how those patients can be protected from any negative aspect of sharing their data. Savage received a PhD in pharmacology from Vanderbilt University. Contact her at sara.r.savage@vanderbilt.edu.



Improving the Security and Efficiency of Private Genomic Computation Using Server Aid

Marina Blanton | University at Buffalo
Fattaneh Bayatbabolghani | University of Notre Dame

In many genomic applications, especially nonmedical applications, computations are carried out in a server-mediated setting where the server enables joint genomic computations between users. Thus, it's sensible to utilize the server's computational capabilities to aid data protection.

The use of genomic data is rapidly expanding, and the need to protect such highly sensitive data from potential abuse is indisputable. The cost of sequencing one's genome has dramatically decreased in recent years and is continuing to decrease, making such data more readily available for numerous applications, including

- personalized medicine using genomic tests prior to prescribing a treatment to ensure its success,
- paternity testing using DNA data,
- genomic compatibility testing allowing potential or current partners to determine whether their future children are likely to inherit genetic conditions, and
- determining ancestry and building genealogical trees by examining DNA data of many individuals and finding relationships among specific individuals.

Genomic tests are increasingly used for medical purposes to ensure the best treatment. Several services for nontreatment-related use of DNA data have flourished as well—for instance, 23andMe (www.23andme.com),

Ancestry.com (www.ancestry.com), and GenePartner.com (www.genepartner.com)—allowing for various forms of DNA data comparison, be it for building ancestry trees, determining genomic compatibility, or other purposes.

DNA is highly sensitive and must be protected from misuse. It's more sensitive than other types of an individual's biometry because it allows for unique identification and can reveal a plethora of information about the individual. For instance, one can determine the predisposition of an individual and his or her relatives to medical conditions, thus exposing these relatives' information as well. Furthermore, our understanding of genomes is continuously growing, and exposure of DNA data might lead to consequences that we can't even anticipate today. For this reason, the security community has recognized the need to protect the privacy of DNA data when it's being used in genomic computations.¹⁻⁴

Although protecting the privacy of genomic data is important for all applications, in our opinion, individuals

are less able to influence the way medical procedures are conducted than they are services in which they voluntarily participate. For example, individuals considering using a gene-based matchmaking online service to meet a potential partner might initially be reluctant to share their DNA data with the service (or the partner) for the purpose of genetic compatibility tests. However, if the service guarantees that no sensitive information about their DNA (other than the intended outcome) will be revealed to any party throughout the computation, they might revisit the decision to participate. Thus, this article focuses on nontreatment-related applications in which individuals might choose to participate.

Because user interaction and genomic computation in such applications are normally facilitated by some service provider or third party, such service providers are uniquely positioned to aid individuals with private computation on their sensitive genomic data. Thus, instead of invoking traditional two-party privacy-preserving computation mechanisms, participants can now pursue the server-mediated setting and utilize the server to lower their cost of computation. Therefore, we can view computation as being carried out between two users, Alice and Bob, and a server that provides no input into the computation and learns no output but whose involvement allows for higher security guarantees and lower computational costs.

Genomic Background

A genome represents an individual's complete hereditary information. Information extracted from the genome can take different forms. One type is a single nucleotide polymorphism (SNP), which corresponds to a well-known variation in a single nucleotide (a nucleotide can be viewed as a simple unit represented by a letter A, C, G, or T). Normally, each SNP is referenced by a specific index and its value in an individual is represented as a bit, while representations consisting of three values (0, 1, and 2) are used as well. The two possible nucleotide variations at a given position are called alleles. A mutation can occur at a single allele inherited from one parent (a minor mutation) or two alleles inherited from both parents (a major mutation); the values 0, 1, and 2 indicate no, minor, and major mutations, respectively. A binary SNP representation uses 0 for no mutations and 1 for either kind of mutation (minor or major).

Another type of data extracted from a genome is based on short tandem repeats (STRs), which occur when a short region consisting of two or more nucleotides is repeated and the occurrences are adjacent to one another. Unrelated individuals are likely to have a different number of repeats of a given STR sequence in certain regions in their DNA; thus, STRs are often used for identity testing or testing between close relatives.

In this article, we focus on two specific genomic tests: paternity and genetic compatibility.

Paternity Test

This test is normally based on STRs. One's STR profile consists of an ordered sequence of n two-element sets $S = \langle \{x_{1,1}, x_{1,2}\}, \{x_{2,1}, x_{2,2}\}, \dots, \{x_{n,1}, x_{n,2}\} \rangle$, where each value corresponds to the number of repeats of a specific STR sequence at specific locations in the genome. For each STR i , one of $x_{i,1}$ and $x_{i,2}$ is inherited from the mother and one from the father. Thus, in the paternity test with a single parent, there are two STR profiles $S = \langle \{x_{i,1}, x_{i,2}\} \rangle_{i=1}^n$ and $S' = \langle \{x'_{i,1}, x'_{i,2}\} \rangle_{i=1}^n$ corresponding to the child and the contested father, respectively. To determine whether S' corresponds to the child's father, the test computes whether, for each i , the set $\{x_{i,1}, x_{i,2}\}$ contains (at least) one element from the set $\{x'_{i,1}, x'_{i,2}\}$. This means that the intersection of sets $\{x_{i,1}, x_{i,2}\}$ and $\{x'_{i,1}, x'_{i,2}\}$ must be nonempty for each i .

Genetic Compatibility Test

A genetic compatibility test is used when potential or existing partners would like to determine the possibility of transmitting to their children a genetic disease with Mendelian inheritance. In particular, if a minor mutation is present, it often has no impact on one's quality of life, but with a major, the disease manifests itself in severe forms. If both partners silently carry a single mutation, they have a noticeable chance of conceiving a child carrying the major variety. Thus, a genetic compatibility test for a given genetic disease would look for the presence of minor mutations in both partners.

Screening for a disease consists of testing whether a specific mutation string appears at a specific location in the DNA. Future tests for more complex diseases might look for the presence of multiple mutations, but for the purpose of this article, we assume that the test's output is a bit (that is, the individual is tested as a disease carrier or not). If both partners test positive, then the outcome of the genetic compatibility test will be treated as positive; otherwise, it's negative.

Security Guarantees

The solutions we describe here achieve several security properties.

Protection of Private Data

From a security viewpoint, it's typically expected that no information of any kind about private data is available to the participants during privacy-preserving computation. That is, users Alice and Bob input their private genomic data into the computation and learn the computed outcome but can't infer any additional information about one another's private data. Adding the server

to the computation must not compromise any security guarantees, which means that the server learns no information of any kind about Alice's or Bob's data.

Adversarial Model

Security literature typically distinguishes between two standard ways of modeling the behavior of secure computation participants: they can be either *semihonest* (passive or honest but curious) or *malicious* (active). A semihonest participant correctly follows the protocol specification but attempts to learn additional information by analyzing the messages received during the execution, whereas a malicious participant can arbitrarily deviate from the protocol specification in an attempt to learn unauthorized information.

The server can typically be expected not to deviate from its prescribed behavior, as it would lose its reputation and, consequently, revenue if any attempts at cheating become known. A dishonest insider observing the server's networks and stored data should be unable to recover any sensitive information about users, and the company usually detects and mitigates attempts to tamper with the software running on the server. Thus, we assume that the server is semihonest. If the server can't be trusted to follow the computation, a stronger security model that tolerates a malicious or covert server (one that misbehaves but doesn't want to be detected) needs to be applied. However, this is beyond our article's scope.

We similarly assume that the server won't collude with users (putting its reputation at risk) or let users affect its operation. On the other hand, users Alice and Bob might not know each other well (for instance, if they met through a matchmaking website), so it's reasonable for them to be cautious. Therefore, they might want to engage in a protocol that ensures correctness and privacy even when the other user is malicious.

We note that alternative adversarial models for server-aided two-party computation have been treated in the literature (such as in "Whitewash: Outsourcing Garbled Circuit Generation for Mobile Devices"⁵ and others) and can be pursued if different or stronger guarantees are desired.

Input Certification

Another important consideration from a security standpoint is enforcing correct (that is, truthful) inputs to be entered into the computation. This requirement is outside the traditional security model for secure multiparty computation (even in the presence of fully malicious actors) and normally isn't addressed, but it becomes important in the context of genomic computation. This is because, for certain types of genomic tests, it's very easy for one participant to modify his or her inputs

to learn sensitive information about the other party's genetic conditions. For example, consider genetic compatibility tests. If the partners can each separately evaluate their DNA for a specific disease's fingerprint, the joint computation can consist of a simple AND of the bits provided by both parties (for one or more diseases). Now if a malicious participant sets his or her input bits for all tested diseases to 1 and the outcome is positive, the participant learns that the other party is a carrier for a specific medical condition (or at least one condition from the set of specific conditions, depending on how the computation is set up). We thus want to prevent malicious participants from modifying their inputs to genomic computation. We preserve integrity of inputs by requiring them to be certified by certification authorities, such as medical facilities, without disclosing the private values that were certified.

Fairness

Fairness is another desired security property of a solution that realizes privacy-preserving computation; it guarantees that if one participant prematurely quits the computation, he or she can't learn more information about the data used in the computation (for instance, by recovering partial or complete output) than the other protocol participants. Because of the nature of the data used in genomic computations, achieving fairness should be an essential part of privacy-preserving solutions for this domain.

Secure Server-Aided Computation

Having spelled out the security guarantees that we'd like to see in a secure solution, we're now ready to proceed with building such a solution. The first construction we present addresses all security guarantees except enforcing input correctness via input certification. This feature is added in a later construction.

Initial Secure Solution

Recall that the goal was to employ a server that facilitates communication between users Alice and Bob to improve the efficiency and security of privacy-preserving computation. There are several well-researched secure computation techniques that have seen drastic performance improvements during recent years. One of them is *garbled circuit evaluation*, which can be used to securely evaluate any desired function by two parties Alice and Bob on their private inputs. In the past few years, several publications modified garbled circuit evaluation techniques to work in the two-party server-aided setting that we pursue, or more generally in a three-party setting, with the goal of maintaining its efficiency while offering stronger security guarantees.⁶⁻⁸ This is the approach our solutions use here.

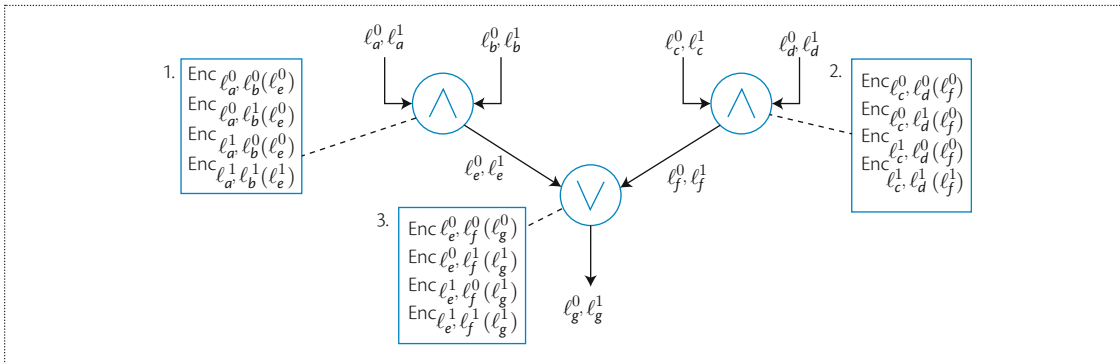


Figure 1. A sample garbled circuit with AND and OR gates produced by a circuit garbler. Tables 1, 2, and 3 are stored in a randomly permuted order.

There are alternative approaches to secure two-party or server-aided two-party computation, the most popular of which are techniques based on *homomorphic encryption*. Different variants of homomorphic encryption are known based on their ability to perform transformations on encrypted data without interaction; the most widely used schemes are public-key crypto systems. Because of the fundamental differences in how underlying techniques for secure multiparty computation work, the cost of elementary operations (such as for Boolean and integer arithmetic) significantly differ among frameworks. For the computations we consider in this article, garbled circuit-based techniques offer a performance advantage, and thus we don't address solutions based on homomorphic encryption here.

Before we proceed, we provide a high-level description of how garbled circuit evaluation works. The use of garbled circuits lets two parties, Alice and Bob, securely evaluate any function f of their choice. Given an arbitrary function $f(x_A, x_B)$ that takes private inputs x_A and x_B from Alice and Bob, respectively, the parties first represent the function as a Boolean circuit. One party, say Alice, acts as a circuit generator and creates a garbled representation of the circuit. She does so by associating each binary wire with two random labels ℓ^0, ℓ^1 (that is, one label corresponds to 0 and the other to 1) and creating binary gates in such a way that, given two labels for its input wires, one can recover a label associated with the output wire. Figure 1 provides a simple example of a garbled circuit with AND and OR gates.

The other party, Bob, acts as a circuit evaluator and evaluates the circuit in its garbled representation, gate by gate, without knowing the meaning of the labels that he handles during the evaluation. Before evaluation can start, Bob needs to obtain the random labels corresponding to both Alice's and his inputs. Because Alice knows the labels and their meaning, she can just send the labels corresponding to her input bits to Bob.

The labels corresponding to Bob's private input are communicated to Bob using a cryptographic primitive known as a 1-out-of-2 *oblivious transfer*. It allows him to obtain one out of two wire labels corresponding to each of his input bits from Alice without revealing anything to Alice. After evaluating the circuit on private inputs, the output labels can be mapped to their meaning and revealed to either or both parties.

The fastest version of garbled circuit evaluation is secure in the presence of a semihonest garbler and a malicious evaluator (assuming the oblivious transfer is resilient to malicious receivers). However, when either participant can be malicious, more complex techniques need to be involved. These techniques typically result in two orders of magnitude slowdown compared to the version secure in the presence of semihonest participants. Thus, the goal is to build on the fast solution and maintain efficiency by using another participant (the server).

The solution we present is general and works for any type of computation. Figure 2 depicts the interaction between Alice, Bob, and the server for securely evaluating a function f of their choice on Alice's and Bob's private inputs.

Recall that the server is semihonest and follows the computation, whereas Alice and Bob can deviate from the prescribed computation in the attempt to learn unauthorized information. This means that if we charge the server with the task of creating a garbled circuit for f , we know that it will be formed correctly and the underlying security guarantees will be preserved. As part of circuit garbling, Alice also learns the garbled label pairs ℓ_i^0, ℓ_i^1 for each wire i corresponding to her input bits, but not for other circuit wires. Once the circuit is created, the server communicates it to Bob (step 1).

Bob will evaluate the circuit, but before evaluation can begin, he needs to obtain one label for each wire corresponding to his and Alice's input bits. To obtain

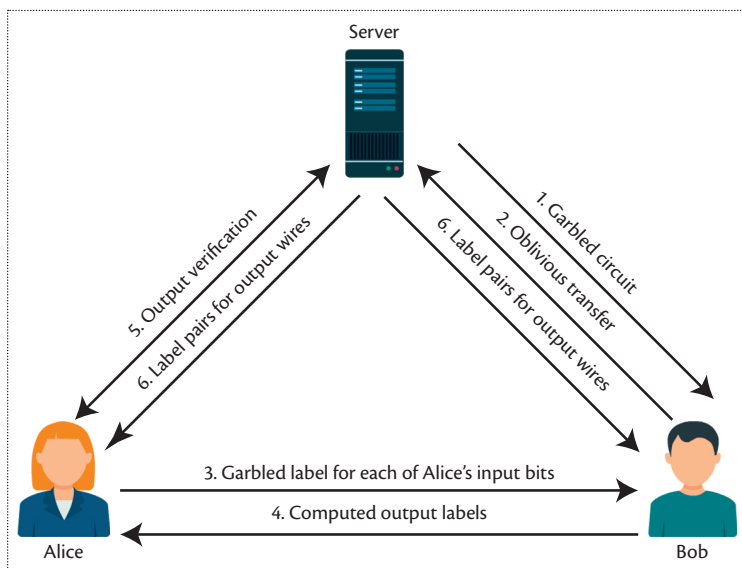


Figure 2. Initial construction for server-aided secure two-party computation. Users Alice and Bob engage in joint secure computation on their private data with the help of a server.

the labels corresponding to his input bits, he engages in oblivious transfer with the server as in the original garbled circuit approach (step 2). Alice, on the other hand, knows the label pairs corresponding to her input bits and can select and send to Bob (step 3) the appropriate label corresponding to her input (that is, she sends ℓ_i^0 if her i th input bit is 0, and ℓ_i^1 otherwise). Note that if Alice misbehaves and doesn't send correct labels, the computation won't be able to proceed and neither party will learn the output. Similarly, if Bob doesn't follow the prescribed computation, the execution is aborted and neither party learns any information.

Once circuit evaluation is complete, Bob has (random) labels associated with the output bits but doesn't know what they mean. We can't simply have the server send the label pairs for the output wires to Bob, as this will allow Bob to interpret the output for himself but deny output to Alice (or provide her with incorrect output). Thus, Bob is instructed to first send the computed labels to Alice (step 4). Alice uses the server's help to verify that she has valid output labels—that is, labels that the server created (step 5). This is done by having the server send hashes of the labels in a random order for each output wire to Alice. Alice consequently hashes the labels she received from Bob and checks whether all of them appear on the server's list. This will allow Alice to verify that she indeed has a valid label for each output wire without learning the meaning of that label.

Once verification is complete, Alice and Bob obtain the label pairs from the server in the original order (step 6), which allows both of them to interpret the

output bits and learn the result of the computation. Note that fairness is achieved in that either both parties learn the correct output or neither party obtains any information.

Adding Input Certification

We next consider an enhanced security setting in which Alice's and Bob's inputs are validated before computation. Recall that this is done to eliminate the possibility of a malicious user manipulating his or her input to learn the other user's sensitive genomic information. As mentioned earlier, it was previously not known how this feature could be realized in regular secure two-party computation, but the use of the server allows us to support this new security property.

In this version of the computation, we assume that Alice's and Bob's inputs are certified by one or more certification authorities and Alice and Bob possess digital signatures on their inputs issued by these authorities. In the context of genomic tests, the signatures will come from medical facilities that assemble their genomic data or perform the relevant genomic tests.

For the purposes of this solution, it's necessary that the signatures that Alice and Bob obtain from certification authorities have certain special properties. In particular, because they will need to prove some statements about their input without revealing the input itself, the underlying signature scheme must support proving statements about a signed message without revealing the message. Such signature schemes are known as *signature schemes with protocols*. The use of such signatures typically involves first proving that one has a signature on a message, the value of which stays protected, followed by proving additional statements about the message. In the solution we describe, Alice and Bob will need to prove statements that consist of equality, disjunction (OR), and conjunction (AND). Researchers had long ago discovered how to realize such proofs without revealing any additional information about the value used in the proof statements; these are called *zero-knowledge proofs of knowledge* because no information about the relevant values is revealed beyond the statement of the proof itself.

The structure of the solution that uses certified inputs remains the same as in the first construction. Only the steps in which Alice and Bob enter their inputs into the computation need to be modified (steps 2 and 3 of the original protocol). For simplicity, let's assume that all of Alice's and Bob's inputs are to be verified.

We start with Bob's verification mechanism. Bob engages in the oblivious transfer in the role of the receiver with the server for each bit of his input. After proving that he possesses a signature from a certification authority, Bob proves that the signed bit was 0 and the

key he formed for the purposes of the oblivious transfer corresponds to 0, or that the signed bit was 1 and the key he's using in the oblivious transfer corresponds to 1. Note that in addition to using equality to 0 and 1, this proof also includes two conjunctions and one disjunction. This statement can also be generalized to the cases in which a single signature on a multibit message is used to provide several input bits into the computation.

Alice's proof to the server after circuit generation has a similar logic. For each bit of her input, she shows that either the signed bit was 0 and her input label will be ℓ_i^0 , or the signed bit was 1 and her input label will be ℓ_i^1 . Information about the label ℓ_i^0 or ℓ_i^1 is encoded using a cryptographic commitment that ensures that the server doesn't learn anything about that value and, at the same time, prevents Alice from changing the value when she consequently interacts with Bob in step 3. (Commitment is a cryptographic construction that protects the secrecy of the committed value but prevents the sender from later opening the commitment to a value different from the one used to form the commitment.) If verification is successful for each bit of Alice's input, the server forwards to Bob the label commitments it received from Alice. Now when Alice sends her input labels to Bob in step 3, she also shows that the labels correspond to the opening of the commitments she supplied to the server earlier. This constitutes the proof that the input Alice provides into the protocol was indeed signed by a certification authority.

Secure Genomic Computations

We're now ready to proceed with showing how Alice and Bob can use the solution described in the previous section to realize secure server-aided genomic computation. For both paternity and compatibility tests, we assume that Alice and Bob have information extracted from their respective genomes, which they privately store and input into the computation.

Paternity Test

In paternity tests, participants often don't trust each other and might be inclined to tamper with the computation to influence the result. However, it's difficult to learn the other party's genetic information by modifying one's input into the function. In particular, recall that the output of a paternity test is a single bit, which indicates whether the exact match was found. If a malicious participant engages in the computation with the same victim multiple times and modifies the input in an attempt to discover the victim's genomic data, the single-bit output doesn't help the attacker learn how his or her inputs are to be modified to be closer to the victim's input. Thus, we establish that the first security setting—with malicious users and a semihonest server,

but without input certification—is suitable for running paternity tests.

This test would normally be run between an individual and a contested father of that individual according to the corresponding computation we described earlier. We implement the computation using a Boolean circuit as follows: to compute whether a set intersection of $\{x_{i,1}, x_{i,2}\}$ and $\{x'_{i,1}, x'_{i,2}\}$ for any given i is not empty, we could XOR the vectors $\langle x_{i,1}, x_{i,2}, x_{i,1}, x_{i,2} \rangle$ and $\langle x'_{i,1}, x'_{i,1}, x'_{i,2}, x'_{i,2} \rangle$ and compare each of the four elements in the resulting vector to 0. Note that because with garbled circuits, XOR gates can be evaluated much faster and with no communication overhead,⁹ it's desirable to minimize the number of non-XOR gates in a circuit design. The (in)equality to 0 testing is performed using $k - 1$ OR gates, where k is the bit length of all $x_{i,j}$ s and $x'_{i,j}$ s. Then, if the result is 1 for any of the four elements, it means that the corresponding values differ.

We next compute the AND of the results of the four inequality tests. If the result is 1, it means that all four elements are not equal; that is, the intersection is empty. Finally, we OR the resulting bits across all i s and output the complement of the computed bit. If the result of the OR is 1, it means that at least one set intersection was empty and the test failed; thus, the output is 0. Figure 3a depicts this computation.

Genetic Compatibility Test

A genetic compatibility test consists of two users, Alice and Bob, evaluating the possibility of their children inheriting at least one recessive genetic disease. We assume that they agree on a list of genetic diseases to be included in the computation (this list can be standard, for example, suggested by the server or recommended by a medical association). Note that testing for a specific genetic disease is only meaningful if both parties want to be tested for it; thus, we assume that Alice and Bob can reconcile the differences in their lists.

To maximize privacy, we would like the computation to be as conservative as possible. Thus, given a list of genetic diseases D , we design the function to be evaluated to first run a compatibility test for each disease and, if at least one test is positive, output 1; otherwise, output 0. That is, the function can be interpreted as producing 1 if Alice and Bob's children have a chance of inheriting the major variety for at least one of the tested diseases, and 0 otherwise. Evaluating this function can be viewed as the first step in Alice and Bob's interaction. If the output was 1, they might jointly decide to run more specific computation to determine the responsible disease(s).

So, for each $d_i \in D$, Alice and Bob will need to locally determine whether they are carriers of d_i before the joint computation begins. Thus, their input into the joint

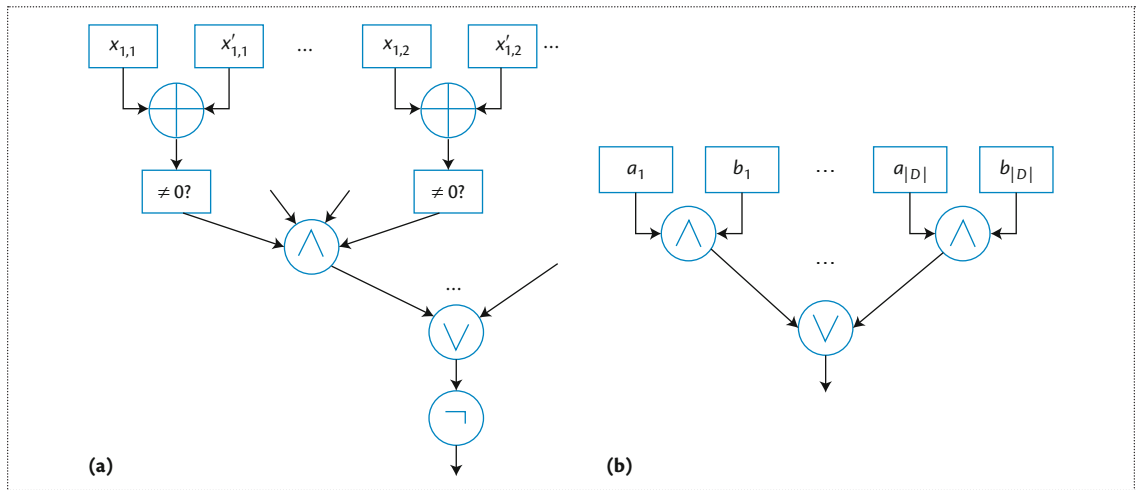


Figure 3. Realization of circuits for (a) paternity and (b) genetic compatibility tests. The computation is represented as Boolean circuits and consequently garbled and evaluated in the garbled form.

computation consists of $|D|$ bits each, and the result is 1 if there's at least one $d_i \in D$ such that Alice's and Bob's i th input bits are both 1. This computation can be realized as a simple circuit consisting of $|D|$ AND and $|D|-1$ OR gates. The i th AND gate computes the AND of the i th input bit of Alice and Bob, and the OR gates are applied to the results to determine whether at least one of the bits was set. Figure 3b depicts this computation.

Notice that it's easy for malicious Alice or Bob to learn sensitive information about the other party by using certain inputs. That is, if a malicious user sets all his or her input bits to 1, he or she will be able to learn whether the other party is a carrier of at least one disease in D . This poses substantial privacy concerns, particularly for matchmaking services that routinely run genetic compatibility tests between many individuals. Thus, we require that Alice and Bob certify the results of testing for each genetic disease on the list and enter certified inputs into the computation. (Note that a medical facility that performs sequencing can also certify the test results; alternatively, the medical facility performing test certification can require that the genome on which the test is based is certified by the facility performing sequencing.) The stronger security setting with certified inputs is the most appropriate for this computation.

A signature that certifies the outcome of testing for a particular disease must come with the name of the corresponding disease. Then while the (certified) test results that Alice and Bob enter into the computation remain private, the name of the disease included in each signature is revealed to the server. This will allow the server to ensure that Alice and Bob enter information about the same diseases in the computation and can't deviate from the prescribed behavior in an attempt to cheat. Even when the users don't want to reveal the

names of the diseases on their list to the server, it's still possible for the server to enforce compliance with correct computation by means of shared commitments and additional zero-knowledge proofs.

Note that although the circuit used in compatibility computation is rather simple, the overall solution is more involved owing to the use of input certification not present in paternity testing. Also, by having each party independently certify the results of disease testing ahead of time, we reduce the cost of joint computation, and the results of disease testing (including certification) can be reused an unlimited number of times.

Other Types of Computation

Because the constructions we describe in this article are general, they can be used for genomic and nongenomic applications, including medical purposes. Furthermore, when it's crucial to guarantee input integrity, using the second construction with certified inputs would be beneficial. For instance, when personalized medicine computation relies on a third-party service provider, our security model might apply. Utilizing certified inputs will guarantee input correctness, which is especially important here because using incorrect data puts patients' health at risk.

Performance Evaluation

Here, we demonstrate the presented techniques' performance for privacy-preserving paternity and compatibility computation. The implementation—available at github.com/fattaneh88/PETS—was done in C/C++ with Miracl library (www.certivox.com/miracl) for large number arithmetic and JustGarble library (cseweb.ucsd.edu/groups/justgarble) for garbled circuit implementation. Each party (Alice, Bob, and the server) ran

Table 1. Performance of secure paternity and compatibility tests.

Party	Paternity test			Compatibility test		
	Computation (ms)	Communication (Kbytes)		Computation (ms)	Communication (Kbytes)	
		Sent	Received		Sent	Received
Alice	0	3.7	0.06	1,853	34.4	0.06
Bob	717	31.7	56.9	1,542	36.4	3.0
Server	457	53.3	31.7	2,859	2.9	70.6

on a 3.2-GHz machine using a single core. Note that this is a modest setup because the server is expected to be more powerful in practice and using multiple cores can significantly reduce computation time.

Paternity Test

Implementation of the paternity test corresponds to its earlier description without input certification. The inputs for both Alice and Bob consisted of 13 two-element sets, in which each element was nine bits long. In practice, the US CODIS system uses 13 pairs, whereas the European SGM Plus identification method utilizes 10; thus, we chose 13 for the experiments. Furthermore, because no element in any set exceeds 500,¹⁰ its length is set to nine bits. The resulting circuit had 935 gates, out of which 468 were cheap XOR gates. Table 1 gives the experiment’s results. Note that we list computation time and communication size separately because the speed of communication channels can greatly vary.

As Table 1 indicates, this protocol is well-suited for settings in which one user is very resource constrained. Also, compared to traditional two-party garbled circuits computation in the presence of malicious participants, this solution reduces both computation and communication for the participating users by at least two orders of magnitude (even if it might appear that Bob has a heavier load than the server).⁶ This is primarily because many circuits need to be garbled and evaluated in the two-party setting with malicious participants to ensure that a correct circuit is used for function evaluation. This solution also favorably compares to alternative server-aided two-party constructions including Whitewash.⁵

Genetic Compatibility Test

Recall that the genetic compatibility test is run in the setting where Alice’s and Bob’s inputs must be certified. We chose the variant of the solution that reveals

the list of diseases D to the server (that is, a standard list is used). In this experiment, we set the number of tested diseases $|D| = 10$ and thus Alice and Bob provided 10 input bits into the circuit accompanied by 10 signatures (any desired value of $|D|$ will work, with performance linear in $|D|$). The circuit consisted of only 19 gates. Table 1 shows this test’s performance. Input certification contributes most of the solution’s overhead (98 to 100 percent, depending on the party), but it’s still on the order of one to three seconds for all parties. More than 95 percent of Alice’s and Bob’s work can be pre-computed and performed ahead of time.

If input certification needs to be used with another functionality (for instance, genetic compatibility with a different number of diseases, paternity test, and so on), its performance in our secure execution framework can be estimated based on the input size in bits and performance of the compatibility tests in Table 1 (for 10 input bits). Because such protocols’ overall runtime is heavily dominated by the time to obliviously verify signatures, varying the circuit size doesn’t significantly impact the overall time.

The need to protect genomic data in a variety of applications is undeniable today, and serves as a strong motivation for applying secure computation techniques to genomic computations. In this article, we describe an efficient and secure construction for server-aided two-party computation where both users can act maliciously, which is consequently enhanced with input certification to guarantee input integrity. Input certification in the context of general secure two-party computation is new, and such research results are only starting to appear. Additional information about the techniques presented in this article can be found in “Efficient Server-Aided Secure Two-Party Function Evaluation with Applications to Genomic Computation.”⁶ ■

References

1. E. Ayday et al., "Protecting and Evaluating Genomic Privacy in Medical Tests and Personalized Medicine," *ACM Workshop Privacy in the Electronic Society (WPES 13)*, 2013, pp. 95–106.
2. P. Baldi et al., "Countering GATTACA: Efficient and Secure Testing of Fully-Sequenced Human Genomes," *ACM Conf. Computer and Communications Security (CCS 11)*, 2011, pp. 691–702.
3. F. Bruekers et al., "Privacy-Preserving Matching of DNA Profiles," *IACR Cryptology ePrint archive report 2008/203*, 2008; eprint.iacr.org/2008/203.pdf.
4. E. De Cristofaro, S. Faber, and G. Tsudik, "Secure Genomic Testing with Size- and Position-Hiding Private Substring Matching," *ACM Workshop Privacy in the Electronic Society (WPES 12)*, 2012, pp. 107–118.
5. H. Carter, C. Lever, and P. Traynor, "Whitewash: Outsourcing Garbled Circuit Generation for Mobile Devices," *Proc. Ann. Computer Security Applications Conference (ACSAC 14)*, 2014, pp. 266–275.
6. M. Blanton and F. Bayatbabolghani, "Efficient Server-Aided Secure Two-Party Function Evaluation with Applications to Genomic Computation," *Privacy Enhancing Technologies Symp. (PETS 16)*, 2016, pp. 144–164.
7. Y. Ishai et al., "Secure Computation with Minimal Interaction, Revisited," *Advances in Cryptography (CRYPTO 15)*, 2015, pp. 359–378.
8. P. Mohassel, M. Rosulek, and Y. Zhang, "Fast and Secure Three-Party Computation: The Garbled Circuit Approach," *Proc. ACM Conf. Computer and Communications Security (CCS 15)*, 2015, pp. 591–602.

9. V. Kolesnikov and T. Schneider, "Improved Garbled Circuit: Free XOR Gates and Applications," *Proc. Int'l Colloquium Automata, Languages and Programming (ICALP 08)*, 2008, pp. 486–498.
10. S. El-Alfy and A. El-Hafez, "Paternity Testing and Forensic DNA Typing by Multiplex STR Analysis Using ABI PRISM 310 Genetic Analyzer," *J. Genetic Engineering and Biotechnology*, vol. 10, no. 1, 2012, pp. 101–112.

Marina Blanton is an associate professor in the Department of Computer Science and Engineering at the University at Buffalo, the State University of New York. Her research interests include applied cryptography, security, and privacy. Blanton received a PhD in computer science from Purdue University. She's a Senior Member of ACM and IEEE. Contact her at mblanton@buffalo.edu.

Fattaneh Bayatbabolghani received a PhD in computer science and engineering from the University of Notre Dame. Her primary research interest is designing general and efficient privacy preserving solutions motivated by real-world applications. Contact her at fbayatba@nd.edu.



Read your subscriptions through the myCS publications portal at <http://mycs.computer.org>

IEEE  computer society

PURPOSE: The IEEE Computer Society is the world's largest association of computing professionals and is the leading provider of technical information in the field.
MEMBERSHIP: Members receive the monthly magazine *Computer*, discounts, and opportunities to serve (all activities are led by volunteer members). Membership is open to all IEEE members, affiliate society members, and others interested in the computer field.
OMBUDSMAN: Email ombudsman@computer.org.
COMPUTER SOCIETY WEBSITE: www.computer.org

Next Board Meeting: 12–13 November 2017, Phoenix, AZ, USA

EXECUTIVE COMMITTEE

President: Jean-Luc Gaudiot
President-Elect: Hironori Kasahara; **Past President:** Roger U. Fujii; **Secretary:** Forrest Shull; **First VP, Treasurer:** David Lomet; **Second VP, Publications:** Gregory T. Byrd; **VP, Member & Geographic Activities:** Cecilia Metra; **VP, Professional & Educational Activities:** Andy T. Chen; **VP, Standards Activities:** Jon Rosdahl; **VP, Technical & Conference Activities:** Hausi A. Müller; **2017–2018 IEEE Director & Delegate Division VIII:** Dejan S. Milošević; **2016–2017 IEEE Director & Delegate Division V:** Harold Javid; **2017 IEEE Director-Elect & Delegate Division V-Elect:** John W. Walz

BOARD OF GOVERNORS

Term Expiring 2017: Alfredo Benso, Sy-Yen Kuo, Ming C. Lin, Fabrizio Lombardi, Hausi A. Müller, Dimitrios Serpanos, Forrest J. Shull
Term Expiring 2018: Ann DeMarle, Fred Douglass, Vladimir Getov, Bruce M. McMillin, Cecilia Metra, Kunio Uchiyama, Stefano Zanero
Term Expiring 2019: Saurabh Bagchi, Leila De Floriani, David S. Ebert, Jill I. Gostin, William Gropp, Sumi Helal, Avi Mendelson

EXECUTIVE STAFF

Executive Director: Angela R. Burgess; **Director, Governance & Associate Executive Director:** Anne Marie Kelly; **Director, Finance & Accounting:** Sunny Hwang; **Director, Information Technology & Services:** Sumit Kacker; **Director, Membership Development:** Eric Berkowitz; **Director, Products & Services:** Evan M. Butterfield; **Director, Sales & Marketing:** Chris Jensen

COMPUTER SOCIETY OFFICES

Washington, D.C.: 2001 L St., Ste. 700, Washington, D.C. 20036-4928
Phone: +1 202 371 0101 • **Fax:** +1 202 728 9614 • **Email:** hq.ofc@computer.org
Los Alamitos: 10662 Los Vaqueros Circle, Los Alamitos, CA 90720
Phone: +1 714 821 8380 • **Email:** help@computer.org

MEMBERSHIP & PUBLICATION ORDERS

Phone: +1 800 272 6657 • **Fax:** +1 714 821 4641 • **Email:** help@computer.org
Asia/Pacific: Watanabe Building, 1-4-2 Minami-Aoyama, Minato-ku, Tokyo 107-0062, Japan • **Phone:** +81 3 3408 3118 • **Fax:** +81 3 3408 3553 • **Email:** tokyo.ofc@computer.org

IEEE BOARD OF DIRECTORS

President & CEO: Karen Bartleson; **President-Elect:** James Jefferies; **Past President:** Barry L. Shoop; **Secretary:** William Walsh; **Treasurer:** John W. Walz; **Director & President, IEEE-USA:** Karen Pedersen; **Director & President, Standards Association:** Forrest Don Wright; **Director & VP, Educational Activities:** S.K. Ramesh; **Director & VP, Membership and Geographic Activities:** Mary Ellen Randall; **Director & VP, Publication Services and Products:** Samir El-Ghazaly; **Director & VP, Technical Activities:** Marina Ruggieri; **Director & Delegate Division V:** Harold Javid; **Director & Delegate Division VIII:** Dejan S. Milošević

revised 31 May 2017





Inference Attacks against Kin Genomic Privacy

Erman Ayday | Bilkent University

Mathias Humbert | Swiss Data Science Center

Genomic data poses serious interdependent risks: your data might also leak information about your family members' data. Methods attackers use to infer genomic information, as well as recent proposals for enhancing genomic privacy, are discussed.

Individuals desiring to control their personal data face significant *interdependent privacy risks*—risks that involve the leakage of one's personal data due to data shared by other individuals. With recent advances in whole genome sequencing, genomic data in particular poses serious interdependent privacy risks.

Genomic data has many unique characteristics: it is highly valuable, is an individual's distinctive fingerprint, rarely changes throughout an individual's lifetime, is non-revocable, and includes sensitive information about an individual (such as disease status or physical characteristics).^{1,2} But, the main reason genomic data poses interdependent privacy risks is that it's correlated within family members. Thus, one person's genome-related data (for instance, raw genome, variant call format file, genomic test results, or aggregate statistics) might leak information about the genome-related data of his or her family members.

This issue goes all the way back to the DNA dragnets that first raised serious concerns among privacy advocates. Here, we present recent developments on the information security front, including

- how attackers can infer an individual's genomic data from the partial genomes of his or her family

members, background knowledge about genomics (simple statistics, high-order correlations, and so on), and the individual's phenotypic information;

- how attackers can determine an individual's membership in a particular genomic dataset (for example, a beacon) from only the results of basic queries to that dataset and partial genomic knowledge about the individual's family members;
- how attackers can deanonymize the deidentified genomes in a public dataset by using the kinship information; and
- how attackers can efficiently infer kinship from public anonymous genomic databases.

Background

Before discussing these developments in further detail, we introduce the important genomic elements relevant to this article.

Genomic Elements

The vast majority (approximately 99.5 percent) of DNA is similar among human beings. Of the remaining 0.5 percent, the most common variant in the human genome is called a *single nucleotide polymorphism*

(SNP). An SNP is a variation of a nucleotide at a specific position in the genome that affects at least 1 percent of individuals in a given population (typically referred to as a *common SNP*). As of November 2016, the Single Nucleotide Polymorphism database (dbSNP; www.ncbi.nlm.nih.gov/projects/SNP) lists approximately 154 million common SNPs in human beings. An SNP, like any other base pair, has two nucleotides. Each nucleotide can take either the major or minor allele. The major allele is the most commonly observed nucleotide in a given population, whereas the minor is the rare nucleotide. If we represent the major allele as B and the minor allele as b , an SNP can take values in $\{BB; Bb; bb\}$, where B and b take values in the alphabet $\{A; T; G; C\}$. SNP values are also known as an individual's genotype.

SNPs are especially sensitive from a privacy perspective because many of these polymorphic positions are associated with severe diseases. For example, carrying particular values at two SNPs (rs7412 and rs429358) on the Apolipoprotein E (ApoE) gene indicates an increased risk for Alzheimer's disease.

Due to genetic inheritance laws, family members share more SNPs than unrelated individuals. Thus, SNPs can be used to infer kinship between two individuals. Moreover, kinship information can infer hidden (or unknown) SNP values of relatives. Also commonly used for kinship inference are *short tandem repeats* (STRs). STRs consist of two to 13 nucleotides repeated numerous times in a row on the DNA strand. For instance, GATAGATAGATA is an STR of period four repeating three times. STRs have a higher mutation rate than other areas of DNA, leading to high genetic diversity.

Reproduction

Mendel's first law of inheritance—the law of segregation—states that alleles are passed independently from parents to child for different meioses (children). Moreover, at each SNP position, the child inherits one allele from the mother and one from the father. Each allele from the parents is randomly selected from their two alleles with probability 0.50. Hence, if the mother has an SNP value of BB and the father has an SNP value of Bb , the child will inherit an SNP equal to BB or Bb , both with probability 0.50. If both parents carry an SNP equal to Bb , then the child's SNP will take a value of BB or bb with probability 0.25, and value Bb with probability 0.50. Finally, given both parents' genomes, the child's genome is independent of all other ancestors' genomes.

One exception to Mendel's law is the Y chromosome. The Y chromosome is inherited (almost) intact along a family's male line. Thus, a father's Y chromosome is the same as his son's Y chromosome. Due to this property, multiple genealogy companies offer services to reunite

distant patrilineal relatives by genotyping a few dozen highly polymorphic STRs across the Y chromosome (called Y-STRs).

Another exception to the law of segregation is mitochondrial DNA (mtDNA), which is the DNA located in mitochondria of cells. mtDNA is inherited only from the mother, and hence enables researchers to trace a family's maternal lineage.

Inference Attacks on Kin Genomic Privacy

In this section, we discuss the main threats against kin genomic privacy.

DNA Dragnets

The privacy risks posed by genomic data's collection and use in forensics have been widely discussed in the context of DNA dragnets. DNA dragnets involve collecting tissue or saliva samples from people in a certain region to hunt criminals. The collected biological samples are then used to construct DNA databases. Although collecting such data from suspected criminals or from those who've given their informed consent is acceptable, there are still serious privacy implications.

A main concern about DNA dragnets is the conditions under which law enforcement is legally allowed to collect individuals' biological samples. Under the US Fourth Amendment, law enforcement must have a reasonable suspicion that a person is involved in a crime before requiring a search or seizure. However, the rules for DNA collection are still uncertain. For instance, in Melbourne, Florida, riding a bike at night without two functioning lights could lead to a DNA swab.³

Another concern is the duration such samples are kept in DNA databases and whether law enforcement can use the samples for other investigations. In 2015, Maryland's Supreme Court ruled that law enforcement could use DNA voluntarily provided to police investigating one crime to solve another.³

Also of concern is using research databases that collect biological samples in criminal investigations—without informing the donors about such use.⁴ Such forensic investigations have occurred in Australia, New Zealand, Norway, the UK, and Sweden for criminal identification, disaster victim identification, and paternity identification. A prominent example was the use of Sweden biobank blood samples to investigate the 2003 murder of a Swedish foreign minister.

One last serious privacy concern about DNA dragnets relates to kinship: law enforcement might use an individual's DNA from a DNA database to accuse a family member whose biological sample was never collected. Some US states already allow such familial searching of DNA databases. However, there are

concerns over whether the right to privacy is violated in the process.

DNA technology used by genealogists to identify unknown relatives and DNA dragnets used by law enforcement have been successfully combined to track down criminals. For example, police spent nearly 20 years (starting in the 1970s) chasing the BTK (“bind, torture, and kill”) serial killer.⁵ Use of DNA in forensics and familial DNA connections finally helped them identify the killer. Although the police already had the suspect’s DNA samples from the crime scenes and strong evidence that BTK was a man named Dennis Rader, they didn’t have the reasonable doubt necessary to get a DNA swab from Rader. Police learned that Rader’s daughter had recently been to the hospital for a pap smear. Thus, via a judge’s order (but without the daughter’s knowledge), the police received a sample of the daughter’s DNA from the hospital, determined the familial match between that sample and the crime scene DNA samples, and eventually caught Dennis Rader.

On one hand, familial search in forensics DNA databases is a powerful tool for the police. Experts state that this technique increases the number of suspects identified through DNA by 40 percent.³ On the other hand, privacy advocates question the legitimacy of obtaining information with this technique because it turns family members into genetic informants without their knowledge or consent.

Quantifying Kin Genomic Privacy

In previous work, we provided a quantification framework for assessing the effect on kin genomic privacy of family members revealing their genomes.⁶ To precisely quantify privacy, we mimicked an adversary who has access to some genome(s) in a given family and wants to infer the genomes of other family members. To do so, the adversary relies on the intergenome correlations (data between relatives); the observed genomic and phenotypic data; and, potentially, intragenome correlations (so-called linkage disequilibrium), typically if a genome is only partially observed. Our efficient inference algorithms were based on belief propagation and graphical models. Belief propagation let us reduce the complexity of computing marginal distributions of random variables from time exponential to linear in the number of considered variables.

Once the belief propagation algorithm output the posterior marginal probabilities given the observed genome(s) and phenotype(s), we quantified the change in genomic privacy with respect to the prior probability distribution given by general population statistics. To do so, we relied on the expected estimation error and success rate (which requires us to know the ground truth, or actual SNP value) and on entropy-based

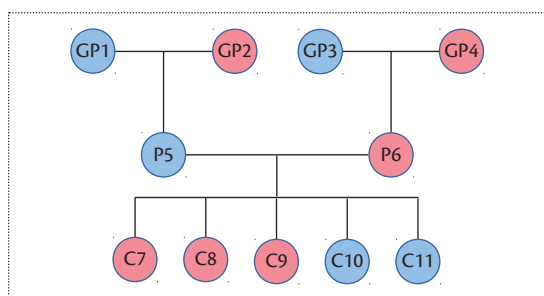


Figure 1. The CEPH/Utah Pedigree 1463 family tree consisting of 11 family members, which includes four grandparents (GP1 to GP4), two parents (P5 and P6), and five children (C7 to C11).⁸

metrics, which measure the adversary’s uncertainty and don’t require the ground truth.

We evaluated the proposed inference attacks and showed their efficiency and accuracy by using real genomic data from CEPH/Utah Pedigree 1463.⁷ Specifically, we selected 11 family members: the four grandparents (GP1 to GP4), the two parents (P5 and P6), and the five children (C7 to C11; see Figure 1). We focus here on the results of all common SNPs available on chromosome 1 (approximately 80,000). Table 1 shows the evolution of the expected estimation error and the success rate (the probability of inferring the correct SNP value) given the observation of zero to three different relatives. The three main rows represent the targeted individual (whose genomic data is hidden), and the columns represent the observed genomic data used to infer the hidden, targeted data. Looking at the P5 row, we see that we can decrease the average error by 50 percent by observing only P5’s two parents, and by even more if we also observe one of his children. Note that the proportion of SNPs inferred with success greater than 0.90 increases from 20 to 57 percent by observing P5’s parents. This proportion increases to 87 percent when seven of his relatives are observed (not shown in table). This clearly demonstrates that genomic privacy can be dramatically damaged by others’ sharing behavior.

Effect of High-Order Correlations in the Genome

To analyze the use of high-order correlations in the genome to improve existing work on inference attacks on genomic privacy, we also considered the phenotype–genotype relationships (such as physical traits or disease information).⁸ We used the complex correlations in the genome by applying Markov and recombination models between the *haplotypes*—nucleotides on a single chromosome that are so closely linked that they’re

Table 1. Absolute and relative levels of genomic privacy of the grandparent (GP1), parent (P5), and child (C7) whose genome is hidden (H), given the observation (\emptyset) of zero to three relatives.

H/O	Error*	\emptyset	P5	P5, GP2	C7, GP2	C7, C8, GP2
GP1	Absolute average error	0.446	0.322	0.309	0.404	0.385
	Relative average error (%)	100	72	69	91	86
	Single nucleotide polymorphisms (SNPs) with success rate >0.90 (%)	20	28	29	23	23
		\emptyset	GP1, GP2	C7, C8	C7, P6	GP1, GP2, C7
P5	Absolute average error	0.480	0.242	0.286	0.312	0.203
	Relative average error (%)	100	50	60	65	42
	SNPs with success rate >0.90 (%)	20	57	38	29	57
		\emptyset	P5	P5, C8	P5, P6	P5, P6, C8
C7	Absolute average error	0.489	0.344	0.301	0.182	0.182
	Relative average error (%)	100	70	62	37	37
	SNPs with success rate >0.90 (%)	20	28	40	64	64

*We use the absolute error to measure the genomic privacy of GP1, P5, and C7 for each individual, the error relative to the initial error (without observing any data) as a percentage, and the proportion of SNPs with a success rate over 0.90. The success rate is the probability of inferring the correct SNP value.

usually inherited as a unit. Then, similar to existing work,⁶ we proposed an efficient graph-based, iterative message-passing algorithm to consider all the aforementioned background information for the inference. Overall, our results show that an attacker’s inference power significantly improves by using complex correlations and phenotype information along with information about family bonds.

For evaluation, we focused on 100 neighboring SNPs on the CEPH/Utah Pedigree 1463’s DNA sequence on the 22nd chromosome. Using data from the 1000 Genomes Project (www.internationalgenome.org) and HapMap (www.ncbi.nlm.nih.gov/genome/probe/doc/ProjHapmap.shtml), we modeled the genome’s higher-order correlations (Markov and recombination models).

Among the 100 SNPs, we randomly hid 50 of the father’s SNPs and tried to infer them by gradually increasing the attacker’s background information. We also assumed that the attacker knew three of each family member’s phenotypes associated with the considered SNPs. We began revealing 50 random SNPs (out of 100) of other family members, starting from the most distant to the father in terms of number of family

members. To quantify genomic privacy, we used two metrics: estimation error and entropy.

Figure 2 shows our results for the attacker’s error (we achieved similar results for the entropy-based metric). The case of $k = 1$ (Markov chain with order 1 with no phenotype information) represents our previous work.⁶ Our results show that high-order correlations and phenotype information contributed significantly to the attacker’s inference power. For the Markov chain model, the attacker’s inference didn’t improve much for orders of Markov chain (k) greater than 3. The recombination model increased the attacker’s inference power more than the Markov chain model.

Suppose we’re working on a dataset consisting of a trio (father, mother, and child) and trying to infer a particular SNP of the father given the mother’s and child’s SNPs. Following Mendel’s law, if the child is homozygous (carrying two identical nucleotides) in that SNP position, we can easily infer the nucleotide in one strand of the father. However, if both the child and the mother are heterozygous (carrying two different nucleotides) in that SNP position, we can’t get any information about the nucleotide passed on from the father to the child.⁶

We can ameliorate this limitation by using haplotype information. Haplotypes are identical by descent (IBD) if they're identical and inherited from a common ancestor. There are several ways to detect IBD.^{9,10} Previously, we used Beagle¹¹ for this and showed IBD's contribution to the inference attack.¹² Beagle allows SNPs to be in linkage disequilibrium (LD) by modeling haplotype frequencies.

By employing this haplotype information, we introduced a new inference attack to find one of the parent's SNPs by using the genomes of the other parent and the children. We used the regions that are inherited together and worked from the idea that if the child's SNPs in a haplotype block aren't coming from the mother's genome, then they're coming from the father's. Then, we deduced that the child's other haplotype is inherited from the father. We evaluated our approach on CEPH/Utah Pedigree 1463 dataset, and showed that accurate inference about the father's SNPs could be accomplished using less data (that is, less genomic data from fewer family members) than previously.⁶

Membership Inference in Genomic Databases

In 2008, Nils Homer and his colleagues identified an attack against genomic privacy that determined a targeted individual's membership in a genomic database based on summary statistics about this database.¹³ By comparing a significant portion of the targeted individual's SNPs with the released statistics, the adversary could infer with high precision whether the individual was a member of the database.

The following year, Sriram Sankararaman and his colleagues proposed another statistical inference method, one based on likelihood ratio, to derive a theoretical bound on the attack's true-positive at a given false-positive rate.¹⁴ They showed that it's possible to detect relatives of the target whose SNPs are available to the adversary. Notably, they found that detecting a target's first-order relative (sibling, child, or parent) requires approximately four times as many SNPs as detecting the target with the same bound on false-positive and false-negative rates. Moreover, they empirically demonstrated that if the adversary has access to approximately 33,000 independent common SNPs, the true-positive rate decreases from 0.95 (for detecting the original individual) to 0.22 for detecting a first-order relative, and 0.03 for a second-order relative, at a false-positive rate of 10^{-3} .

More recently, Suyash Shringarpure and Carlos Bustamante developed an attack against genomic data-sharing beacons.¹⁵ Beacons are webservers that answer allele presence queries such as "Do you have a genome that has a specific nucleotide (A) at a specific genomic

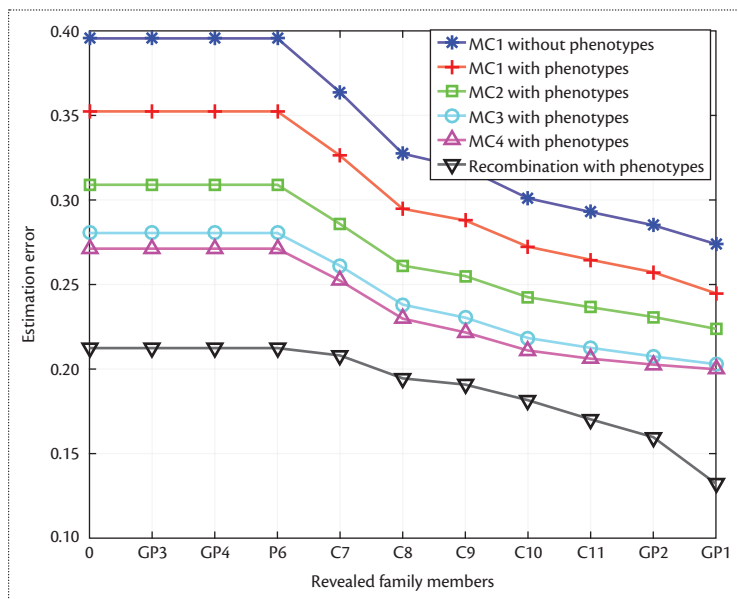


Figure 2. Decrease in father's genomic privacy by attacker's incorrectness. We revealed partial genomes of other family members for different high-order correlation models in the genome. MC is Markov chain model (with different orders).

position (position 11,272 on chromosome 1)?" with either "yes" or "no." By relying on a likelihood-ratio test, the authors showed that the responses to such queries could be used to reidentify individuals in a beacon.

Moreover, Shringarpure and Bustamante showed that relatives are also prone to such a reidentification attack. Similar to Sankararaman and his colleagues, the authors used a single parameter to model the degree of relatedness (the probability that two individuals share an allele at a single SNP: 1.00 for identical twins, 0.50 for parent-offspring and sibling pairs, 0.25 for first cousins, and so on) and derive the updated likelihood-ratio test as a function of this parameter. Using simulated data, they showed that in a beacon with 1,000 individuals, target reidentification was possible—at a more than 0.95 true-positive rate and 0.05 false-positive rate—with only 5,000 queries; first-order relative reidentification required approximately 40,000 queries. The true-positive rate dropped to 0.50 for second-order relatives, and approximately 0.23 for third-order relatives, with 40,000 queries.

Deanonimizing Publicly Available Genomic Datasets

As discussed, the Y chromosome is (almost) preserved along the male line of a given family. Thus, for communities in which last name is also preserved along the male line, the Y chromosome and last names are correlated. Such correlation can be accessed through public genealogy databases.

Melissa Gymrek and her colleagues recently showed that individuals' last names could be recovered by querying recreational genealogy databases with their Y-STRs.¹⁶ Furthermore, the combination of last name with other auxiliary information such as age and state (which can be easily obtained from public resources) could be used to triangulate the target's identity. Eventually, such triangulation would lead an attacker to link the anonymized genomic data stored on a public repository to the donor's real identity.

The authors used the public genealogy databases Ysearch (www.ysearch.org) and SMGF (www.smgf.org), both of which are free and have built-in search engines. When users input their (or someone else's) Y-STR profile, the database returns the last name of the corresponding donor. Gymrek and her colleagues also assumed that anonymized genomic data (from which they obtained the target's Y-STR profile) is available with the target's birth year and state of residency. Note that the US's Health Insurance Portability and Accountability Act of 1996 (HIPAA) doesn't protect these two pseud identifiers.

Finally, the authors determined the target's real identity by entering the target's last name, birth year, and state of residency into online public record search engines. They showed that this combination yielded a median result set (the set containing potential donors of a given anonymized genome) of 12. They also reported five successful surname inferences—in which the anonymized genome's donor could be uniquely identified—from Illumina datasets of three large families that were part of the 1000 Genomes Project, which eventually exposed nearly 50 research participants' identities.

Countermeasures

Here, we briefly discuss some potential countermeasures against these privacy risks.

Cryptography-Based Solutions

Keeping genomic data in encrypted form, instead of making it publicly available, and providing query results only to specific individuals (such as patients, medical centers, or researchers) might mitigate some of the aforementioned attacks. Cryptography-based techniques can protect both kin and personal genomic privacy. To this end, researchers have proposed cryptographic solutions for different query types.

There's been a significant amount of work on privacy-preserving pattern matching and the comparison of genomic sequences. Juan Ramon Troncoso-Pastoriza and his colleagues proposed an algorithm for private string searching on the DNA sequence by using a finite state machine.¹⁷ Their work was revisited by Marina Blanton and Mehrdad Aliasgari, who developed an

efficient method for sequence comparison using garbled circuits.¹⁸ Furthermore, Muhammad Naveed and his colleagues proposed a scheme based on functional encryption for privacy-preserving similarity tests on genomic data.¹⁹ Recently, Xiao Shaun Wang and his colleagues proposed an efficient privacy-preserving protocol to find genetically similar patients in a distributed environment.²⁰

Other works have focused on private clinical genomics. Emiliano De Cristofaro and his colleagues proposed a secure protocol between two parties that tests genomic sequences without leaking private information about the genomic sequence or the test's nature.²¹ Pierre Baldi and his colleagues used private-set intersection to present an effective algorithm for privacy-preserving clinical tests and direct-to-consumer methods on DNA sequences.²² Rui Wang and his colleagues proposed computing on genomic data by distributing the task between a data provider and consumer through program specialization.²³ Erman Ayday and his colleagues designed a scheme that protects the privacy of users' genomic data while enabling medical units to access the data to conduct medical tests or develop personalized medicine methods.²⁴ Finally, Zhicong Huang and his colleagues developed an information-theoretical technique to securely store genomic data.²⁵

One last line of investigation has explored the use of cryptography-based techniques such as homomorphic encryption, secure hardware, and secure multiparty computation.^{26,27}

Differential Privacy-Based Solutions

Cryptography-based techniques help individuals query genomic databases in a privacy-preserving way. However, such solutions don't prevent an attacker from making inferences from the results of such queries. As for cryptographic mechanisms, the techniques for mitigating membership inference were developed to protect personal genomic privacy in general. However, differential privacy, a well-known technique for answering statistical queries in a privacy-preserving manner,²⁸ can be easily adapted to preserve kin genomic privacy at a lower cost for utility because membership inference is more successful for individuals whose genomic data is known than for their kin.

To prevent such attacks, differential privacy has been used to compose privacy-preserving query mechanisms for genome-wide association study (GWAS) settings.^{29,30} Caroline Uhler and her colleagues proposed methods for releasing differentially private minor allele frequencies (MAFs), chi-square statistics, p -values, top- k most relevant SNPs to a specific phenotype, and specific correlations between particular SNP pairs.²⁹ These methods are notable because traditional

differential privacy techniques would be unsuitable: the number of correlations studied in GWAS is much larger than the number of people in the study. However, differential privacy is typically based on a mechanism that invokes Laplacian noise and, thus, requires a very large number of research participants to guarantee acceptable privacy and utility levels.

Aaron Johnson and Vitaly Shmatikov explained that computing the number of relevant SNPs and the pairs of correlated SNPs is the goal of a typical GWAS.³⁰ They provided a distance score mechanism to add noise to the output. All relevant queries required by a typical GWAS are supported, including the number of SNPs associated with a disease and the most significant SNPs' locations. Empirical analysis suggests that the new distance score-based, differentially private queries produced better, though still far from acceptable, utility for a typical GWAS. Differential privacy might also be a solution for the beacon attack, with a tradeoff in utility.

Optimization-Based Solutions

Differential privacy techniques perturb the data before releasing it, and cryptographic techniques are generally too inefficient for research settings. To avoid these issues, some individuals might decide to publicly share their data in clear (without encryption), for example, to help medical research progress. In a previous work, we proposed an optimization-based mechanism for reaching a suitable tradeoff between shared SNPs' usefulness and family members' genomic privacy.³¹ Optimization-based solutions could potentially mitigate all the attacks we've discussed. The optimization-based solution we discuss subsequently is particularly tailored to inference attacks.

Consider individuals who want to share their genome, yet are concerned about the subsequent privacy risks for themselves and their family. We designed a system that maximizes disclosure utility without exceeding a certain level of privacy loss within a family, considering kin genomic privacy, the family members' personal privacy preferences, the SNPs' privacy sensitivities, the correlations between SNPs, and the SNPs' research utility. Our solution automatically evaluates the privacy risks for all family members and decides which SNPs to disclose. It relies on the quantification framework discussed earlier and combinatorial optimization.

First, we defined a linear optimization problem that aims to maximize the utility of disclosed SNPs. Utility increases linearly with the number of shared SNPs, while satisfying all family members' genomic and health privacy constraints. This problem is very similar to the optimization literature's multidimensional knapsack problem; we relied on the branch-and-bound algorithm to find the optimal SNP subset to be disclosed. Second,

we applied a fine-tuning algorithm to account for the impact of intragenome correlations (linkage disequilibrium) on privacy. Our results indicated that, given the current data model, we can protect an entire family's genomic privacy while still making available an appropriate subset of genomic data. The approach's main disadvantage is that the considered optimization problem is nondeterministic polynomial time-complete and doesn't admit any fully polynomial-time approximation scheme. Therefore, we can't consider a significant number of SNPs using this problem.

Future Research Directions

Individuals are increasingly using direct-to-consumer services such as 23andMe, AncestryDNA, and FamilyTreeDNA to obtain their genomic information. Some share this information on public genome-sharing websites such as openSNP.org, mainly to contribute to genomic research. Although most share their genomic data on such platforms in an anonymized way, others either directly reveal their real identities or share sufficient information to cause deanonymization.^{16,32} By analyzing the genomic data of such websites' users, attackers might be able to infer family bonds; if at least one family member is identifiable or deanonymized, attackers might be able to reconstruct the actual family tree along with their genomic data.

Although this poses a serious privacy risk for contributors to anonymized genomic datasets, these datasets are crucial to genomic research. To find the balance between privacy and utility, an optimization-based solution, similar to the one we discussed, could be used. By selectively hiding dataset participants' SNPs, such an optimization-based technique would also hide familial relationships between the donated genomes and maximize the utility of the data shared by the donors.

Other types of biomedical data are becoming increasingly available, such as epigenomic or transcriptomic data. In particular, DNA methylation, one of the most important epigenomic elements, was recently shown to be reidentifiable through genotype inference,³³ because parts of the DNA methylation are influenced by the genome. These correlations between DNA methylation and the genome imply the existence of interdependent privacy risks for relatives' DNA methylation data. Therefore, it's crucial to precisely quantify these interdependent risks and analyze whether they appear beyond the parts of the DNA methylation that are correlated with the genome.

The kinship-related privacy implications of genomic data will only continue to grow as genomics gain importance and more people get their DNA sequenced.

Thus, it's crucial that we consider and implement appropriate protective mechanisms when using individuals' genomic data in various applications. ■

Acknowledgments

Erman Ayday was supported by funding from the European Union Horizon 2020 Research and Innovation Programme (Marie Sklodowska-Curie grant 707135) and the Scientific and Technological Research Council of Turkey, TUBITAK (grant 115C130).

References

1. M. Naveed et al., "Privacy in the Genomic Era," *ACM Computing Surveys*, vol. 48, no. 1, 2015; doi.org/10.1145/2767007.
2. Y. Erlich and A. Narayanan, "Routes for Breaching and Protecting Genetic Privacy," *Nature Rev.*, vol. 15, no. 6, 2014, pp. 409–421.
3. L. Kirchner, "DNA Dragnet: In Some Cities, Police Go from Stop-and-Frisk to Stop-and-Spit," *ProPublica*, 12 Sept. 2016; www.propublica.org/article/dna-dragnet-in-some-cities-police-go-from-stop-and-frisk-to-stop-and-spit.
4. V. Dranseika, J. Piasecki, and M. Waligora, "Forensic Uses of Research Biobanks: Should Donors Be Informed?," *Medicine, Health Care, and Philosophy*, vol. 19, no. 1, 2016, pp. 141–146.
5. E. Nakashima, "From DNA of Family, a Tool to Make Arrests," *Washington Post*, 21 Apr. 2008; www.washingtonpost.com/wp-dyn/content/article/2008/04/20/AR2008042002388.html.
6. M. Humbert et al., "Quantifying Interdependent Risks in Genomic Privacy," *ACM Trans. Privacy and Security*, vol. 20, no. 1, 2017, pp. 1–31.
7. R. Drmanac et al., "Human Genome Sequencing Using Unchained Base Reads on Self-Assembling DNA Nanoarrays," *Science*, vol. 327, no. 5961, 2010, pp. 78–81.
8. I. Daznabi et al., "An Inference Attack on Genomic Data Using Kinship, Complex Correlations, and Phenotype Information," to be published in *IEEE/ACM Trans. Computational Biology and Bioinformatics*, 2017.
9. B.L. Browning and S.R. Browning, "A Unified Approach to Genotype Imputation and Haplotype-Phase Inference for Large Data Sets of Trios and Unrelated Individuals," *Am. J. Human Genetics*, vol. 84, no. 2, 2009, pp. 210–223.
10. J.M. Rodriguez, S. Batzoglou, and S. Bercovici, "An Accurate Method for Inferring Relatedness in Large Datasets of Unphased Genotypes via an Embedded Likelihood-Ratio Test," *Proc. 17th Int'l Conf. Research in Computational Molecular Biology (RECOMB 13)*, 2013, pp. 212–229.
11. B.L. Browning and S.R. Browning, "A Fast, Powerful Method for Detecting Identity by Descent," *Am. J. Human Genetics*, vol. 88, no. 2, 2011, pp. 173–182.
12. F. Balci et al., "A New Inference Attack against Kin Genomic Privacy," *Proc. Privacy-Aware Computational Genomics (PRIVAGEN 15)*, 2015; www.cs.bilkent.edu.tr/~erman/pubs/PrivaGen_inference.pdf.
13. N. Homer et al., "Resolving Individuals Contributing Trace Amounts of DNA to Highly Complex Mixtures Using High-Density SNP Genotyping Microarrays," *PLoS Genetics*, vol. 4, no. 8, 2008; doi.org/10.1371/journal.pgen.1000167.
14. S. Sankaraman et al., "Genomic Privacy and Limits of Individual Detection in a Pool," *Nature Genetics*, vol. 41, no. 9, 2009, pp. 965–967.
15. S.S. Shringarpure and C.D. Bustamante, "Privacy Risks from Genomic Data-Sharing Beacons," *Am. J. Human Genetics*, vol. 97, no. 5, 2015, pp. 631–646.
16. M. Gymrek et al., "Identifying Personal Genomes by Surname Inference," *Science*, vol. 339, no. 6117, 2013; doi.org/10.1126/science.1229566.
17. J.R. Troncoso-Pastoriza, S. Katzenbeisser, and M. Celik, "Privacy Preserving Error Resilient DNA Searching through Oblivious Automata," *Proc. 14th ACM Conf. Computer and Communications Security (CCS 07)*, 2007, pp. 519–528.
18. M. Blanton and M. Aliasgari, "Secure Outsourcing of DNA Searching via Finite Automata," *Proc. 24th Ann. IFIP WG 11.3 Working Conf. Data and Applications Security and Privacy (DBSec 10)*, 2010, pp. 49–64.
19. M. Naveed et al., "Controlled Functional Encryption," *Proc. ACM SIGSAC Conf. Computer and Communications Security (CCS 14)*, 2014, pp. 1280–1291.
20. X.S. Wang et al., "Efficient Genome-Wide, Privacy-Preserving Similar Patient Query Based on Private Edit Distance," *Proc. ACM SIGSAC Conf. Computer and Communications Security (CCS 15)*, 2015, pp. 492–503.
21. E. De Cristofaro et al., "Secure Genomic Testing with Size- and Position-Hiding Private Substring Matching," *Proc. 12th ACM Workshop Privacy in the Electronic Society (WPES 13)*, 2013, pp. 107–118.
22. P. Baldi et al., "Countering GATTACA: Efficient and Secure Testing of Fully-Sequenced Human Genomes," *Proc. ACM SIGSAC Conf. Computer and Communications Security (CCS 11)*, 2011, pp. 691–702.
23. R. Wang et al., "Privacy-Preserving Genomic Computation through Program Specialization," *Proc. ACM Conf. Computer and Communications Security (CCS 09)*, 2009, pp. 338–347.
24. E. Ayday et al., "Protecting and Evaluating Genomic Privacy in Medical Tests and Personalized Medicine," *Proc. 12th ACM Workshop Privacy in the Electronic Society (WPES 13)*, 2013, pp. 95–106.
25. Z. Huang et al., "Genoguard: Protecting Genomic Data against Brute-Force Attacks," *Proc. IEEE Symp. Security and Privacy (SP 15)*, 2015; doi.org/10.1109/SP.2015.34.

26. M. Kantarcioglu et al., "A Cryptographic Approach to Securely Share and Query Genomic Sequences," *IEEE Trans. Information Technology in Biomedicine*, vol. 12, no. 5, 2008, pp. 606–617.
27. M. Canim, M. Kantarcioglu, and B. Malin, "Secure Management of Biomedical Data with Cryptographic Hardware," *IEEE Trans. Information Technology in Biomedicine*, vol. 16, no. 1, 2012, pp. 166–175.
28. C. Dwork, "Differential Privacy," *Proc. 33rd Int'l Conf. Automata, Languages and Programming (ICALP 06)*, 2006, pp. 1–12.
29. C. Uhler, A. Slavkovic, and S.E. Fienberg, "Privacy-Preserving Data Sharing for Genome-Wide Association Studies," *J. Privacy and Confidentiality*, vol. 5, no. 1, 2013, pp. 137–166.
30. A. Johnson and V. Shmatikov, "Privacy-Preserving Data Exploration in Genome-Wide Association Studies," *Proc. 19th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD 13)*, 2013, pp. 1079–1087.
31. M. Humbert et al., "Reconciling Utility with Privacy in Genomics," *Proc. 13th Workshop Privacy in the Electronic Society (WEPS 14)*, 2014, pp. 11–20.
32. M. Humbert et al., "De-anonymizing Genomic Databases Using Phenotypic Traits," *Proc. 15th Privacy Enhancing Technologies Symp. (PETS 15)*, 2015, pp. 99–114.
33. M. Backes et al., "Identifying Personal DNA Methylation Profiles by Genotype Inference," *Proc. 38th IEEE Symp. Security and Privacy (SP 17)*, 2017; doi.org/10.1109/SP.2017.21.

Erman Ayday is an assistant professor of computer science at Bilkent University. His research interests include privacy-enhancing technologies (including big data and genomic privacy), data security, and trust and reputation management. Ayday received a PhD in electrical and computer engineering from Georgia Tech. He's a member of IEEE and ACM. Contact him at erman@cs.bilkent.edu.tr.

Mathias Humbert is a senior data scientist at the Swiss Data Science Center, ETH Zurich, and École Polytechnique Fédérale de Lausanne (EPFL). His research interests include genomic privacy, privacy in online social networks, and location privacy. Humbert received a PhD in interdependent privacy from EPFL. He's a member of IEEE and ACM. Contact him at mathias.humbert@epfl.ch.



Executive Committee (ExCom) Members: Jeffrey Voas, President; Dennis Hoffman, Sr. Past President, Christian Hansen, Jr. Past President; Pierre Dersin, VP Technical Activities; Pradeep Lall, VP Publications; Carole Graas, VP Meetings and Conferences; Joe Childs, VP Membership; Alfred Stevens, Secretary; Bob Loomis, Treasurer

Administrative Committee (AdCom) Members: Joseph A. Childs, Pierre Dersin, Lance Fiondella, Carole Graas, Samuel J. Keene, W. Eric Wong, Scott Abrams, Evelyn H. Hirt, Charles H. Recchia, Jason W. Rupe, Alfred M. Stevens, Jeffrey Voas, Marsha Abramo, Loretta Arellano, Lon Chase, Pradeep Lall, Zhaojun (Steven) Li, Shihpyng Shieh

<http://rs.ieee.org>

The IEEE Reliability Society (RS) is a technical society within the IEEE, which is the world's leading professional association for the advancement of technology. The RS is engaged in the engineering disciplines of hardware, software, and human factors. Its focus on the broad aspects of reliability allows the RS to be seen as the IEEE Specialty Engineering organization. The IEEE Reliability Society is concerned with attaining and sustaining these design attributes throughout the total **life cycle**. **The Reliability Society has the management, resources, and administrative and technical structures to develop and to provide technical information via publications, training, conferences, and technical library (IEEE Xplore) data to its members and the Specialty Engineering community. The IEEE Reliability Society has 28 chapters and members in 60 countries worldwide.**

The Reliability Society is the IEEE professional society for Reliability Engineering, along with other Specialty Engineering disciplines. These disciplines are design engineering fields that apply scientific knowledge so that their specific attributes are designed into the system / product / device / process to assure that it will perform its intended function for the required duration within a given environment, including the ability to test and support it throughout its total life cycle. This is accomplished concurrently with other design disciplines by contributing to the planning and selection of the system architecture, design implementation, materials, processes, and components; followed by verifying the selections made by thorough analysis and test and then sustainment.

Visit the IEEE Reliability Society website as it is the gateway to the many resources that the RS makes available to its members and others interested in the broad aspects of Reliability and Specialty Engineering.

myCS

Read your subscriptions through the myCS publications portal at <http://mycs.computer.org>





Genomic Security (Lest We Forget)

Tatiana Bradley | University of California, Irvine
Xuhua Ding | Singapore Management University
Gene Tsudik | University of California, Irvine

Genomic privacy has attracted much attention from the research community, because its risks are unique and breaches can lead to terrifying leakage of sensitive information. The less-explored topic of genomic security must address threats of digitized genomes being altered, which can have dire consequences in medical or legal settings.

As full genome sequencing becomes increasingly practical and affordable, it's not hard to imagine a (near) future where large numbers of people store and maintain their digitized genomes. Ubiquitous access to one's digitized genome opens the door to a wide range of applications, ranging from serious (for instance, disease screening or paternity testing) to social (for instance, ancestry tracing or compatibility/dating). At the same time, a genome represents a veritable gold mine of extremely personal and sensitive information about its owner as well as that person's ancestors, descendants, and siblings. Furthermore, as the ultimate static biometric, a leaked genome can't be revoked or modified, thus exacerbating privacy concerns. Consequently, genomic privacy is a very timely and important subject, which has, in recent years, understandably attracted much attention from the research community. (See "Whole Genome Sequencing: Revolutionary Medicine or Privacy Nightmare?" for an overview of genomic privacy challenges.¹)

With the spotlight on the privacy front, where moderate progress has been made, comparatively less

attention has been devoted to genomic *security*. This is surprising because security is at least as important as privacy. In the context of personalized medicine, a modified genome can lead to wrong drugs or treatments being prescribed or administered. In terms of paternity or common ancestry testing, a modified genome can yield incorrect test results, which can translate into equally incorrect legal decisions.

Some recent work on genomic security (for instance, G.K. Ragesh and K. Baskaran's "Cryptographically Enforced Data Access Control in Personal Health Record Systems"²) focused on access control for health records, which—though important—doesn't prevent the possibility of an insider modifying genomic data. In particular, Ragesh and Baskaran sought to prevent, rather than detect, unauthorized modifications.²

One possible reason for genomic security *not* having received much attention thus far is that it's perceived not to pose any new challenges. In this article, we show that this conventional wisdom might be unjustified. After taking a closer look at genomic security, we identify some new challenges that can't be resolved by

naively applying current techniques. These challenges stem from several factors, including the size and longevity of the human genome, an unconventional application model, bandwidth and computation complexities, and the need to balance security with privacy.

Genomic Security

We envisage a generic application scenario with the following key features:

- An individual—Alice—obtains her digitized genome from an authorized sequencing lab (SL).
- Alice stores the result on her personal device, for instance, a laptop or smartphone.
- Later on, Alice wants (or is mandated) to conduct a genetic test, the purpose of which might be legal, medical, or social.
- The test requires Alice to provide some specific genomic data—typically, a small portion of the entire genomic sequence—to the application server (tester) that actually performs the test.

This scenario triggers various security issues for all stakeholders. One important issue is certification and periodic recertification of sequencing labs, because Alice clearly needs to trust the SL to correctly sequence and digitize her genome. This process would likely be done by a trusted government agency, for instance, the US Food and Drug Administration (FDA).

Another issue is certification of application-specific servers, which could be trickier due to a wide range of medical, legal, and social applications, each with its own access requirements to specific genomic excerpts. This would let Alice decide which parts of her genome should be revealed to a particular application. For example, a social app might be restricted to accessing segments that determine certain physical characteristics, such as height and hair or eye color, whereas a legal DNA profiling app that uses the short tandem repeat (STR) method might be restricted to accessing Combined DNA Index System (CODIS)-stipulated 13 core loci. This diversity calls for a well-defined policy or authorization syntax such that an application server can be certified to permit access to only a set of fixed and specific genomic locations or ranges thereof.

A related issue is proving rightful ownership; for example, if the test is conducted remotely (that is, over the Internet), how does Alice convince the tester that she supplied her own genomic data? This clearly requires a certification scheme that involves all three entities—the individual, the lab, and the tester.

Despite the obvious importance of all of the above, we focus in this article on more basic issues: the authenticity and integrity of Alice's genomic data in the context

of diverse applications. At first glance, this seems easily addressable via textbook security techniques, such as hash functions and digital signatures. However, as we discuss, the problem is a bit more challenging than it appears.

Genome Representation

In general, the human genome is a sequence of 3.2×10^9 base pairs—two letters chosen from the tiny four-letter alphabet: {adenine (A), cytosine (C), guanine (G), and thymine (T)}. The simplest way to represent it digitally is to use an array of three-bit blocks, each representing the first letter of a base pair at the corresponding absolute position. (An additional bit might be needed to account for sequencing errors, for example, a symbol “X” where a base letter was unreadable). The second letter doesn't need binary representation as it can be deduced from the first one using the base-pairing rule (www.biology-pages.info/B/BasePairing.html).

However, because human genomes have a high degree of similarity, an individual's genome is often represented as a set of differences with respect to a fixed reference genome. In practice, only approximately 3×10^6 base pairs are needed for most genetic applications. Hence, although a full and complete representation of a single genome might take up to 200 Gbytes, a compact version based on a reference representation, for instance, using the 1,000 Genomes Project variant call format (www.internationalgenome.org/wiki/Analysis/vcf4.0), occupies only about 120 Mbytes. For simplicity's sake, we assume the genome reference representation is a list of 3×10^6 tuples of the form: (x, L_x) , where L_x is the base pair at position x . In practice, L_x might contain more complex genomic data regarding position x . Nonetheless, the value of L_x doesn't impact the security issues discussed later.

Stakeholders and Trust Model

Again, the stakeholders in the aforementioned scenario include the individual—Alice, the SL, and the application server—tester. For now, we assume that the SL operates mostly offline, whereas Alice and the tester interact over the Internet or another similarly insecure communication channel. Given proper and timely certification by a higher authority (for instance, the FDA), we assume that everyone trusts the SL. However, the tester doesn't trust Alice regarding the authenticity and integrity of her genomic data. At the same time, Alice doesn't trust the tester with any of her genomic information beyond that which the latter is authorized to access for the particular test.

In the future, the SL's role might be replaced by a personal sequencing device. Such devices, though certainly not affordable today, are already available from vendors

such as Illumina. In the extreme, we can imagine a world in which individuals own and operate their own sequencing devices, perhaps as part of or as an attachment to a smartphone. Naturally, it would be crucial for such a device to be equivalent to an SL in terms of both functionality and trust. In particular, it would have to be certified by a trusted authority and would need to incorporate secure hardware coupled with some degree of tamper resistance as well as a means of secure logging and auditing.

Requirements

The first requirement is an efficient means for Alice to convince the tester of her genomic data's integrity and authenticity.

The second requirement is privacy of Alice's genome: because a typical genomic test uses only a small portion of the entire genome, the rest must be kept secret from the tester. Ideally, information revealed by Alice mustn't allow the tester to learn anything else about Alice's genome. However, this is unrealistic from the outset because a genome isn't random; information that corresponds to certain loci might allow the tester to infer (with absolute certainty, or at least with nonnegligible advantage over a random guess) contents of other loci. Although privacy is a key goal, the inference problem is beyond the scope of this article.

The third requirement is performance: minimal storage, communication, and computation overheads incurred by all stakeholders. This is of highest importance for Alice who might be using a resource-constrained personal device. Of course, following current trends, Alice could outsource storage and computation of her genomic data to a cloud service provider (CSP), which has vastly greater resources than her device. There's still an incentive to minimize all costs, due to the CSP's very large scale of both storage and computing. Outsourcing neither changes the trust model above nor invalidates the requirements. The other two stakeholders—the SL and the tester—are expected to be commercial entities with ample computing, storage, and communication facilities. (An exception would be peer-to-peer social genomic applications, in which a tester might be another personal device.) Nonetheless, it's always desirable to reduce their overheads.

Challenge

A prominent challenge stems from the conflict between security and privacy requirements. On one hand, Alice's privacy implies that she should control her genomic information revealed to the tester. On the other hand, the tester demands authenticity and integrity, which means that Alice must be unable to modify (or delete parts of) her digitized genome.

This issue is exacerbated by the compact reference representation. Consider a simple example. Suppose that the tester requests a sequence of X base letters, starting at position Y . We assume that Alice's genome has just one difference in this range: an A at position Y' (for $Y' - Y < X$). The next difference is a C at position $Y_{\text{next}} \geq Y + X$, while the previous difference is a G at position $Y_{\text{prev}} < Y$. An honest Alice would send the tester a single tuple: (Y', A) . She would also attain maximal privacy by revealing nothing beyond the minimum required by the tester.

Alternatively, a malicious Alice could cheat and send an empty string, thus claiming that her genome and the reference have no differences in the range $[Y, Y + X]$. If we assume that each difference is somehow individually authenticatable (for example, signed by the SL at sequencing time), Alice can't create base letter differences where none exist. However, she can easily omit actual differences from the requested range. In the database security literature, this is sometimes called the *range query completeness* problem, where a more generic term "records" is used instead of "differences." It also has a trivial solution: adjacent differences must be securely (cryptographically) bound, that is, authenticating a difference at position Y' must allow the tester to securely determine that previous and next differences occur at positions Y_{prev} and Y_{next} respectively.

This method is readily applicable in our context; for example, for each difference at position Y' involving a base letter $L_{Y'}$, SL could sign a tuple: $(Y', L_{Y'}, Y_{\text{prev}}, Y_{\text{next}})$ where Y_{prev} and Y_{next} are as defined before, with two special symbols (for instance, $-\text{inf}$ and $+\text{inf}$) indicating the start and end. For each difference in the requested range, Alice would send the tester one such signed tuple, and any cheating on her part would be trivially detectable. If Alice really had no differences in the entire $[Y, Y + X]$ range, there would necessarily exist either (or both) the closest previous or next closest difference, represented as a distinct signed tuple. It's easy to see that if Alice provides an SL-signed tuple corresponding to either position, the tester can verify it and thereby determine that Alice's genome has no differences in the $[Y, Y + X]$ range.

Although secure, this approach sacrifices some of Alice's genomic privacy. Note that, in the above example, the tester learns (potentially a lot) more than it's entitled to learn. Specifically, regardless of the number of differences in the $[Y, Y + X]$ range, the tester learns the positions of two other differences: Y_{prev} and Y_{next} . There seems to be no easy solution to this.

As this discussion illustrates, reconciling privacy and security isn't obvious, at least if reference representation is used. In the rest of this article, we discuss ways to

simultaneously attain integrity, authenticity, and completeness for the tester as well as privacy for Alice.

Naive Approaches

We start with some very naive approaches to authenticity and integrity. Though not quite practical, they provide insights into ensuing design challenges and lead us to a somewhat practical baseline technique.

No Privacy

In the no privacy (NoP) approach, after sequencing, the SL signs the compact genome representation and references its owner's identity (Alice) and/or the owner's public-key certificate. Thereafter, Alice can easily prove authenticity and integrity to the tester by transferring the whole signed genome and authenticating herself in the process. This incurs for Alice the lowest possible costs for storage (just the cleartext genome) and computation (almost none). In return, Alice has no privacy whatsoever, while communication overhead is maximal. The tester's costs are similar to Alice's, albeit storage is needed only temporarily, up to signature verification.

Finer-Grained Privacy

In the finer-grained privacy (FGP) approach, the SL partitions Alice's genome sequence into segments and separately signs each, using some unique identifier to tie all the segments together as well as to bind them to Alice. This way, Alice sends the tester the smallest set of signed segments that contain necessary/requested positions and base letters. One possibility is to pick uniform-size segments, which makes for easier processing and storage. Alternatively, genomic specialists can determine segments of variable lengths according to the application needs; for example, standard test types might call for specific fixed ranges. We don't pursue this further as it's orthogonal to our study. This approach offers weak privacy for Alice because it leaks extra (not strictly required) information to the tester. The actual amount of leakage depends on the segmentation algorithm and the specific tester application.

Baseline: Extreme FGP

Taking FGP to the extreme, we can obtain an optimal mix of security and privacy at the expense of storage. In this case, called extreme FGP (eFGP), the SL uses the full genome representation, instead of the compact (reference-based) version—that is, it individually signs every single base letter along with its position. As a result, Alice attains optimal privacy because only data corresponding to requested (and, presumably, duly authorized) positions is revealed. For its part, the tester can individually authenticate each position/base-letter pair and verify ownership.

eFGP's tradeoff is in performance: all parties incur much higher costs than NoP. SL has to compute 3.2×10^9 signatures. With RSA, the minimum near-term safe key/modulus size is 2,048 bits (anticipated to be secure until 2030), while elliptic curve cryptography (ECC) needs 224 bits for roughly the same security. (Both 2,048-bit RSA and 224-bit ECC are believed to offer 112 bits of security.) We can discount the SL's computation complexity because, as a commercial entity, it has ample resources and can always find a way to pass the extra costs onto its customers. Alice doesn't need to verify individual base-letter signatures; at delivery time, the SL can supersign the whole genome separately, and Alice can verify just that one signature.

Of more concern is storage, that is, space complexity: even if we ignore storage for position metadata, signatures themselves result in data expansion of two to three orders of magnitude, depending on the signature type. This translates into hundreds of gigabytes (ECC) or nearly a terabyte (RSA) per genome. For Alice, storing this much data on a personal device, and communicating it, is likely to be prohibitive in the near future. On the other hand, assuming that a typical test involves only 0.1 percent of the genome (which approximates the typical difference between any two humans), Alice's communication with the tester would be commensurately less intensive, that is, 1,000 times less.

For the tester, eFGP requires as many signature verifications as the number of base letters requested from Alice. This is where the choice of the signature scheme matters most. For instance, it's well known that, with small public exponents, RSA is generally 10 to 30 times faster than elliptic curve (EC) digital signature algorithm (DSA) for verification. The next question is whether the extra bandwidth consumed by RSA signatures is outweighed by faster verification. The answer depends on several variables, such as network speed and requested plaintext size.

Consider the following example. On a commodity 2015 MacBook Pro, OpenSSL reports signature verification speeds of 15,702/s and 1,540/s for RSA and ECC, respectively. We assume a 1-Gbps network and equally capable interfaces for Alice and the tester. Also, the tester can pipeline signature verification, that is, verify each base-letter signature immediately on receipt. We set $k = 3.2 \times 10^6$, which corresponds to 0.1 percent of the genome, and RSA and ECC sizes of 2,048 and 224 bits, respectively. Then, RSA transfer delay is estimated as $(2,048 \times 3.2 \times 10^6)/10^9 \approx 6.5$ s, and DSA as $(224 \times 3.2 \times 10^6)/10^9 \approx 0.7$ s. These delays are clearly dwarfed by signature verification times: $3.2 \times 10^6/15,702 \approx 203.8$ s for RSA and $3.2 \times 10^6/1,540 \approx 2,078$ s for ECC. If we pick a much smaller $k = 1,000$, signature verifications would be 0.064 s for RSA and

0.65 s for ECC, while transfer delays remain relatively insignificant: 0.002 s for RSA and 0.0002 s for ECC.

Consequently, at least for the time being, RSA has a clear performance advantage. It's easy to see that the gap would grow significantly larger with bigger key sizes, for example, 3,072 and 256 bits. Although other tester CPUs could yield very different results, it seems unlikely (though not impossible) that ECC would outperform RSA, unless congestion or other factors drastically reduce network speed. Today, very low network speeds can be encountered if Alice and the tester communicate over a 2G or 3G cellular network; however, gigabit cellular is already available and will probably become pervasive in a few years.

Note that although virtually all modern signature algorithms use the well-known hash-and-sign technique, our earlier discussion ignores the cost of hashing, because it's assumed to be negligible compared to that of signature verification. In addition, all signatures in eFGP are computed on distinct plaintexts, because each "message" includes a base letter, its position, and a reference to Alice's identity (and/or her public-key certificate).

An auxiliary issue is storage (disk) read speed on Alice's device. Although disk read speeds of modern smartphones don't yet match top network speeds, commodity laptops easily reach gigabits/second disk read speeds, for example, MacBook Pro in 2015. We can safely assume that smartphones will catch up in a few years. Note that storage write speed on the tester's side is less important because of presumably abundant resources.

In summary, eFGP offers a useful baseline: it achieves the best balance between security for the tester and privacy for Alice. Its main drawback is performance.

Performance Optimizations

Here we consider some means of improving the baseline eFGP's performance.

Batch Verification

One natural way to speed up the tester's computation is by using batch signature verification. This way, Alice still sends the same data to the tester, which accumulates all plaintext hashes and all signatures and verifies the entire collection at the cost of one signature verification. (In other words, an accumulated hash is verified against an accumulated signature.) The best-known example is the batch version of full-domain hash (FDH)-RSA,^{3,4} an RSA variant that requires an FDH—a cryptographic hash function that yields digests of the same bit size as the RSA modulus. However, batch FDH-RSA requires computing separate accumulators of message hashes

and signatures, which costs $2k$ modular multiplications, where k is the number of signatures.

Because plain RSA signatures can be used safely with a fixed small public exponent of 3, each signature verification (without batching) entails two modular multiplications, resulting in the same $2k$ total. Therefore, there appears to be no performance gain for the tester in using batch FDH-RSA. In fact, the latter might be more expensive because FDH can be slower than a plain hash function.

Though batch techniques aren't unique to RSA, most others either require different public-private exponents per message or are applicable to batching signatures by multiple signers.

Condensed and Aggregated Signatures

Another potential optimization is condensed signatures,⁵ which is very similar to batch verification, except that it's Alice who accumulates all signatures (by the same signer) into a single condensed signature and sends it, along with all plaintexts, to the tester. The latter accumulates all plaintext hashes and verifies one signature. Condensed signatures appear to be a perfect match for RSA because of its comparatively large signature size. Similar to batch, an FDH-RSA variant must be used here. Assuming a small public exponent, the tester computes only k (rather than $2k$ in batch RSA) modular multiplications, although Alice is now forced to compute the other k to produce the condensed signature.

There are also more general techniques, such as aggregated signatures, exemplified by the BGLS (Boneh, Gentry, Lynn, and Shacham) signature scheme.⁶ BGLS and its follow-ons allow k signatures produced by k signers over k distinct messages to be aggregated into one signature. By verifying this signature against all k messages, each message's authenticity and integrity are ascertained. (As mentioned earlier, all base-letter messages are unique.) Also, BGLS doesn't require signers to be distinct; in fact, it's more efficient when all aggregated signatures are by the same signer. Aggregation performed by Alice requires k EC multiplications. As with condensed RSA, bandwidth overhead is minimal. The tester's verification requires k EC multiplications and one signature verification (pairing).

On one hand, k modular or EC multiplications performed by Alice is a costly endeavor, because her personal device might be computationally weak. On the other hand, Alice can precompute a condensed or aggregated signature. Furthermore, bandwidth savings can be substantial, for example, close to 6 s for $k = 3.2 \times 10^6$ in our example above. It thus remains unclear whether there's a performance incentive as far as using condensed signatures, unless bandwidth complexity must be minimized or precomputation by Alice is free.

Merkle Hash Tree

A popular tool in computer security, a Merkle hash tree (MHT) is a data structure for efficient authentication of any member or subset of a large set. It's a (typically, binary) tree where leaves are hashes of individual set members, and each interior node is the hash of its two children. Assuming a suitable cryptographic hash function, the tree root is the collective indirect hash of all leaves. The root node's signature thus authenticates the entire tree. In an MHT with n leaves, given an $O(\log n)$ -size co-path, any set element (leaf) can be authenticated as being part of the tree by hashing upward toward the root and verifying the root signature. Constructing an MHT takes $O(2n)$ hashes and one signature. It's necessary to store only the leaves and the (signed) root because all interior nodes can be reconstructed with $O(n)$ hashes. One notable application for MHTs is efficient certificate revocation checking.

We can easily adopt the MHT construct to the problem at hand, as follows. The SL constructs Alice's MHT with the ordered sequence of base letters serving as the leaves, then signs the root. Alice reveals a genomic segment—a sequence of contiguous base letters—to the tester. To do so, Alice provides the segment and a co-path consisting of all sibling nodes on the path(s) from the root to the common ancestor(s) of the segment. The tester reconstructs the Merkle tree's root and verifies the SL's signature. Co-path length is bounded by the MHT height of approximately $32 \approx \log_2 3.2 \times 10^9$. Thus, Alice sends the tester up to 32 hashes (8 Kbits total at 256 bits/hash) and a root signature in addition to the requested base-letter segment.

For a k -long segment, this method involves negligible bandwidth overhead and only requires the tester to perform a single signature verification as well as $2k + 32 - \log k$ hashes.

One issue is Alice's storage: the entire tree takes more than 200 Gbytes with a 256-bit hash function. A well-known way to cut the storage cost by half is for Alice to reconstruct the tree at runtime. Then, Alice's storage would be the same as in NoP. However, the downside is the need to compute 3.2×10^9 hashes on demand, which is impractical.

Another issue is Alice's privacy: Alice reveals only what's absolutely necessary—that is, the requested segment base letters. Unfortunately, the co-path gives away additional information. Consider the example in Figure 1: leaves 2 through 6 correspond to the base-letter segment CGATA. The accompanying co-path would include nodes 1 and 12, but not base letters in positions 1, 7, and 8. However, knowledge of node 1 allows the tester to learn G, and node 12 can be used to learn T and G in positions 7 and 8, respectively. This is due to the low entropy of individual base letters;

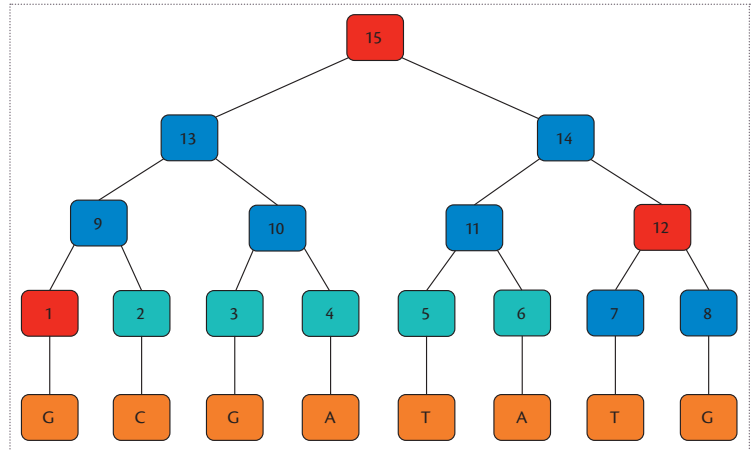


Figure 1. Merkle hash tree (MHT) leakage example. Knowledge of node 1 directly reveals leaf G, and knowledge of node 12 lets the tester know the leaves T and G for position 7 and 8, respectively.

there are only four possibilities for 1, and 16 possibilities for 7 and 8, which makes exhaustive searching easy. Of course, a co-path node's height exponentially influences the complexity of an exhaustive search. Given an interior node at height z , $4^{(2^z - 1)/2}$ trials are necessary, on average, to learn its descendant leaf base letters. Therefore, an exhaustive search is practical up to about $z = 5$, implying that up to 32 extra base letters might be learned by the tester.

Salted Merkle Tree

The natural next step is to prevent privacy leakage in MHT. This can be achieved using a salted Merkle hash tree (sMHT). For each base letter L_i at position i , the SL generates a pseudorandom salt s_i . The corresponding leaf LF_i is computed as $F_{s_i}(L_i, i)$, where $F()$ is a keyed pseudorandom function indexed on s_i , for instance, HMAC. An alternative is $LF_i = H(s_i, L_i, i)$ where $H()$ is a cryptographic hash function.

The rest of the tree is constructed as before. All salts are given to Alice by the SL as part of the initial digitized genome transfer. Salt bit size should be sufficient to rule out brute-force attacks, that is, at least 128. Then, Alice sends the tester all requested base letters along with their salts. This is in addition to the signed root and the co-path.

sMHT offers the same privacy for Alice, as well as the same integrity and authenticity guarantees for the tester, as eFGP. However, sending salts consumes additional bandwidth, comparable to eFGP without condensed or aggregated signatures. Unlike signatures, salts can't be compressed or accumulated. Salts also impose much higher storage overhead for Alice. There is a trivial way to avoid it if the SL generates all salts using a keyed pseudorandom function with a key K_a , for instance,

$s_i = F'_{K_a}(i)$, and shares K_a with Alice as part of the initial transfer. Then, Alice can easily recompute, on demand, all salts corresponding to the leaves in the revealed base-letter segment.

Another issue with both MHT and sMHT is the number of contiguous segments revealed to the tester. Our discussion above assumed only one such segment of variable size. It's quite possible that some genetic tests require many segments from disparate places in the genome. In that case, bandwidth and computational complexity of eFGP (with or without condensed/aggregated signatures) is unaffected, whereas tree-based techniques would require multiple partial (up to the height of the least common ancestor of all segment-formed subtrees) co-paths, one for each segment.

Redactable Signatures

An alternative approach for balancing authenticity and integrity for the tester with privacy for Alice is to replace standard signatures (for instance, RSA or EC-DSA) with specialized methods. One attractive concept is *redactable signatures* (RS), introduced more-or-less concurrently by both Robert Johnson and his colleagues and Ron Steinfeld and his colleagues.^{7,8} An RS scheme allows authorized “cryptographic redactions” of a signed message. In other words, given a redactable signature with a signed message, an authorized party can redact the message and obtain a new valid signature, without knowledge of the signer’s signing key. In the context of RS, we can view a base-letter segment as a “redaction” of the full genome, in which all other data is crossed out. SL computes a redactable signature over Alice’s genome. This signature is then redacted to suit the specific segment to be sent to the tester.

Because RS is a very general concept, both eFGP and sMHT can be viewed as redactable signature schemes; indeed, very similar approaches are described by Johnson and his colleagues.⁷ (We note that they also suggest salted MHTs.⁷ However, because they’re computed in a special way, salts for revealed base letters don’t need to be transmitted, as they can be recomputed by the tester.) Several RS variations have been proposed, for example, hiding sizes of redacted areas⁹ and RS over nonstring data.¹⁰ However, these features appear irrelevant to the context of genomic security.

Signature Aggregation and Chaining

The final approach we discuss is digital signature aggregation and chaining (DSAC).^{11,12} It’s a very simple technique, similar to the one sketched out earlier for secure range queries. It provides authenticity, integrity, and completeness. The basic idea is to construct signatures over a sequence of elements such that it becomes

easy to demonstrate authenticity, integrity, and completeness of a reply to any range query. Given a genomic sequence $\{L_1, \dots, L_N\}$, the SL computes a signature chain in two steps, for $0 < i \leq N$:

- $R_0 = s_0, R_i = [L_p, i, s_p, H(R_{i-1}, s_{i-1})]$ and
- $\sigma_i = F_{sig}(R_i)$,

where F_{sig} is any suitable hash-and-sign signature function, $\{s_0, \dots, s_N\}$ are $N + 1$ pseudorandom salts (same as in sMHT), and $H()$ is a hash function. Without getting into further details, it’s easy to see that to authenticate and verify integrity and completeness of a reply to a range query $[i, j]$, it suffices to produce $H(R_{i-1}, s_{i-1})$ as well as $\{L_p, \dots, L_j\}, \{s_p, \dots, s_j\}$ and σ_j .

From the bandwidth perspective, this is a particularly appealing technique due to its minimal overhead. However, DSAC’s most attractive aspect is the verification cost: $(j - i)$ hashes with salts and one signature validation of σ_j . The downside of DSAC is its storage cost, which is as large as eFGP.

Limitations of Current Techniques

We gave an overview of several fairly simple approaches to genomic security. All offer roughly equivalent security (authenticity and integrity) for the tester. As far as Alice’s privacy, eFGP, sMHT, and DSAC offer the best privacy by revealing only the required information. As far as performance, eFGP with condensed/aggregated signatures has the lowest possible bandwidth overhead, although computation overhead amounts to $O(k)$ multiplications for Alice and the tester. sMHT has very low computation overhead dominated by $O(2k + 32)$ hashes and a signature verification, while its bandwidth overhead is slightly higher, unless many disparate (non-contiguous) segments are involved. Finally, DSAC also offers very low bandwidth overhead coupled with the only k hashes and one signature verification.

To compare performance, Table 1 estimates several overhead factors, including the number of signatures the SL computes, the number of signatures the tester verifies, the number of bits Alice stores and transmits, and the number of cryptographic operations Alice performs.

As Table 1 shows, although all schemes except NoP offer optimal security and privacy, none incurs overheads close to the lower bounds. For example, in the case of sMHT, Alice stores approximately 214,000 times and transfers approximately 28 times more data; this is in addition to the 32-fold computation cost.

Improving Efficiency

Further work is needed to reduce computation overhead. One obvious step is to avoid the full genome

Table 1. Performance comparison of a realistic sample set of variable values: $N = 3.2 \times 10^9$; $N_r = 3.2 \times 10^6$; $k = 1,000$; $s_\sigma = 2,048$; $s_h = 256$; and $s_s = 128$.*

Approach	Sequencing lab (no. of signatures computed)	Tester (no. of signatures verified)	Alice's workload		
			Storage (bits)	Communication (bits)	Computation (no. of hash operations)
No privacy	1	1	$3N_r + s_\sigma$	$3N_r + s_\sigma$	–
Extreme finer-grained privacy (eFGP)	N	k	$3N + s_\sigma N$	$3k + s_\sigma k$	–
eFGP + aggregation	N	1	$3N + s_\sigma N$	$3k + s_\sigma$	$O(k)$
Merkle hash tree (MHT)	1	1	$3N + 2s_h N + s_\sigma$	$3k + s_h \log N + s_\sigma$	$O(\log N)$
Salted MHT	1	1	$3N + 2s_h N + s_s N + s_\sigma$	$3k + s_s k + s_h \log N + s_\sigma$	$O(\log N)$
Digital signature aggregation and chaining	N	1	$3N + s_\sigma N + s_s N$	$3k + s_s k + s_\sigma$	–
Lower bound	1	1	$3N_r + s_\sigma$	$3k + s_\sigma$	–

* N is number of base pairs in full genome representation; N_r is number of base pairs in reference representation; k is number of base pairs requested by the tester; s_σ is signature bit size; s_s is bit size of salt; and s_h is bit size of hash function digest.

representation, which takes a heavy storage toll. Ideally, the SL would sign a reference representation of Alice's genome and grant Alice the ability to redact arbitrary portions of this representation, which are outside the range requested by the tester, as well as efficiently prove that nonredacted portions (all properly signed by the SL) are complete—that is, Alice hasn't omitted anything from the requested range.

We sketch out one possible approach that satisfies these requirements and offers an optimal tradeoff among security, privacy, and efficiency. The main idea is for the SL to sign all pairs of adjacent mutations, similar to the trivial method we described earlier. However, actual positions and contents of mutations aren't revealed; instead, the SL signs cryptographic commitments to both contents and positions of adjacent mutations. Each signed tuple contains two commitments. A reference representation with k mutations would need $k + 1$ signed tuples. Note that two dummy sentinel mutations are needed to demarcate the beginning and end of the genome. When the tester requests all mutations in a specific range, Alice supplies one or more tuples. If the positions of both mutations in a tuple are within range, Alice decommits their locations and contents. (The tester can easily verify correctness.) If the lower-indexed mutation is within range and the higher one isn't, Alice decommits only the former. She then proves (in zero knowledge) that the other mutation's

committed value (position) is greater than the upper range limit. A similar process is followed if a signed tuple's higher-indexed mutation is in the range while the lower one isn't. In the case in which the requested range contains no mutations, Alice releases a single signed tuple, wherein the lower-indexed mutation is below the lower range limit, and the higher-indexed mutation is above the upper range limit. She then provides two zero-knowledge proofs, each showing that committed positions are outside the requested range. Proving that a committed (and secret) integer is within a specific range is both possible and quite efficient, using techniques such as those offered by Fabrice Boudot.¹³

Due to length restrictions for the present article, we don't elaborate on this approach.

Anonymity

In the context of some genetic (for instance, parentage) tests, Alice might want to hide her identity from the tester. For *pseudonymity*, it suffices for the SL to tie Alice's genome to a random pseudonym or a pseudonymous public-key certificate. Alice can then communicate with the tester over some anonymous channel, such as Tor. Stronger privacy (that is, anonymity) requires that any two genetic tests must be unlinkable. Clearly, none of the methods described above is unlinkable. However, there is some hope for redactable signatures, which can be made unlinkable, as shown in "Composable

and Modular Anonymous Credentials: Definitions and Practical Constructions.”¹⁴

We argue that genomic security has been underappreciated in favor of privacy. We believe security is vital to adoption of emerging and future personal genomic applications. The interesting mix of integrity, authenticity, and privacy requirements for multiple parties translates into a research challenge. We explored several fairly intuitive approaches, none of which satisfies all ideal security and performance requirements. Clearly, much remains to be done. ■

References


1. E. Ayday et al., “Whole Genome Sequencing: Revolutionary Medicine or Privacy Nightmare?” *IEEE Computer*, vol. 48, no. 2, 2015, pp. 58–66.
2. G.K. Ragesh and K. Baskaran, “Cryptographically Enforced Data Access Control in Personal Health Record Systems,” *Procedia Technology*, vol. 25, 2016, pp. 473–480.
3. M. Bellare, J.A. Garay, and T. Rabin, “Fast Batch Verification for Modular Exponentiation and Digital Signatures,” *Proc. Int’l Conf. Theory and Applications of Cryptographic Techniques* (EUROCRYPT 98), LNCS 1403, Springer, 1998, pp. 236–250.
4. M. Bellare and P. Rogaway, “The Exact Security of Digital Signatures—How to Sign with RSA and Rabin,” *Proc. Int’l Conf. Theory and Applications of Cryptographic Techniques* (EUROCRYPT 96), LNCS 1070, Springer, 1996, pp. 399–416.
5. E. Mykletun, M. Narasimha, and G. Tsudik, “Signature Bouquets: Immutability for Aggregated/Condensed Signatures,” *Proc. European Symp. Research in Computer Security* (ESORICS 04), LNCS 3193, Springer, 2004, pp. 160–176.
6. D. Boneh et al., “Aggregate and Verifiably Encrypted Signatures from Bilinear Maps,” *Proc. Int’l Conf. Theory and Applications of Cryptographic Techniques* (EUROCRYPT 03), LNCS 2656, Springer, 2003, pp. 416–432.
7. R. Johnson et al., “Homomorphic Signature Schemes,” *Topics in Cryptology*, LNCS 2271, Springer, 2002, pp. 244–262.
8. R. Steinfeld, L. Bull, and Y. Zheng, “Content Extraction Signatures,” *Proc. Int’l Conf. Information Security and Cryptology* (ICISC 01), LNCS 2288, Springer, 2001, pp. 285–304.
9. E.-C. Chang, C.L. Lim, and J. Xu, “Short Redactable Signatures Using Random Trees,” *Topics in Cryptology*, LNCS 5473, Springer, 2009, pp. 133–147.
10. C. Brzuska et al., “Redactable Signatures for Tree-Structured Data: Definitions and Constructions,” *Proc. Int’l Conf. Applied Cryptography and Network*

- Security* (ACNS 10), LNCS 6123, Springer, 2010, pp. 87–104.
11. M. Narasimha and G. Tsudik, “DSAC: Integrity for Outsourced Databases with Signature Aggregation and Chaining,” *Proc. ACM Int’l Conf. Information and Knowledge Management* (CIKM 05), 2005, pp. 235–236.
12. M. Narasimha and G. Tsudik, “Authentication of Outsourced Databases Using Signature Aggregation and Chaining,” *Proc. Int’l Conf. Database Systems for Advanced Applications* (DASFAA 06), 2006, pp. 420–436.
13. F. Boudot, “Efficient Proofs That a Committed Number Lies in an Interval,” *Advances in Cryptology* (EUROCRYPT 00), Springer, 2000, pp. 431–444.
14. J. Camenisch et al., “Composable and Modular Anonymous Credentials: Definitions and Practical Constructions,” *Advances in Cryptology* (ASIACRYPT 15), LNCS 9453, Springer, 2015, pp. 262–288.

Tatiana Bradley is a PhD student in computer science at the University of California, Irvine (UCI). Her research focuses on applied cryptography, including privacy-preserving computation. Contact her at tebradle@uci.edu.

Xuhua Ding is an associate professor at the School of Information Systems of the Singapore Management University. His research interests include trustworthy computing, system security, applied cryptography, and multimedia security. Ding received a PhD in computer science from the University of Southern California (USC). Contact him at xhding@smu.edu.sg.

Gene Tsudik is a Chancellor’s Professor of Computer Science and director of the Secure Computing and Networking Center at UCI. His research interests include topics in security and applied cryptography. Tsudik received a PhD in computer science from USC. He’s a Fellow of ACM, IEEE, and AAAS and a foreign member of Academia Europaea. Contact him at gts@ics.uci.edu.

 Read your subscriptions through the myCS publications portal at <http://mycs.computer.org>

Cybersecurity Framework Adoption:

Using Capability Levels for Implementation Tiers and Profiles

Adenekan Dedeké | Northeastern University

With business data breaches on the rise, NIST introduced the 2014 Cybersecurity Framework (CSF) to help companies reduce the cybersecurity risks threatening their critical infrastructures. CSF's key elements are described, with recommendations for organizations at various levels of adoption.

The likelihood of business data breaches is rising, with 60 percent of all organizations suffering more than one security incident in 2015.¹ To address this threat, 48 percent of organizations increased investments in security technologies, and 73 percent developed a data breach response plan. Furthermore, on 12 February 2013, former US President Barack Obama issued Executive Order 13636 to improve the cybersecurity of industries with critical infrastructures.²

After working collaboratively with stakeholders from many US economy sectors, on 12 February 2014, NIST issued its Cybersecurity Framework (CSF; www.nist.gov/cyberframework) in an attempt to answer the question, What factors and areas must an organization consider if it wants to effectively reduce the cybersecurity risks that threaten its critical infrastructures?

Gartner estimates that by 2020, more than 50 percent of organizations will be using the NIST framework, up from the estimated 30 percent that adopted it in 2015.³ One of CSF's advantages is that it encourages a shift in cybersecurity management from a compliance to a risk management orientation.⁴ However, as I discuss in this article, two adoption roadblocks must be removed. First, the differences between these compliance- and risk-oriented approaches must be clarified. And second, additional methods to guide the initial

implementation and progressive expansion of CSF will be needed.

What Is Compliance-Oriented Cybersecurity?

The beginnings of the compliance-oriented approach can be traced to the enactment of rules requiring organizations to deploy minimum safeguards. For example, government regulations, such as the Health Insurance Portability and Accountability Act (HIPAA), and industry specifications, such as the Payment Card Industry Digital Security Standard (PCI-DSS), describe security controls that had to be implemented in specific industries. Once such laws are enacted and their standards accepted, firms ignoring the requirements face a noncompliance risk. Specifically, if an organization doesn't implement the safeguards described in a standard or regulation, it exposes itself to economic and legal risks. Regrettably, rather than being viewed as security improvement ventures, such laws are sometimes perceived as "the cost of doing business." Organizations might hire a vendor to help implement safeguards and achieve compliance with the least possible cost and effort. Thus, this approach is sometimes called the "check-the-box" approach to security. Typically, the compliance-oriented approach is effective for scenarios

in which a specific problem, one with known causes or threats, must be resolved with a new remedy or solution.

Unfortunately, the cybersecurity threats and challenges that organizations face are too diverse to be compiled into a single list. For example, healthcare industry organizations are so varied that the HIPAA and HIPAA–HITECH (Health Information Technology for Economic and Clinical Health) rules offer multiple guidelines, rather than prescriptions, for the implementation of administrative, technical, and physical safeguards.⁵ Hence, different organizations can deploy different controls and practices to satisfy the HIPAA compliance requirement of having “reasonable and appropriate safeguards.”⁵ This creates a reality in which the meaning of HIPAA compliance might vary—either slightly or widely—from one healthcare organization to another.

One way to circumvent the weaknesses of overly broad safeguards is to create laws that target particular processes and technologies. This is what the PCI-DSS attempts to do by focusing on making payment processing and its systems more secure. Although this narrow focus makes the standard more prescriptive, a study of PCI-DSS adopters showed that organizations found it difficult to maintain PCI-DSS compliance from one year to the next.⁶ Less than one-third (28.6 percent) of certified companies were in full compliance less than a year after their first successful compliance certification. The study also found that many PCI-DSS implementations manifested a narrow overreliance on prevention and a lack of attention to attack detection, damage mitigation, and residual risk identification. In other words, the compliance-oriented approach didn’t offer adequate cybersecurity protection. The study recommended that PCI-DSS certification be seen as the baseline or industrywide minimum acceptable cybersecurity standard.⁶

What Is Risk-Oriented Cybersecurity?

The risk-based approach frames cybersecurity threats and vulnerabilities as risks rather than as events. *Risk* is defined as the estimation of the likelihood that a specific threat source will exploit a particular vulnerability and produce a negative impact on (or harm) an organization.⁷ The severity of a specific impact is determined by the degree of harm that it could inflict on an organization’s mission if or when specific IT assets are rendered inoperative. The magnitude of the potentially harmful impact could also be used to assign a “business value” score or a “criticality” score to IT assets and resources.⁷ Information security risks are defined as the potentially adverse impacts on an organization’s operations—including its mission, services, image, and reputation—that could arise from the loss of confidentiality, integrity, or availability of information and information systems.⁸

These two approaches to cybersecurity differ in the scope of deployed controls and the purposes pursued. For example, organizations that adopt a compliance-oriented approach focus on certifying their internal processes’ security using a certification/compliance framework, whereas firms that adopt a risk-oriented approach focus on ensuring that both their internal and external processes are secure enough to withstand emerging external threats. Furthermore, compliance-oriented adopters are likely to deploy the minimum number of controls required to achieve formal compliance or certification, while risk-oriented adopters are more likely to deploy baseline controls as well as controls that are required to help the firm reduce emerging security risks.

Exploring the NIST Cybersecurity Framework’s Core

The NIST CSF consists of three elements: the core, the profile, and the implementation tiers. The core consists of five functions. Each function has several categories, with each category divided into specific technical and management activities (outcomes). The five functions are defined as follows:⁹

- The *identify* function defines the actions related to the understanding of policies, governance structures, asset categorization, cybersecurity risks, and priorities relevant for managing cybersecurity risks to systems, assets, data, and capabilities. Categories include asset management, business environment, governance, risk assessment, and risk management strategy.
- The *protect* function covers activities related to the development and implementation of safeguards to protect critical infrastructure services and to train staff and employees. Categories include access control, awareness and training, data security, information protection processes and procedures, maintenance, and protective technology.
- The *detect* function involves activities related to the development and deployment of appropriate searching, monitoring, and detection activities to identify cybersecurity events. Categories include anomalies and events, security continuous monitoring, and detection processes.
- The *respond* function includes activities related to the development and implementation of appropriate plans and processes to take action regarding a detected cybersecurity event. Categories include response planning, communications, analysis, mitigation, and improvements.
- The *recover* function describes activities related to the development and implementation of appropriate plans and processes to recover from cybersecurity events and to restore services and capabilities

impacted by such events. Categories include recovery planning, improvements, and communications.

These CSF functions reflect how NIST defines risk-based management. The framework recognizes four components of risk management: framing, assessment, response, and monitoring.⁸ The identify function addresses risk framing and assessment. The protect function focuses on managing risk through mitigation, training, and risk prevention mechanisms. The detect and response functions combine to enable monitoring and proactive responses, respectively. The recover function focuses on postattack actions to recover from cybersecurity incidents and restore an organization's capabilities and services.

Hence, one of CSF's central advantages is that it covers a broad range of areas and outcomes. However, some researchers argue that it falls short in a few areas. For example, some contend that CSF excludes data privacy outcomes and statutory and regulatory issues, and is too complicated for management and board members to understand.⁴ Nevertheless, I argue that companies could resolve some of these criticisms on their own. Organizations could add new functions and outcomes, for example, data privacy outcomes, to the CSF according to their goals and priorities. Similarly, they could ignore functions and outcomes irrelevant to their contexts.

However, one area that CSF doesn't cover strongly is risk transfer. I think that risk transfer is a central dimension of risk management that should be included in the CSF.

Cyberrisk Transfer: Advantages and Limitations

Cybersecurity insurance is a product that's designed to mitigate losses caused by cyberincidents such as data breaches, network damage, and cyberextortion. Hence, organizations could use this service as a mechanism to transfer portions of their risks to an insurance company. Cybersecurity policy coverage falls into two categories: *first-party loss insurance*, which covers direct losses to a company arising from events such as business interruption and destruction of data and property, and *third-party loss insurance*, which covers losses that a company causes to its customers and others.¹⁰ The literature indicates that insurance companies offer third-party policies more often than first-party policies.¹¹

Two potential advantages are often cited regarding the transfer of information security risks. The Department of Commerce's Internet Policy Task Force argued that cybersecurity insurance vendors could potentially increase cybersecurity program adoption by promoting the widespread implementation of preventive measures

throughout the market and encouraging the adoption of best practices by linking them to insurance premiums.¹⁰ Moreover, I argue that organizations will have more money to invest in security safeguards and controls if they don't have to maintain—all by themselves—a large budget reserve to cover the potential costs of information security breaches. Cybersecurity insurance lets firms decide the portion of financial risk they want the insurance company to cover and the portion they want to bear on their own.

However, a few roadblocks hinder the realization of risk transfer's benefits. First, there's a lack of actuarial data, which causes insurance companies to charge (too) high premiums for first-party policies.

Second, insurance firms fear that so-called "cyberhurricanes" could overwhelm them. A cyberhurricane is a major computer-related cyberincident that results in a large number of claims. For example, imagine a Trojan horse virus spreading from Europe to Asia, imposing massive financial losses on numerous companies covered by the same insurer. This insurer could become bankrupt as a result of the massive losses caused by this single incident. Cyberhurricanes are problematic because there's not only a lack of data about such events—making it difficult for insurers to analyze and plan for them¹²—but there's also a lack of common cybersecurity standards across industries and limited knowledge about the effects of different cyberattacks.¹⁰

A third roadblock is that it's difficult to clarify and quantify covered cyber-related losses and to assign liability for those losses to actors.¹¹

Nevertheless, because companies are increasingly buying cyberinsurance to cover third-party losses, I argue that, in the future, more firms will realize how cyberinsurance positively affects the recovery function. The process of qualifying for cyberinsurance requires organizations to specify which of its assets are to be covered by insurance and define the scope of threat exposure to these assets. Realization of the scope of exposed assets should motivate IT leaders to create more effective recovery plans. In addition, for assets covered by insurance policies, insurance companies will likely offer lower premiums to companies with recovery plans that meet certain standards. Hence, firms that purchase cyberinsurance policies will be motivated to develop better recovery plans.

Therefore, I recommend that, within the recovery function, CSF include a *recovery planning* category with the following cyberrisk transfer outcomes, or subcategories:

- RC.RP-1—the recovery plan is executed during or after event,
- RC.RP-2—risk transferability analysis or audit is executed,

- RC.RP-3—the cyberrisk insurance policy is evaluated and acquired, and
- RC.RP-4—the cyberrisk insurance coverage is reviewed and updated.

Recognizing and Managing Implementation Risks

A challenging aspect of the NIST CSF is the management of implementation risks. Here, I discuss three such risks and propose guidelines for managing them.

Reducing Implementation Risks

To have an effective cybersecurity program, organizations must overcome three implementation risks:

- *implementation creep*—trying to implement a cybersecurity program across too many departments at once;
- *frameworks creep*—trying to combine too many frameworks into one universal framework; and
- *controls creep*—trying to deploy too many controls at once.

Each of these risks is likely to result in a longer implementation process, delayed positive results, loss of morale, and higher investments of money and effort. There's no single best, universal way of managing these risks. However, some general guidelines can be proposed based on extant work.

For example, Intel's pilot NIST implementation focused on two departments.¹³ In addition, Ted Gary recommended that firms prioritize business services based on risk assessment and then implement the most important controls for the highest-risk services.¹⁴ The Center for Internet Security Critical Security Controls (CSC) has already ranked and categorized 20 controls based on their criticality.¹⁵ It posits that the first five controls can reduce the risk of cyberattack by approximately 85 percent. Gary recommended that new NIST adopters implement the first five CSC controls. Hence, I propose the following thesis:

The less experienced an organization is, the more likely it is to be successful with a cybersecurity implementation if it

- selects three or so departments for the initial implementation,
- deploys one or two of the frameworks (rather than several), and
- focuses on initially implementing the most critical controls.

Because cybersecurity is a long-term venture, there's little to be gained from a rushed implementation process.

Understanding Risk Assessment's Role

Once the management team chooses the departments to focus on, it would then authorize risk assessments for these departments. The analysis would include the documentation of the known threats, breaches, and vulnerabilities. It would cover both the internal and external environments. The assessment would include an estimation of the likelihood of a cybersecurity event for specific information, processes, and technologies. There would also be an estimation of the potential impact of such cyberincidents on the organization. In addition, assessment of the regulatory environment would reveal security requirements that are imposed by laws and industry standards. After completing the risk assessments, the organization would have the information necessary to determine and describe its cybersecurity state of affairs, which is typically documented by implementation tiers.

Implementation Tiers

Because the implementation of a cybersecurity program is a journey rather than a one-time project, it's critical for organizations to have a way to document their current and future states. Progress could be monitored with a scoring scheme that makes it possible to compare where an organization is now to where it desires to be in future. There are two kinds of implementation tiers in the literature.

In the first, qualitative descriptors are used to differentiate the sophistication levels of cybersecurity implementations. One example is the implementation tiers proposed by NIST.⁹ NIST defines the stages of growth as four tiers, framing each stage as a progression of improvements that occur in terms of risk management, integrated risk management, and external participation. Table 1 (columns 1 and 2) describes the first two tiers. At the tier 1 stage, an adopter has undeveloped or insufficiently developed policies, practices, and risk management approaches. At the tier 4 stage, an adopter will manifest the most developed policies and practices. NIST's implementation tiers are not maturity levels. Hence, they don't have a scoring scheme.

In the second type, NIST's implementation tiers are customized by adding new categories and scoring schemes. Intel's custom tiers focus on a broader range of elements, such as people, technology, and processes, and use a scoring scheme ranging from 1 to 4 (highest maturity level). So, Intel's approach (Table 1, columns 3 and 4) is more detailed than NIST's.

How do organizations choose between the two approaches? The implementation tiers' content offers a hint. For example, if the central issue for a firm is monitoring the evolution of the risk management practices and culture, then NIST's implementation tiers would be

Table 1. A comparison of two implementation tier approaches.

NIST's implementation tiers		Intel's maturity level-oriented tiers	
Tier 1 (partial)	Tier 2 (risk informed)	Tier 1 (partial)	Tier 2 (risk informed)
Risk-management process		People	
<ul style="list-style-type: none"> – Cybersecurity risk practices are informal – Cybersecurity priorities aren't informed by the organization's risk objectives 	<ul style="list-style-type: none"> – Cybersecurity risk practices are approved – Cybersecurity priorities are informed by the organization's risk objectives 	<ul style="list-style-type: none"> – Lack of cybersecurity training – Lack of awareness of security risks 	<ul style="list-style-type: none"> – Employees have security training – Employees have awareness of risks and security resources
Integrated risk management		Process	
<ul style="list-style-type: none"> – Limited risk awareness at organization level – Practices are informal – Irregular implementation of security risk management 	<ul style="list-style-type: none"> – Awareness of security risk at the organization level but not organizationwide – Risk-informed, management-approved processes are defined and implemented 	<ul style="list-style-type: none"> – Informal risk management process – Lack of prioritization of threats into business decisions 	<ul style="list-style-type: none"> – Cyberactivities are risk informed – Management processes are risk informed – Cyberrisk information is shared – Staff has adequate resources to perform cybersecurity duties
External participation		Technology	
<ul style="list-style-type: none"> – Lack of processes to coordinate and collaborate with other entities 	<ul style="list-style-type: none"> – Firm knows its role but has no formal processes to coordinate and collaborate with other entities 	<ul style="list-style-type: none"> – Lack of tools – Poor tool management – Inadequate tool deployment – Technology lags behind current threats 	<ul style="list-style-type: none"> – Appropriate tools are deployed – Tools are maintained – Tools cover risk areas – Technology keeps pace with threats
		Ecosystem	

preferable. However, if an adopter is interested in creating a road map that covers the development of resources such as people, processes, technologies, and ecosystems, Intel's maturity-oriented framework¹³ would be more appropriate.

Even though Intel's tiers focus on vital resource areas and use a scoring scheme, it's still not easily connected to the NIST CSF categories. Hence, it can't be used to monitor a cybersecurity project's progress.

Capability Level-Based Implementation Tiers

Table 2 shows the structure of capability level-based implementation tiers (CIT). The columns of the framework are the NIST CSF functions, while the rows show

four capability levels, whose names were adopted from the NIST framework.⁹ The descriptions under each grid were developed using NIST's implementation tiers⁹ and Carnegie Mellon University's maturity model¹⁶ as references.

As Table 2 shows, each grid focuses on a functional area and depicts practices and assets indicative of each capability level. The capability levels are cumulative. For example, capability level 3 includes the level 2 requirements plus additional capabilities. The CIT scoring scheme is based on the concepts of capability and maturity levels. I modified and adapted the capability- and maturity-level definitions from Carnegie Mellon University's Software Engineering Institute.¹⁶ *Capability level* is defined as the scope of deployment of risk

Table 2. Capability level–based implementation tiers.

Level	Function				
	Identify	Protect	Detect	Respond	Recover
L1: partial (1–25 points)	The relevant outcomes are pursued by untrained staff, inadequate policies, using no/ few tools, ad hoc processes, inadequate technology, and no information references.	The relevant outcomes are limited by poor awareness and training, inadequate policies, few access controls, inadequate data security tools, ad hoc policies, and inadequate protective technologies.	The relevant outcomes are limited by poor detection of events, inadequate monitoring, ad hoc processes, and inability to recognize penetrations and invasions.	The relevant outcomes are limited by slow response to detected events due to poor response planning, lack of analysis, slow mitigation, and poor communications.	The relevant outcomes are limited by lack of recovery planning, poor recovery process practices and readiness, and lack of effective communications.
L2: risk informed (26–50 points)	The relevant outcomes are pursued by trained staff, using adequate policies, tools, and processes. The outcomes conform to expectations and are monitored, controlled, and reported.	The relevant outcomes are pursued by informed employees and trained staff, adequate policies, adequate access controls, adequate data security tools, adequate policies, and adequate protective technologies.	The relevant outcomes are pursued by informed employees and trained staff, adequate policies, event detection and monitoring tools, formal processes, and adequate ability to recognize penetrations and invasions.	The relevant outcomes are pursued by informed and trained employees who deploy adequate response planning, adequate analysis, mitigation capabilities, and communications.	The relevant outcomes are pursued by informed and trained employees who possess adequate recovery planning and readiness. Adequate communications and improvements are used.
L3: repeatable (51–75 points)	The relevant outcomes and practices are operated at capability level 2, but the policies and practices are now risk informed and updated to adapt to changing threats. The outcomes fall within acceptable risk tolerance.	The relevant outcomes and practices are operated at capability level 2, and risk-informed management is used to select, deploy, evaluate, and review fitness of controls, policies, access controls, data security tools, and technologies.	The relevant outcomes and practices are operated at capability level 2, and risk-informed management is used to determine appropriateness of detection and monitoring tools and formal processes.	The relevant outcomes and practices are operated at capability level 2, and risk-informed management is used to determine appropriate response plans, analysis, mitigations, and communications.	The relevant outcomes and practices are operated at capability level 2, and risk-informed management is used to determine appropriate recovery plans, improvements, and communications.
L4: adaptive (76–100 points)	The relevant outcomes and practices are operated at capability level 3 and the outcomes are regularly monitored, assessed, and reported organizationwide. The practices and policies are institutionalized and regularly assessed and improved.	The relevant outcomes and practices are operated at capability level 3, and protection controls are monitored, assessed, and reported organizationwide. The policies are institutionalized. The policies and controls are regularly assessed and improved.	The relevant outcomes and practices are operated at capability level 3, and the effectiveness of detection and monitoring tools is monitored, assessed, improved, and reported organizationwide. The practices and policies are institutionalized.	The relevant outcomes and practices are operated at capability level 3, and the effectiveness of response plans, analysis, mitigations, and communications is monitored, assessed, improved, and communicated. The practices are institutionalized.	The relevant outcomes and practices are operated at capability level 3, and the effectiveness of recovery plans, analysis, mitigations, and communications is monitored, assessed, improved, and communicated.

Table 3. Example of an organization’s current and target profiles based on capability levels.

Function	Category	Current		Target		Capability gap (G)	Weight (W)	Priority (W * G)
		Capability profile	Maturity	Capability profile	Maturity			
Detect	Anomalies and events	20	Level 1	55	Level 3	35	3	105
	Security continuous monitoring	20		65		15	2	30
	Detection processes	10		50		40	3	120
Respond	Response planning	28	Level 2	35	Level 2	7	3	21
	Communication	25		35		10	2	20
	Analysis	30		55		25	2	50
	Mitigation	35		60		25	2	50
	Improvements	12		20		8	1	8
Recover	Recovery planning	25	Level 1	35	Level 2	10	3	30
	Improvements	20		30		10	2	20
	Communication	10		28		10	1	18

management practices, training, appropriate resources, policies, and procedures in pursuing the goals of a functional area of the NIST framework.

The first column shows each capability level’s range of scores, adopted from Stephen Coraggio and his colleagues’ work.¹⁷ To make the scores meaningful, they must be linked to something objective. Jason Christopher and his colleagues proposed that scores be based on the level of completion of the controls in a functional area, with four progress milestones.¹⁸ I extended these to the following five classes: fully implemented (76–100), largely implemented (51–75), somewhat implemented (26–50), partially implemented (1–25), and not implemented (0).

Maturity level is defined as the achievement of a threshold capability level for a specific number of categories associated with a functional area of the NIST CSF. An organization should map capability scores to maturity levels in a manner that serves its business mission. For example, a mapping rule similar to the following could be adopted: maturity level 1 means that 70 percent or more of categories are assigned a capability level 1 rating, maturity level 2 means that 70 percent or more of the categories are assigned a capability level 2 rating, and so on.

Linking capability to maturity levels has two advantages. First, the scheme reinforces that capability

acquisition is the path to improved maturity levels. Second, it proposes that maturity levels will improve more if organizations become competent in a broad range of categories within a function, rather than just a few areas.

Developing Capability Level-Based Profiles

The NIST CSF proposed the use of *profiles*—road maps to guide the implementation of cybersecurity projects over time.⁹ The “current profile” indicates an organization’s current cybersecurity outcomes, while the “target profile” depicts future outcomes. Table 3 shows an example of capability level-based profiles.

In this sample profile, the detect function is assigned maturity level 1, even though the mean capability score is 26.67. In contrast, although the mean capability score for the respond function is 26, this function is assigned maturity level 2. This is because four of the categories (that is, 80 percent) in the respond function exceed the threshold capability level score of 25. Only 33 percent of the detect function’s categories have 25 points or higher. Table 3 also shows the *capability gap*, which is the difference between current and future capability levels, as well as the *weight*, or priority given to each capability,¹⁷ where 0 = unimportant, 1 = valuable, 2 = important, 3 = both urgent and important (critical). The product of gap and weight yields a score that lets management

assign resources, beginning with the highest weighted gap score.

NIST's CSF is changing the way cybersecurity is implemented across various US industries. I hope that the recommendations and examples provided here will enhance the implementation and future adoption of and changes to this valuable framework. ■

References

1. *Is Your Company Ready for a Big Data Breach? The Second Annual Study on Data Breach Preparedness*, report, Ponemon Inst., Sept. 2014; www.experian.com/assets/data-breach/brochures/2014-ponemon-2nd-annual-preparedness.pdf.
2. B. Obama, "Executive Order—Improving Critical Infrastructure Cybersecurity," The White House, 12 Feb. 2013; obamawhitehouse.archives.gov/the-press-office/2013/02/12/executive-order-improving-critical-infrastructure-cybersecurity.
3. "Using the NIST Cybersecurity Framework," webinar, Gartner, 2015; www.gartner.com/webinar/3163821/player?commId=180719&channelId=5500&srcId=1-4730952011.
4. J. Guinn, "Why You Should Adopt the NIST Cybersecurity Framework," PwC, 2015; www.pwc.com/cybersecurity.
5. *Understanding HITRUST's Approach to Risk vs. Compliance-Based Information Protection*, white paper, Health Information Trust Alliance, May 2014; hitrustalliance.net/documents/csf_rmf_related/RiskVsComplianceWhitepaper.pdf.
6. *Verizon 2015: PCI Compliance Report*, Verizon, 2016; www.verizonenterprise.com/resources/report/rp_pci-report-2015_en_xg.pdf.
7. G. Stoneburner, A. Goguen, and A. Feringa, *Risk Management Guide for Information Technology Systems: Recommendations of the National Institute of Standards and Technology*, NIST Special Publication 800-30, NIST, 2002; csrc.nist.gov/publications/nistpubs/800-30/sp800-30.pdf.
8. *Guide for Conducting Risk Assessments—Information Security*, NIST Special Publication 800-30, rev. 1, NIST, 2012; nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-30r1.pdf.
9. *Framework for Improving Critical Infrastructure Cybersecurity*, NIST, version 1.0, 12 Feb. 2014; www.nist.gov/sites/default/files/documents/cyberframework/cybersecurity-framework-021214.pdf.
10. *Cybersecurity, Innovation and the Internet Economy*, Dept. Commerce Internet Policy Task Force, June 2011; www.nist.gov/sites/default/files/documents/itl/Cybersecurity_Green-Paper_FinalVersion.pdf.
11. *Cybersecurity Insurance Event Readout Report*, US Dept. Homeland Security, Nat'l Protection and Programs Directorate, Nov. 2012; www.dhs.gov/publication/cybersecurity-insurance.
12. O. Suess, "High-Profile Cyber Attacks Drive Growth of Cyber Crime Insurance," *Insurance J.*, 10 May 2017; www.insurancejournal.com/news/international/2017/05/10/450464.htm.
13. T. Casey et al., *The Cybersecurity Framework in Action: An Intel Use Case*, solution brief, Intel, 2015; www.intel.com/content/www/us/en/government/cybersecurity-framework-in-action-use-case-brief.html.
14. T. Gary, "Portland Uses the Cybersecurity Framework and Critical Security Controls Aligns Business Risk and Security," Tenable Blog, 14 June 2016; www.tenable.com/blog/portland-uses-the-cybersecurity-framework-and-critical-security-controls.
15. "CIS Controls," Center for Internet Security, 2017; www.cisecurity.org/controls.
16. M.-B. Chrissis, M. Konrad, and S. Shrum, *CMMI—Guidelines for Process Integration and Product Improvement*, Addison-Wesley, 2003.
17. S. Coraggio, J. Rogers, and N. Hilgeman, *NIST Cybersecurity Framework—Implementing the Framework Profile*, Booz, Allen, & Hamilton White Paper Commercial Solutions, 2015; www.boozallen.com/content/dam/boozallen/documents/2015/07/nist-cybersecurity-framework.pdf.
18. J.D. Christopher, N. Bartol and E. Goff, "NIST Cybersecurity Framework Implementation: Energy Sector Approach," *Proc. SANS ICS Security Summit*, 2014; www.sans.org/summit-archives/file/summit-archive-1493737804.pdf.

Adenekan Dedeke is an executive professor of supply chain and information management in the D'Amore-McKim School of Business, Northeastern University. His research interests include ethics, security frameworks, information quality, privacy, and the impact of information technologies on work and decision making. Contact him at a.dedeke@northeastern.edu.

myCS

Read your subscriptions through the myCS publications portal at <http://mycs.computer.org>

152 Simple Steps to Stay Safe Online:

Security Advice for Non-Tech-Savvy Users

Robert W. Reeder, Iulia Ion, and Sunny Consolvo | Google

Users often don't follow expert advice for staying secure online, but the reasons for users' noncompliance are only partly understood. More than 200 security experts were asked for the top three pieces of advice they would give non-tech-savvy users. The results suggest that, although individual experts give thoughtful, reasonable answers, the expert community as a whole lacks consensus.

With almost daily news of high-profile cybersecurity incidents, users naturally wonder what they can do to protect themselves against attacks. Indeed, as cybersecurity professionals, we're often asked by concerned friends and family for advice on what to do to stay safe online. But, somewhat to our own surprise, we're dumbfounded about what to say in these situations. On one hand, we could say hundreds of things about online security; after all, the security field is so complex, it takes years to learn. On the other hand, those asking us for advice just want a few easy-to-remember things they can start applying right away. Getting from the hundreds of things down to a handful of the most important is surprisingly challenging.

We set out to find the most important security advice on offer from experts today. Our goal was to find advice for a general audience that could be used, for example, in a public awareness campaign or on an informational website. To inform such general cybersecurity communications, the security field should have a consistent, prioritized set of advice that can be shared with those users looking for the most important things to start doing right away. The entire set might be long, but as long as the most important things are consistently communicated to users at large, users will

have a better chance of understanding and remembering them.

Our approach has its limitations. There are many different computing contexts, and good advice can be highly context dependent. Advice that works for one user might be irrelevant or impossible to follow for another. In some cases, users need assistance to respond to some specific situation, and providing such assistance is important—but it's not our goal. Although there's a need for contextualized advice and assistance, this work targets a different need: the most important advice to share with a general audience.

We Asked the Experts

Our work is guided by two primary research questions: What advice do security experts consider most important? And is there expert consensus and consistency on what advice is considered most important? To identify the prevailing advice of the security community, we surveyed 231 security experts and asked them to name the top three pieces of advice they'd give to a non-tech-savvy user to protect their security online.

Our results provide a broad sample of expert opinion about the highest-priority advice to share with users and reveal a lack of expert consensus. Moreover, on examining

the advice we collected more closely, we found several areas with confusing advice variants (for instance, not clicking on links in email from unknown sources versus not clicking links in email at all). Although almost all of the thoughtful advice we received makes sense in isolation, the security expert community isn't in agreement on how to prioritize the set of advice as a whole or on how to resolve confusing variants in the set. It's understandable if users are confused about what to do; even experts, as a field, don't seem to agree.

Although the question of what advice to give seems fundamental to online security, we identify some clear problems with the existing set of expert advice. We acknowledge that arriving at consensus about the right set of advice is quite difficult, and we don't solve that problem in this article. Instead, we contribute

- data on existing expert opinion on what security advice to give to nonexpert users,
- an analysis of the consensus and consistency of the overall set of advice we found, and
- identification of the problem that the set of the most important security advice isn't widely agreed on.

Background and Related Work

Although we're not aware of past research that has evaluated the state of security advice as a whole, there has been extensive research on advice in specific areas and users' struggles to follow it. We give a brief overview of sources of security advice and research on users' compliance with it.

A great deal of security advice is available to those looking for it. Many service providers, enterprises, universities, and other organizations offer advice in the form of tips and training on how to stay safe online. One of the most comprehensive and authoritative sources of advice intended for nontechnical users is provided by US-CERT (www.us-cert.gov/ncas/tips), which by our count spans 57 pages and offers 534 individual pieces of advice. Recommendations range from common advice like "keep your antivirus software current" to less common advice like "consider challenging service providers that only use passwords to adopt more secure methods." With such a large set of advice, it might be unclear to many users where to get started, to whom the advice applies, and why following the advice will help.

Past research on security advice and users' security behaviors suggests that there's an opportunity for advice to change behavior for the better but also a need to limit, prioritize, and better communicate the advice.

Opportunity to Change Behavior

If users weren't willing or able to take any security measures, formulating good advice would be a moot issue.

However, past work has found that users do have some, albeit limited, willingness and ability to follow good security practices. We surveyed security experts and nonexperts about their security practices and found that nonexperts clearly do follow security practices, but often not the same ones experts do.¹ These findings suggest a need to better communicate expert practices and advice to nonexperts. Rick Wash examined users' reactions to 12 common pieces of security advice and found that users would follow some diligently while ignoring others, depending on their mental models of security.² In a previous study, we found that users—at least those who've experienced an account hijacking—generally accept some responsibility for protecting their online accounts and acknowledge their role in security behaviors like selecting and protecting passwords.³

Need to Limit, Prioritize, and Communicate

Cormac Herley argues that users often reject security advice because the cost of following all commonly given security advice is much greater than the cost of the relatively few low-frequency attacks that succeed.⁴ He argues in another work that, for security advice, "more is not the answer" but acknowledges that some advice is probably needed.⁵

How advice is communicated is a critical part of getting users to follow it. Emilee Rader and her colleagues show that people learn lessons about security via stories they hear, that these lessons can change behavior, and that stories might thus be an effective way to communicate advice to users.⁶

Methodology

We conducted an online survey of security experts about the security advice they would share with non-tech-savvy users. We used Google Forms (www.google.com/forms/about) to write and host the survey, which ran from February through June 2014. We recruited security experts via the Google Online Security Blog⁷—a public blog that is published by Google and widely read by security experts and enthusiasts—and by promoting the survey through our social media accounts. Participation in the survey was voluntary, and we didn't provide compensation. We considered a "security expert" to be anyone who reported having at least five years of experience working in or studying computer security. Our results are based on responses from 231 such expert respondents.

Survey Content

The survey started with the following single, open-ended question:

What are the top three pieces of advice you would give to a non-tech-savvy user to protect their security online?

The survey also asked demographic questions, quality-assurance questions, and a series of other questions, which are reported in our work comparing expert and nonexpert security practices.¹

We chose to elicit qualitative, freeform responses to our top-three-advice question, rather than the quantitative responses that multiple choice or Likert-scale questions would provide. Qualitative data can be difficult to analyze and introduces risks of subjective interpretation by experimenters, but it maximized our chances of getting experts' unvarnished opinions.

We received 245 responses to our survey from experts meeting our criteria of five years or more of security experience. Of these, we eliminated 14 from analysis for incorrectly answering two or more of our four quality-assurance questions.

Security Expert Demographics

Security professionals often have demanding jobs and are highly paid, so we expected a small sample, perhaps a few dozen, to be willing to complete our survey for free. Ultimately, many security experts responded, giving us a sample size and diversity that exceeded our expectations.

Respondents reported diverse geographies, workplaces, and job titles. While 47 percent of respondents were from the US, others were from 25 countries around the world, including, in order of frequency, the UK, Germany, Australia, Japan, India, Israel, and South Africa. In a check-all-that-apply question, 69 percent reported working in industry, 15 percent in academia, 13 percent self-employed, 11 percent in government, and 7 percent in corporate research labs. Respondents reported a vast range of job titles in information security including chief executive officer, chief information security officer, consultant, graduate student, IT specialist, network administrator, security researcher, software engineer, and whitehat hacker.

Of the 231 respondents in our sample of experts, 4 percent were female. Ages ranged from 18 to over 65, with 2 percent in the 18–24 range, 30 percent in the 25–34 year-old range, 32 percent in the 35–44 range, 18 percent in the 45–54 range, 9 percent in the 55–64 range, 3 percent over 65, and 5 percent not providing their age.

Coding Procedure

We analyzed freeform responses to the top-three-advice question using a general inductive approach.⁸ Two of the authors served as raters. The two raters, working independently, read a subset of the responses and proposed codes for common responses. They then met to discuss the codes and agreed on an initial codebook. Having formed an initial set of codes, the raters split

up the data and began coding responses independently. They coordinated to add new codes to the codebook as needed. To assess interrater reliability, both raters independently coded the same subset of our data (10 percent of our sample) using the final codebook and achieved a Cohen's κ of 0.77, which is generally considered substantial agreement.⁸

Ethics

Only voluntarily provided survey data was collected and analyzed for this work. Our organization doesn't have an institutional review board (IRB), so the study wasn't subject to IRB review; however, multiple researchers who have received human subjects training reviewed the survey instrument prior to the experiment. Respondents weren't required or asked to identify themselves. Raw survey data access was restricted to investigators on the research team.

Limitations

Although the sample's size and diversity give us some confidence that it's representative of a large portion of the security expert community, our recruiting methods could introduce sample bias, as virtually all recruiting methods can. Because we recruited via the Google Online Security Blog, it's likely respondents are regular readers of the blog, so they might feel some loyalty to Google. For most security advice, this loyalty probably makes no difference, but some bias might be present in advice, such as the recommendation to use Chrome. We note, however, that some respondents recommended products made by other organizations as well.

Results

Having coded all survey responses, we deemed each code to represent a piece of advice. We assigned 837 codes to our 231 experts' responses (some responses were coded as providing more than three pieces of advice). Of these 837 pieces of advice, 152 were unique. Having found 152 unique pieces of advice, we then counted the frequency of each piece of advice received—that is, how many unique experts mentioned each piece of advice. Our frequency count of 68 for “use unique passwords,” for example, means 68 unique experts mentioned that piece of advice. These frequency counts form the basis of our results. Because we collected such a wide variety of advice, we assigned pieces of advice to categories to make the advice easier to understand and present. We then counted the number of unique experts giving at least one piece of advice in each category.

Table 1 shows the 45 pieces of advice (of the 152 total pieces of advice) that were mentioned by four or more experts, grouped by category. Table 2 provides examples of quotes that were coded as some of

Table 1. The 45 pieces of advice that at least four respondents mentioned.

Advice	Count	Representative quotes
Account security	128	
Use unique passwords	68	Different passwords everywhere. Do not reuse passwords on multiple sites.
Use strong passwords	58	Choose a strong password. Complex password for every site.
Use multifactor authentication	36	Enable multifactor authentication features, if available.
Use a password manager	33	Forget your password—use a password manager to remember it for you.
Use a passphrase	7	Use a passphrase. Use long-form plain language passwords.
Write passwords down	5	Write them down in a notebook and keep it safe.
Other account security	24	Routinely change passwords. Don't leave a shared computer logged in as you.
Updates	97	
Keep systems and software up to date	90	Always be updating (OS and applications). Patch, patch, patch.
Use automatic updates	19	Activate autoupdate.
Other updates	0	
Browsing habits	76	
Use HTTPS	24	Use HTTPS if available. Watch for and understand why HTTPS is important.
Be careful/think before you click	19	Think before you click. Be careful what you click on.
Check URL for expected site	11	Always look at the URL bar to confirm that it's the right site.
Check the hyperlink before you click	8	Examine a link before you click it. Compare links via mouse hover with printed link.
Sensitive info only over HTTPS	6	Check for HTTPS every time you provide personal/sensitive data.
Check for lock icon	5	Look for the lock.
Pay attention to security warnings	5	Don't click through security warnings. Don't ignore security warnings—they are there for a reason.
Check for HTTPS in the URL	4	Check for a green HTTPS to the left of the domain name.
Visit only reputable websites	4	Don't enter sites whose reputation isn't clearly (and positively) assessed in a public database.
Other browsing habits	19	Take the time to read before clicking. Check SSL certificates.
Email habits	59	
Don't open unexpected attachments	19	If you didn't ask for the attachment, don't open it.
Don't click links in emails at all	11	Never click on a link in an email.
Don't click links in email from unknown sender	9	Don't click on links or images in an email from an unknown source.
Be suspicious of email in general	7	Don't trust email. Be skeptical about email.
Be alert for phishing emails	5	Beware spam and phishing emails. Don't fall for phishing attempts.
Beware emails requesting private data	5	No legitimate financial institution will ask for your personal or financial information through email.
Be suspicious even of email from known sender	4	Don't blindly trust every message even if it came from someone you know and trust.
Be suspicious of links in email	4	Be careful following links, especially in email.
Other email habits	19	If a message you receive seems strange, pick up the phone and verify it.
Mindfulness	42	
Be suspicious in general	16	Be skeptical. Always be suspicious; don't trust everybody.
Too good to be true probably is	15	If it seems too good to be true, it likely is. Be aware of "too-good-to-be-true" offers.

Table 1. The 45 pieces of advice that at least four respondents mentioned.

Advice	Count	Representative quotes
Apply real-world judgment online	4	Common sense. Think “would I do this out in the real world?”
Other mindfulness	19	Stay alert, because you are in charge. Assume you don’t understand the risks.
Antivirus	41	
Use antivirus software	35	Use antivirus/antimalware software.
Keep antivirus software up to date	16	Keep antimalware current. Keep antivirus updated.
Other antivirus	3	Leverage two antivirus engines.
Privacy	30	
Limit personal information sharing	14	Never give out personal information. Share less. Don’t give out your email.
Be careful what you share	13	Be wary of information you post on social media.
Other privacy	5	Remain anonymous as much as feasible and practicable. Always browse in private mode.
Browser software	29	
Use Chrome	13	Use Chrome to browse the web.
Use an ad blocker	5	Use a modern browser with an Adblock and Web Reputation add-on.
Don’t use Java	4	Disable Java browser plug-ins or uninstall Java.
Other browser software	17	Run NoScript browser add-on. Disable third-party cookies.
Device security	24	
Don’t run as admin	12	Limit privileges. Don’t log in as an admin unless necessary.
Do sensitive tasks on dedicated devices	4	Use separate devices for casual browsing ... and sensitive ones.
Do sensitive tasks on trusted devices	4	Do online banking/purchases only on a trusted computer.
Lock devices	4	Put passwords/PINs on all your devices. Lock your phone.
Other device security	0	
Software security	22	
Use only software from trusted sources	20	Execute only software coming from reputable websites.
Other software security	2	Only install software you absolutely need.
Network security	15	
Don’t trust open networks	4	Don’t use free/open Wi-Fi. Don’t trust open networks or three-party networks; this can be unsafe.
Other network security	11	Use a VPN service. Keep your firewall turned on. Use a hardware firewall at home.
Backups	10	
Back up your data	10	Back up your data; nothing beats a good backup. Always back up your data.
Other backups	0	
Education	11	
Learn about security	4	Educate yourself on common security problems.
Seek expert help when needed	4	Get help if you are uncertain—quickly. If in doubt, ask.
Other education	3	Be aware of why your computer asks you for permission or passwords.
OS and platform	9	
Use an uncommon OS	4	Using a less-common OS makes you less likely to be attacked.
Other OS and platform	5	If you know how to deal with virtual machines, use them. If possible, use Linux.
Other	34	

Table 2. Examples of less common advice provided by respondents.

Always browse in private mode, and delete cache after each browsing session.
Always double-check the source of an email (the sender).
Disable root certificates for entities that you would be alarmed to see certifying your bank's login page.
Don't write down passwords.
Don't add absolute strangers to your social media accounts.
Don't click on ads.
Don't look for porn.
If you notice anything suspicious, report it appropriately.
If you travel, use the Tor browser from your encrypted hard drive.
Install Microsoft EMET (Enhanced Mitigation Experience Toolkit) and turn the systemwide settings up to maximum.
Let Gmail render your mail attachments instead of opening them locally.
Make sure to set up account recovery options for your Google account.
Never install or upgrade software from a popup screen.
Unless you really know what you're doing, you're better off with documents in the cloud.

the 107 pieces of advice mentioned by three or fewer experts.

Our 837 codes assigned to 231 responses gives an average of 3.26 (with a standard deviation of 1.24) codes assigned per response. Even though the top-three-advice question asked for three pieces of advice, some responses received either more than or fewer than three codes, either because respondents deliberately provided a number other than three pieces of advice, or because the advice a respondent provided as one piece received more than one code (for example, we assigned “make sure your computer and its antivirus software are kept up to date” codes for “keep systems and software up to date” and “keep antivirus software up to date”).

In cases in which related advice was given at different granularity levels, for example, “be suspicious in general” versus “be suspicious of links in email,” we strove to create codes that stayed true to the literal responses from respondents. In these cases, we assigned different codes to both the more generic and the more specific pieces of advice. We elaborate on this issue further in the discussion on generic versus specific advice.

Advice Collected, by Category

We grouped the pieces of advice into 15 categories. In order of the number of unique experts mentioning at least one piece of advice in the category, these

categories were account security, updates, browsing habits, email habits, mindfulness, antivirus, privacy, browser security, device security, software security, network security, backups, education, OS and platform, and other.

Pieces of advice mentioned by three or fewer experts fall into either category-specific “other” advice, or the general “other” category for advice that matched none of the 14 established categories. Category counts shown in Table 1 are unique experts mentioning at least one piece of advice in the category.

Most-Mentioned Advice

As Table 1 shows, the top three pieces of advice the security expert community would give to a non-tech-savvy user are “keep systems and software up to date,” “use unique passwords,” and “use strong passwords.” However, we caution against prioritizing the entire set of advice strictly by rank-ordering the advice by the count of experts who mentioned it. The problem with this approach is that we didn't ask experts to compare one piece of advice against another; we simply asked each individual for his or her own version of the top three. In any case, Table 3 shows the 10 (11 actually, because there is a three-way tie for ninth) most-mentioned pieces of advice, with number of respondents mentioning them.

Discussion

Our results give a sense of the security expert community's overall thoughts on the most important advice today. Much of the advice we collected is familiar, and almost all of it seems reasonable in isolation. It appears that expert respondents to our survey gave thoughtful and sensible responses. But our finding that there are 152 pieces of advice spread across 15 categories suggests a wide breadth of security advice that experts consider important to follow. Just considering these numbers, it's perhaps unsurprising that users don't follow all the advice on offer—there's a lot of it, it spans diverse areas, and it's not clear where to start. Users are probably not receiving a consistent message on what's most important and exactly what to do in each area.

We start our discussion by establishing criteria for what makes good general advice. We then report a series of observations about the advice we collected, discuss challenges with creating good advice, and suggest ways in which the set of advice as a whole might be improved.

Criteria for Good General Advice

We guide our discussion of the advice we found and the potential for improving it by first establishing four criteria that good general advice should meet. These criteria are drawn from work in public awareness communications, which highlights the need for advice that users believe will work (our *effective* criterion), that users can actually do (our *actionable* criterion), and that is understandable (our *consistent* and *concise* criteria).⁹

Effective. Good advice, if followed by a user, should actually improve the user's security situation and lead to better security outcomes. Almost all the advice we collected in this study (see Tables 1 and 2) seems effective against some security threat. Doing almost any of the actions advised by security experts (for instance, using strong passwords) should help improve users' online security.

Actionable. Good advice should be easy for a user to remember and apply when needed, and it shouldn't overly interfere with a user's primary goals. Advice that requires excessive skill (for instance, running a virtual machine), requires expert knowledge (for instance, requiring a user to judge something as "suspicious"), or excessively restricts user activity (for instance, "simply stay offline") might not be reasonably actionable for a user seeking general advice. Although most of the advice we collected is actionable (for instance, "use multifactor authentication"), some advice is less actionable (for instance, "be suspicious in general").

Table 3. Ten most mentioned pieces of advice, coded.

Advice	No. of respondents who mentioned
Keep systems and software up to date	90
Use unique passwords	68
Use strong passwords	58
Use multifactor authentication	36
Use antivirus software	35
Use a password manager	33
Use HTTPS	24
Use only software from trusted sources	20
Use automatic updates	19
Be careful/think before you click	19
Don't open unexpected attachments	19

Consistent. Good advice should be both internally consistent—in that it shouldn't cause confusion with or subsume other advice in the whole set of advice—and presented consistently—in that it should be phrased similarly each time a user hears it and should change as little as possible over time (as long as it remains effective). Consistency helps make advice easier for users to understand, remember, and follow. Looked at as a whole, the body of advice we collected wasn't consistent. The same advice was phrased differently by different participants, and a few pieces of advice were contradictory (for instance, "write passwords down" and "don't write down passwords").

Concise. The set of advice as a whole should be as small as possible. Less advice is easier for users to remember than more advice, and less advice to follow means it's easier to follow all of it. The ultimate goal of our work is to create more concise advice. Given that we found 152 pieces of advice in this study, future work is needed to distill the 152 pieces of advice and communicate to users the most important ones.

Observations about Advice We Collected

We point out several observations about the advice we collected. These observations arose as we considered how the advice as a set could better meet our criteria.

Consensus within categories. Overall, we found a lack of consensus regarding the top three pieces of advice. But looking at our results by category, we find both pockets of consensus and pockets of divergence. Advice in the updates category was consistent that all software and systems should be kept up to date. The other common piece of advice in this category—to enable automatic updates—is clearly in service of the first. Antivirus, privacy, software security, and backups were categories with similar levels of general consensus. However, categories like account security, browsing habits, email habits, mindfulness, and browser software contain numerous pieces of advice, many of them potentially confusing variants or hard-to-discern options. For example, account security contains advice to “use a password manager,” “use a passphrase,” and “write passwords down.” These pieces of advice are all options for solving the same problem: helping a user set strong and unique passwords but still manage to recall them when needed. Each method has its pros and cons, as security experts know. But how is a security nonexpert to choose among these techniques? The nonexpert confronted with all three pieces of advice is likely to be confused.

There’s a lot of important advice. We set out with a goal to find just a handful of the most important advice that could be communicated to users whenever we have a few moments of their attention. Given our finding of a diverse range of advice, all of which is considered important by at least some experts, it might be the case that the security space is simply too complex for a small set of consistent advice to adequately protect the general user population. Perhaps advice communication efforts should focus not on communicating the same advice consistently to everyone, but on identifying particular audiences and customizing advice for each audience.

From “set and forget” to near-constant vigilance. Advice varies in the frequency with which it needs to be applied. Some is “set and forget”—it needs to be done once (or rarely) and can then be ignored—some is needed on occasion, and some requires near-constant vigilance. In the set-and-forget category are pieces of advice like “use antivirus software” and “use automatic updates.” Good antivirus software or automatic updates should require little user interaction after they’re initially set up. Advice needed on occasion includes advice related to choosing passwords and advice like “do sensitive tasks on dedicated devices” and “back up your data.” Much advice requires ongoing vigilance, like most of the browsing habits, email habits, mindfulness, privacy, and education advice. Negative advice, like “don’t run as admin” or “don’t trust open networks,” falls somewhere in between; it

should be noted once, then applied whenever an applicable situation comes up (like considering whether to use the Wi-Fi at a coffee shop).

In general, vigilance might require cognitive attention, so it can be difficult for users. Any advice that requires ongoing vigilance or frequent application should be given to users only if it has high efficacy.

Generic versus specific. Variants of advice in the same area often differed in their level of specificity. Some advice was quite generic, like “use HTTPS,” whereas other advice was more specific, such as to “send sensitive info only over HTTPS.” Or, to compare respondents’ quotes,

Always browse with HTTPS if you can

represents a generic form of advice, whereas

Always look out for the HTTPS and padlock logo when entering credit card details

represents a very specific version of similar advice.

There are arguments in favor of both generic and specific advice. Generic advice applies in more situations and to more users, whereas specific advice is usually more clearly actionable. Non-tech-savvy users instructed to follow the generic advice, “always browse with HTTPS” would have to learn what HTTPS is and how to determine whether they’re browsing with it. However, users instructed to follow the more specific “look for the padlock when entering credit card details” would already have a way to determine whether HTTPS is in use, but might fail to apply that knowledge when entering sensitive data other than credit card details.

Generic advice can help keep the overall set of advice concise, because it doesn’t require enumerating every situation in which the advice should apply and every detail of how to apply the advice. However, generic advice might require skills and judgment that non-tech-savvy users haven’t developed well, such as the advice to “use only software from trusted sources,” which requires careful judgment about how to determine the source of the software and which should be trusted.

Given the merits of both generic and specific advice, balancing them is important. Sometimes, it might be possible to combine them by offering the generic advice followed by specific instructions on how to implement it, for instance, “Always browse with HTTPS if you can; to check for an HTTPS connection, look for the padlock logo in the browser’s address bar.”

Realistic for users to follow. Some advice we collected is likely not actionable because users can’t follow it,

either because it's too restrictive or because it requires too much technical knowledge or skill. Advice like "don't click links in email at all" is probably too restrictive; for many users, advice like "do sensitive tasks on dedicated devices" is probably too restrictive if they can't afford multiple devices. Advice like "don't run as admin" and "use an uncommon operating system" probably requires more technical knowledge than many users have.

Phrasing advice. Even advice to which we assigned the same codes could vary significantly in how experts phrased it. Examples of representative quotes from Table 1 show variants in respondents' phrasing of advice. Here are two quotes from respondents that were both assigned the code *Too good to be true probably is*:

If it is too good to be true, looks like a scam, smells like a scam, or wants your personal details, IT IS A SCAM.

and

A Nigerian Prince would never ask you to launder money for them, nor would the FBI director, etc.

The former quote is more direct and explicit in advising users to trust their instincts and judgment about online offers. The latter contains narrative examples and suggests a lesson without explicitly stating it. It's hard to say which would more likely connect with users, but these examples illustrate the variety of potential ways to phrase the same advice.

Challenges in Creating Good Advice

Our results suggest several challenges in creating good advice. As improvements to the overall state of advice are attempted, it's worth bearing these challenges in mind.

The right advice might change over time with the attack landscape, new technology, and experience. As new attacks arise, new pieces of advice might need to be communicated to users to address them. To make the challenge even harder, attackers might adapt as good advice is adopted. For example, the widespread adoption of antivirus software has presumably made rogue antivirus attacks viable for attackers.¹⁰

Advice that was once thought good might go out of style with experience or other changes. For example, Anne Adams and M. Angela Sasse's 1999 work talks about the difficulty users had with the advice to change passwords frequently,¹¹ which was common advice at the time, but seems to have fallen out of favor (only three of our experts mentioned "change passwords frequently").

Changing advice is a risk to consistency of the advice set. Some change in the set of security advice over time is undoubtedly necessary—and even desirable when it leads to a smaller set of advice or adapts to new threats—but all things being equal, advice that stays constant over time is more likely to be followed than advice that's likely to change.

Even advice that's otherwise good—effective and consistently delivered—can face poor adoption if users don't believe the advice is effective or if they encounter significant drawbacks as a result of following the advice. For example, Kami Vaniea and her colleagues discuss some of the reasons users often reject the advice to install updates, such as the bundling of undesired new features with security updates and the potential for an update to break a working system.¹²

It simply might not be realistic to have a small, consistent set of security advice for general use. However, prioritizing the set to make it easier for users to apply the most important pieces first seems especially important.

Improving the Existing Set of Advice

Improving the state of security advice from today's rather scattered state to a more effective, actionable, consistent, and concise set of advice is no small task. Our exercise here—surveying the current state of top advice according to experts—is only a start; it merely reveals the extensive effort needed to produce a good set of advice.

Advice should also be informed by actual data about attacks, compromises, and breaches. For example, if data on account compromises suggests that password brute-forcing attacks are most prevalent, we should emphasize using password managers. However, this data is difficult to obtain; often, the causes of security issues like account compromise or database breaches are unknown. In other cases, there's reluctance to release such data publicly.

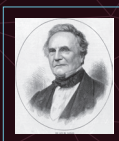
Once the existing set of advice has been pared down to a more concise and internally consistent set, it should be given to users and evaluated in longitudinal studies in which users are observed as they try to apply the advice over time and in multiple relevant situations. Such studies can inform questions about what advice is memorable, easy enough for users to follow, not overly restrictive, and actually likely to produce better security outcomes.

We hope our findings will help focus research on the right set of advice to communicate to users and on what advice is most important and what can be deprioritized. In addition, we seek to alert the usability

and security communities to some of the difficulties users might have following the advice on offer today. We hope usability and security experts will focus on each piece of advice on our list and consider it carefully for inclusion in the set of advice as a whole, according to our four criteria. Through data-informed debate, the communities can pare the set down, prioritize it, standardize the way it is phrased, and package it for more effective dissemination to non-tech-savvy users. ■

References

1. I. Ion et al., "... No One Can Hack My Mind': Comparing Expert and Non-expert Security Practices," *Proc. Symp. Usable Privacy and Security (SOUPS 15)*, 2015, pp. 327–346.
2. R. Wash, "Folk Models of Home Computer Security," *Proc. Symp. Usable Privacy and Security (SOUPS 10)*, 2010, pp. 1–16.
3. R. Shay et al., "My Religious Aunt Asked Why I Was Trying to Sell Her Viagra: Experiences with Account Hijacking," *Proc. SIGCHI Conf. Human Factors in Computing Systems (CHI 14)*, 2014, pp. 2657–2666.
4. C. Herley, "So Long, and No Thanks for the Externalities: The Rational Rejection of Security Advice by Users," *Proc. New Security Paradigms Workshop (NSPW 09)*, 2009, pp. 133–144.
5. C. Herley, "More Is Not the Answer," *IEEE Security & Privacy*, vol. 12, no. 1, 2014, pp. 14–19.
6. E. Rader, R. Wash, and B. Brooks, "Stories as Informal Lessons about Security," *Proc. Symp. Usable Privacy and Security (SOUPS 12)*, 2012, article 6.
7. R.W. Reeder, "If You Could Tell a User Three Things to Do to Stay Safe Online, What Would They Be?," Google Online Security Blog, 26 Mar. 2014; googleonlinesecurity.blogspot.com/2014/03/if-you-could-tell-user-three-things-to.html.
8. J.R. Landis and G.G. Koch, "The Measurement of Observer Agreement for Categorical Data," *Biometrics*, vol. 33, no. 1, 1977, pp. 159–174.
9. R.E. Rice and C.K. Atkin, *Public Communication Campaigns*, Sage, 2012.
10. B. Stone-Gross et al., "The Underground Economy of Fake Antivirus Software," *Economics of Information Security and Privacy III*, 2013, Springer, pp. 55–78.
11. A. Adams and M.A. Sasse, "Users Are Not the Enemy," *Comm. ACM*, vol. 42, no. 12, 1999, pp. 40–46.
12. K.E. Vaniea, E. Rader, and R. Wash, "Betrayed by Updates: How Negative Experiences Affect Future Security," *Proc. SIGCHI Conf. Human Factors in Computing Systems (CHI 14)*, 2014, pp. 2671–2674.



IEEE-CS CHARLES BABBAGE AWARD

CALL FOR AWARD NOMINATIONS Deadline 1 October 2017

▶ ABOUT THE IEEE-CS CHARLES BABBAGE AWARD

Established in memory of Charles Babbage in recognition of significant contributions in the field of parallel computation. The candidate would have made an outstanding, innovative contribution or contributions to parallel computation. It is hoped, but not required, that the winner will have also contributed to the parallel computation community through teaching, mentoring, or community service.

▶ AWARD & PRESENTATION

A certificate and a \$1,000 honorarium presented to a single recipient. The winner will be invited to present a paper and/or presentation at the annual IEEE-CS International Parallel and Distributed Processing Symposium (IPDPS 2017).

NOMINATION SITE
awards.computer.org

AWARDS HOMEPAGE
www.computer.org/awards

CONTACT US
awards@computer.org

Robert W. Reeder is a senior user experience researcher at Google in New York. As a member of Google's Security & Privacy User Experience team, he conducts research at the intersection of human-computer interaction, security, and privacy. Reeder received a PhD in computer science from Carnegie Mellon University. Contact him at rreeder@google.com.

Iulia Ion is a software engineer at Google working on strong authentication and cloud security. She received a PhD in computer science with a thesis on usable security from ETH Zurich. Contact her at iuliaion@google.com.

Sunny Consolvo leads Google's Security & Privacy User Experience team, which focuses on usable privacy and security. Consolvo received a PhD in information science from the University of Washington. She's a member of the *IEEE Pervasive Computing and Proceedings of the ACM on Interactive, Mobile, Wearable, and Ubiquitous Technologies (IMWUT)* editorial boards. Contact her at sconsolvo@google.com.

Assessing Mission Impact of Cyberattacks: Toward a Model-Driven Paradigm

Alexander Kott | US Army Research Laboratory

Jackson Ludwig | The MITRE Corporation

Mona Lange | University of Lübeck

Mission impact assessments (MIAs) seek to assist the integration of business or military operations with cyberdefense, bridging the cognitive gap between operational decision makers and cyberdefenders. There has been increased interest in approaches to MIA that involve the construction and simulation of models of the mission, systems, and attack scenarios to understand an attack's impact.

A business mission's success depends on the communications and information systems (CISs) that support the mission. Cyberattacks on CISs can degrade or disrupt the performance and completion of the associated mission capability. There's a need for technology and procedures to characterize a cyberattack's impact on the mission; the term *mission impact assessment* (MIA) refers to this characterization.

A key objective of MIA is to assist the integration of business or military operations with cyberdefense, particularly in bridging the cognitive gap between operational decision makers and cyberdefenders.¹ In other words, MIA supports the ability to determine how an attack on the CIS infrastructure translates into consequences expressed in business and operational terms and thereby helps decision makers translate operational priorities into cyberdefense priorities. Given a cyberthreat and an attack with certain characteristics, cyber MIA should identify the space of impact scenarios: the CIS assets that would be disrupted, the chain of dependencies through which the CIS disruptions would propagate to business functions, and the resulting degradation in quantity and quality of the outputs of the affected business process.

MIA can be considered as a subfield in the much broader and far more mature field of risk management, particularly in its subprocess called *risk identification*.² This subprocess involves identifying business processes, functions, and supporting assets; threats, including strategically thinking and adaptive threats; vulnerabilities that could be exploited by threats; security controls or measures; and—especially relevant to MIA—technical and business consequences or impacts. In complex, multiorganizational cyber or cyber-physical systems, such impacts can be multifaceted, distributed in time and space, propagated through poorly understood dependencies, difficult to visualize and anticipate, and even counterintuitive. Effective computational approaches to automating or supporting MIA are lacking; this has encouraged the cybersecurity community to pay increasing attention to cyber MIA as a distinct problem.

Although attracting a distinct and growing body of research,³ cyber MIA remains a nascent field. For example, Alexander Motzek and Ralf Möller characterize mission modeling and MIA as an emerging field of research, provide a review of the relevant literature, and argue that many current approaches to MIA typically employ score-based algorithms leading to spurious results.⁴

The MIA research community has had a growing interest in a simulation model-driven paradigm. This requires the creation and validation of mechanisms of modeling the organization whose mission is subject to assessment, the mission (or missions) itself, and the cyber-vulnerable systems that support the mission. These models are then used to simulate or otherwise portray cyberattacks to understand their impacts.

In this article, we illustrate this trend with two specific examples and discuss related efforts. The two examples cover a broad range of business domains and research communities: one relates to the domain of civilian electric power distribution, the other to a military planning enterprise; one is the work of EU researchers, the other of the US. We point out the potential value of such approaches as well as the fact that sufficient evidence of value and feasibility is still lacking.

An Example of a Model: Impact of Cyberattacks on Power Grids

On an operational level, an electrical grid is a network of power providers and consumers connected by transmission and distribution lines with the mission of delivering electricity from suppliers to consumers. For monitoring and control purposes, they're connected to CISs. As recently as the 1990s, many power grid networks were isolated, stand-alone systems, and the day-to-day functioning of an electrical power grid mainly depended on the correct functioning of physical devices such as transmission and distribution lines, generators, and transformers. However, modern power grids and CIS infrastructures are closely coupled. Previously isolated power grids are increasingly integrated with CISs at power utilities, including public infrastructures, to increase business efficiency and effectiveness and to reduce operational costs. This has led to modern power grids becoming large networks consisting of thousands of network devices and applications. An application's operability, performance, or reliability might depend on multiple network services spanning multiple network devices and subnetworks of an infrastructure.

Both the US Department of Homeland Security (DHS) and the US Department of Energy reported an increase in the frequency and sophistication of cyberattacks on electricity systems between 2010 and 2015.⁵ A growing number of host and network intrusion detection systems and firewalls are deployed in electricity systems, leading to a high number of detected low-level events. Managing these low-level events and assessing their potential operational impact is critically important.⁶ A deeper understanding of potential impacts resulting from a successful cyberattack is required, especially for the development of trustworthy smart grids.

This was one of the goals of the Panoptesec project (2013 to 2016) funded by the European Commission's Seventh Framework Program for Research (FP7). The resulting Panoptesec prototype demonstrated a continuous monitoring and response capability to detect, prevent, manage, and react to cyberincidents in real time. For the purposes of this article, given a suspected attack, Panoptesec evaluates its operational impact, that is, performs MIA, among other capabilities.⁷

Panoptesec is one of several recent, related projects such as the Critical Infrastructure Security Analysis (Crisalis) project (www.crisalis-project.eu), funded from 2012 to 2015. Crisalis focused on three themes: securing systems, detecting intrusions, and postmortem analysis of successful intrusions. The bulk of its results relates to intrusion detection in Supervisory Control and Data Acquisition (SCADA) and industrial control systems (ICSs), as opposed to MIA. Although Crisalis research didn't extend into characterizing explicit impacts on business functions and processes, it did touch on interests of MIA where it explored approaches to detecting attacks against ICS devices by observing changes in the industrial process variables.⁸

Also funded by the FP7, for years 2014 to 2016, the Hybrid Risk Management for Utility Networks (HyRiM; hyrim.net) project aimed to identify and evaluate "hybrid risk metrics" for assessing and categorizing security risks in interconnected networks: the utility network physical infrastructure, consisting of, for instance, gas, water pipes or power lines, and the utility's control network including SCADA networks and business and information systems. The project provides risk assessment tools based on a sound and well-understood mathematical foundation. The bulk of the HyRiM research has concentrated on formal mathematical, including game-theoretic, risk models.

Perhaps the most direct ancestor of Panoptesec was the Vital Infrastructure, Networks, Information and Control Systems Management project (cordis.europa.eu/project/rcn/88625_en.html) financed by the EU from 2008 to 2011. Its key objective was to investigate SCADA system vulnerabilities and the cost of cyberattacks on society, focusing on systems for transmission and distribution of electric power. It used a model-based approach to investigate SCADA system vulnerability. Models were defined for the SCADA system, the electrical process, and the society that depends on the electricity supply. The models were linked to assess the propagation of consequences from a cyberattack all the way to the impact—expressed as monetary loss—for the society. The results laid a foundation for further exploration in the Panoptesec project.

As a case study, the Panoptesec consortium set up a testbed—an authentic replication of an Italian water

and energy distribution company's corporate enterprise systems and SCADA system. This allows for testing Panoptesec in an operational environment as well as for experimenting with cyberattackers who can penetrate computer systems, tamper with the accuracy of information, and shut down network services.

Figure 1 illustrates how an enterprise network and an operational network are linked in power grids. An enterprise network consisting of a primary control center is linked to an operational network consisting of two substations and an advanced metering infrastructure, represented by a smart home with a smart meter, heater, and a thermostat. The linkage of enterprise and operational networks is due to sensor measurements and control commands from power system operators in control rooms in an enterprise network being relayed over communication networks from or to a power grid's operational network. Clearly, there are multiple, complex dependencies among business functions and tasks, devices and applications, and network services.

An important finding of this project was the recognition that manual modeling of dependencies is prohibitively expensive in complex enterprises where responsibilities and knowledge are scattered across multiple departments or even third parties. Thus, a key output of the project was the development of an automated approach to learning network dependencies based on network traffic, and then deducing higher-level information about a network's mission based on network services and applications (see Figure 2).

Automated model development relies in part on network services dependency discovery, for which several approaches are known.⁹ Discovery of indirect dependencies among services is particularly challenging. In the Panoptesec project, communication patterns were analyzed and used to derive indirect dependencies based on "similar" temporal patterns of communications between two given pairs of services. Normalized cross-correlation was used heuristically to quantify the degree of similarity.⁷ This heuristic technique has been shown experimentally to outperform several alternative approaches in terms of recall and precision of the discovered indirect dependencies.^{9,10}

Dependencies identified via the automated method are used in part to construct a mission model, for instance, the one shown in Figure 2. Based on the mission model, the applications that are potentially impacted by an attack can be detected. Assuming the operational network is sufficiently modeled in the infrastructure model, MIA includes estimating the potential loss of electric power and whether it could lead to a blackout or brownout in the monitored infrastructure.

By comparing automatically derived mission models to results based on human input, automatically derived

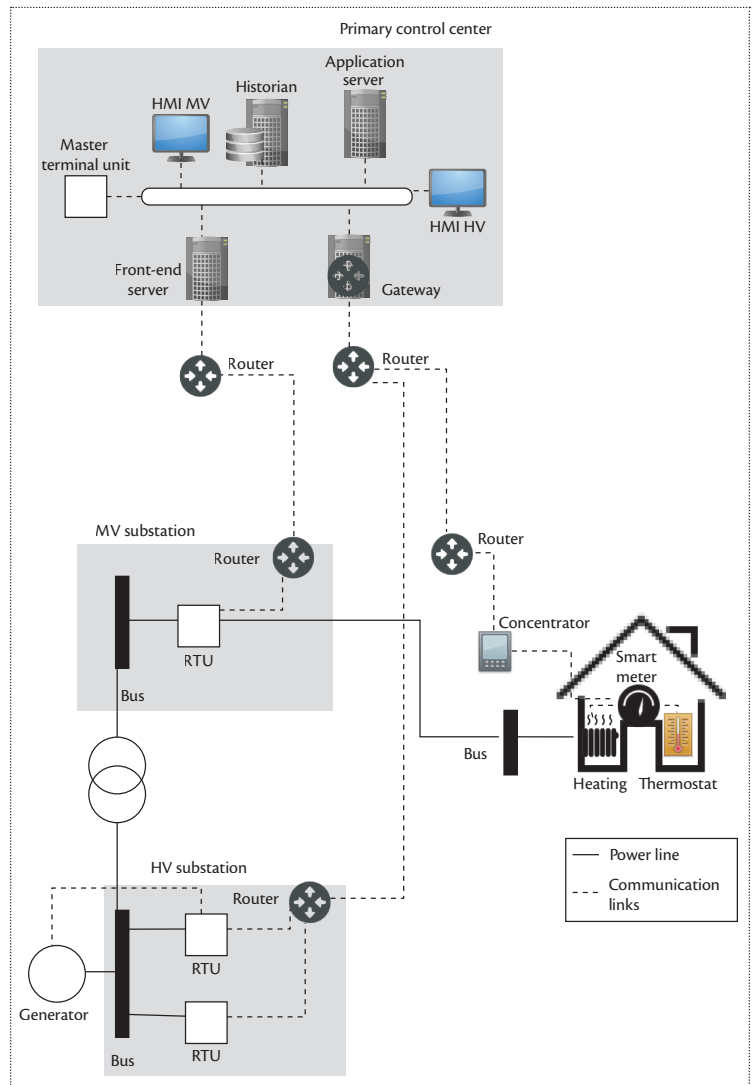


Figure 1. An enterprise network and an operational network linked in a power grid. Even the simplest schematics highlights the diversity and complexity of a modern electric grid's cyber and physical elements. There are multiple dependencies among business functions and tasks, devices and applications, and network services. HMI is human-machine interface, HV is high voltage, MV is medium voltage, and RTU is remote terminal unit.

mission models were found to provide a more detailed understanding of workflows in the network and discovered a surprisingly high number of hidden network dependencies that weren't identified by human operators. Unsurprisingly, network administrators found these automatically discovered, previously unknown network dependencies of great interest.

For example, consider an automatically generated diagram of significant relations between a subset of services in the Panoptesec testbed (see Figure 3). When Panoptesec researchers asked them, the operators explained that human-machine interfaces (HMIs)

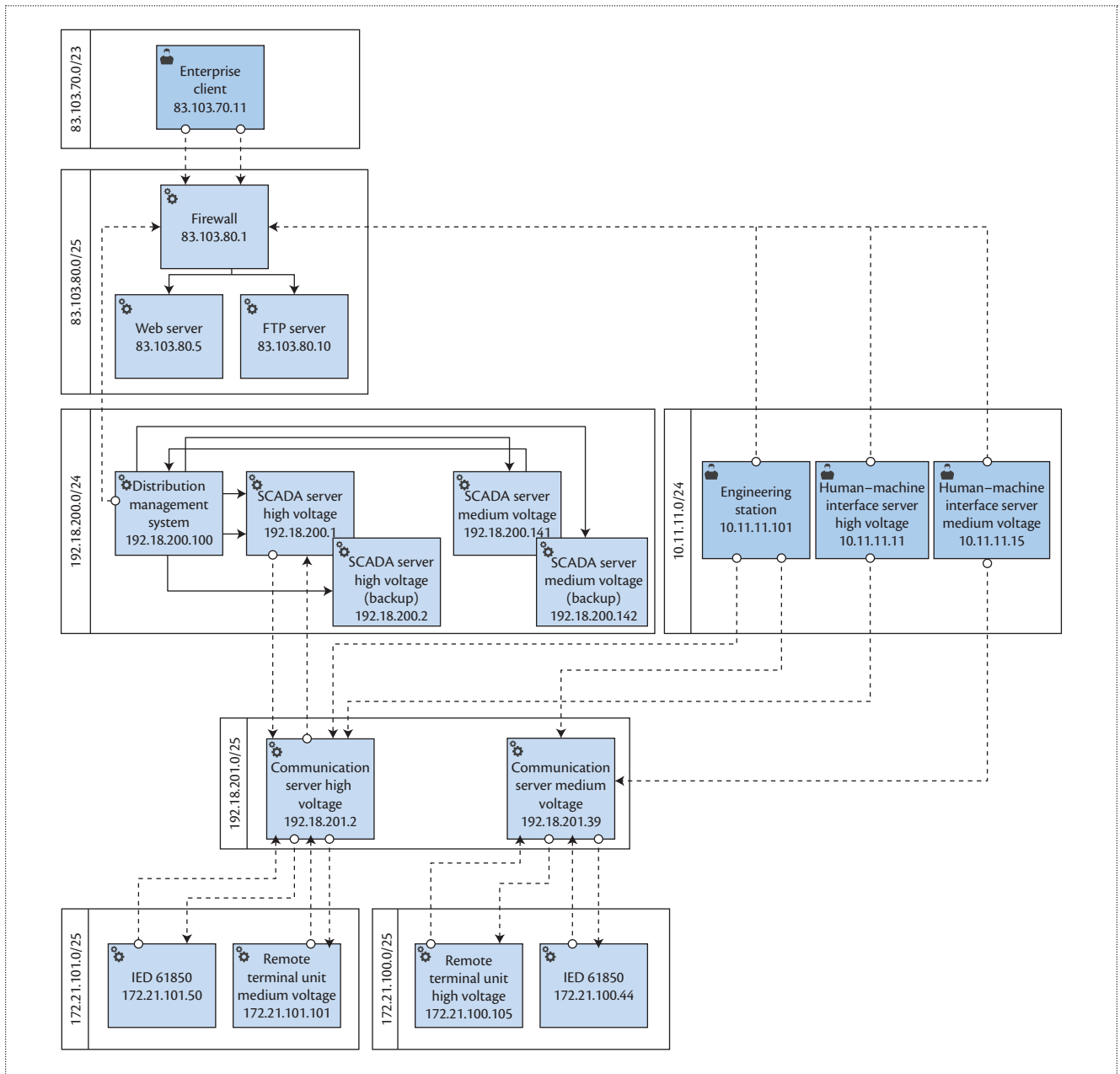


Figure 2. A high-level view of automatically derived mission models. Elongated rectangles represent subnetworks, smaller rectangles represent network devices, and a human silhouette marks client network devices. IED is intelligent electronic device.

(denoted by *msoz*) are used to communicate with the medium-voltage substations (denoted by *TTY*) through the communication servers (denoted by *mferp* and *muel*). Note that these and other names of various system components in Figure 3 are preserved here as found in the actual system.

However, automated analysis revealed a peculiar fact that was unknown to the operators: all HMIs (for example, *msoz22*, *msoz17*, and *msoz19*) were configured to contact *muel1* first, and then contact *muel2* if

muel1 was unavailable. At the same time, *muel1* was configured to be *muel2*'s backup. Therefore, *muel1* normally rejects requests from HMIs, because *muel1* can check and determine that *muel2* is available. Then, after the rejection by *muel1*, the HMIs send requests to *muel2*.

In effect, automated analysis discovered that there was an unexpected, unnecessary—and potentially exploitable—heavy volume of communications between HMIs and *muel1*, even though all

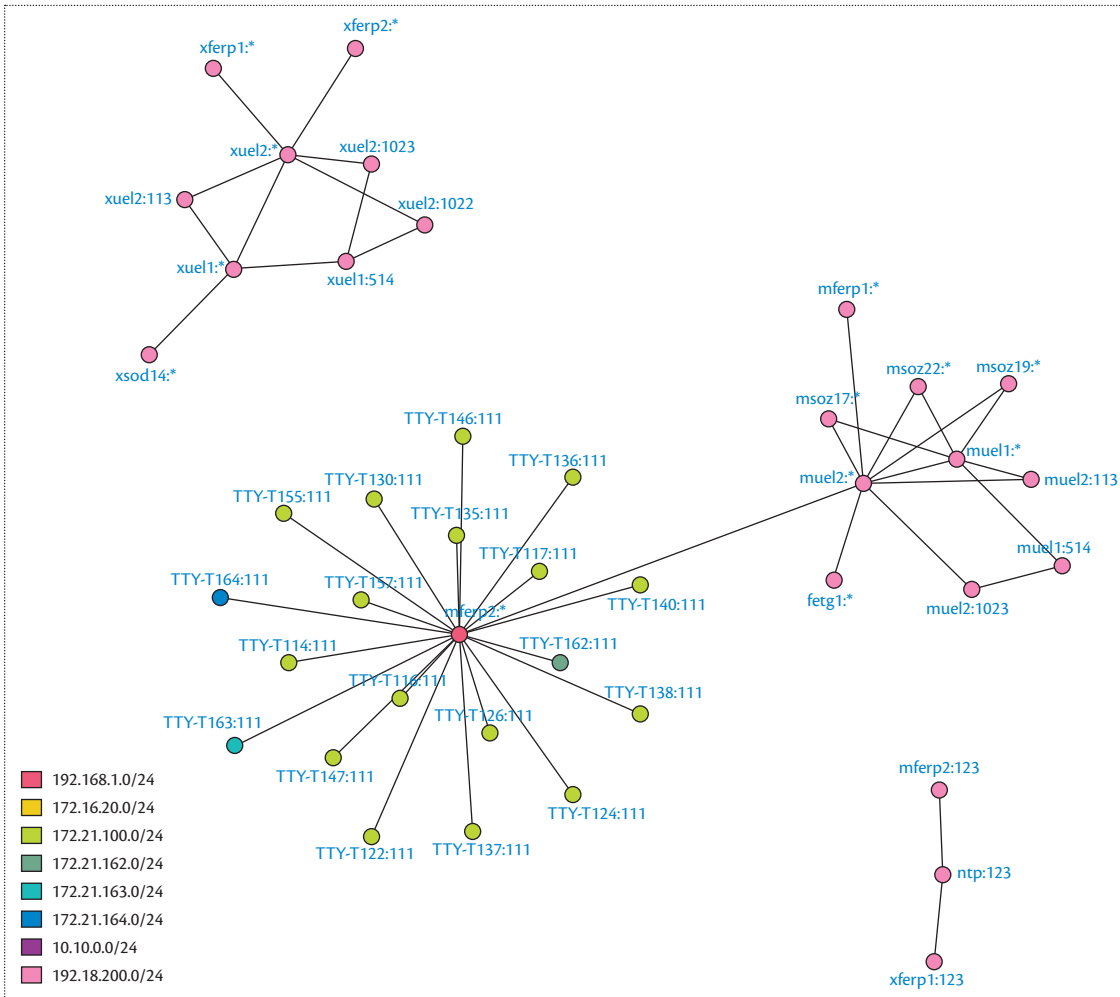


Figure 3. Human–machine interfaces (denoted by msoz) communicate with medium-voltage substations (denoted by TTY) through the communication servers (denoted by mferp and muel). Dependencies between services were identified automatically, revealing unknown dependencies due to misconfigurations.

communication between HMIs and mferp2 eventually occurred through muel2. After automated analysis discovered this fact, network operators adjusted the configurations accordingly. As a part of our research project, we developed a prototype tool for mining network traffic to derive a mission model and are releasing it under an open source license.⁷

To be sure, not everyone agrees that an automatically derived network traffic model is adequate for constructing a comprehensive mission model for MIA. For example, a group of researchers specifically evaluated several tools for automated dependency discovery and found them inadequate for the task; they proceeded to build the models manually, relying on the input of subject matter experts (SMEs). Furthermore, it should be noted that the Panoptesec models have never been fully validated, and the business value of the modeling and

simulation using such models has yet to be rigorously confirmed.

Another Example of a Model: Impact of Cyberattacks on a Military Air Operations Center

In 2015, the US Department of Defense funded a team of research organizations to develop a prototype to explore the feasibility of modeling and simulating concurrently the operational and cyberdomains, and translate the impact of cyberevents into quantifiable impacts on an operational mission’s execution. The outcome of this research effort was a prototype called Analyzing Mission Impacts of Cyber Actions (AMICA).

Understanding mission impact due to cyberattacks requires bringing together layers of information from numerous sources. At the lower layers, network

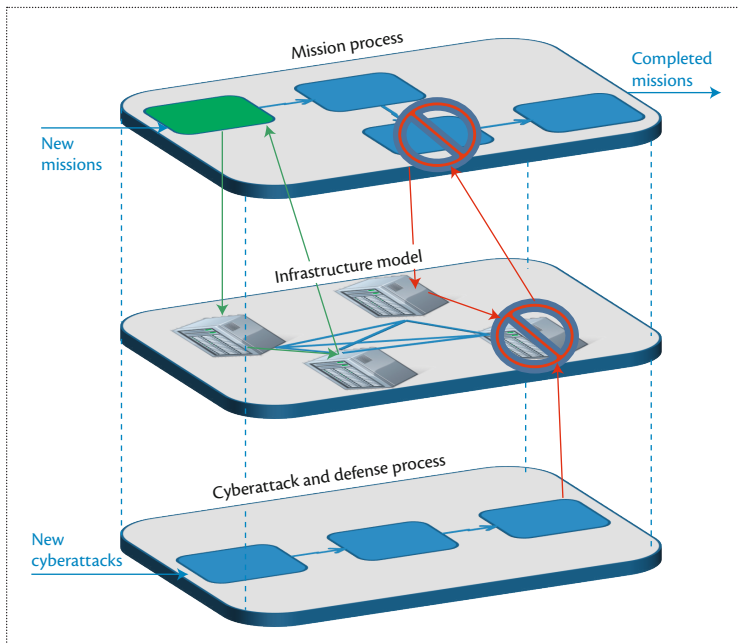


Figure 4. The Analyzing Mission Impacts of Cyber Actions (AMICA) model includes a mission process model, a cyberadversary process model, a cyberdefender process model, and a communications and information system infrastructure model.

topology, firewall policies, intrusion detection systems, system configurations, vulnerabilities, and so on, all play a part. Similarly, network devices and applications also need to be mapped to mission requirements. Because missions are highly dynamic, key network devices and applications likewise become dynamic. To address this, time-dependent models of mission flow and cyberactions (attack and defense) are necessary. AMICA supports exploration and experimentation of the mission impacts of cyberattacks through a flexible, extensible, modular, multilayer modeling system for quantitative assessment of operational impacts of cyberattacks on mission performance.¹¹

As a case study, AMICA was used to examine potential cyber impacts on an Air and Space Operations Center (AOC). At the risk of oversimplification, an AOC is responsible for developing the daily mission priorities and flight schedules for all the aircraft involved in a military campaign.¹² To conduct its mission, an AOC requires a large team of people working around the clock performing a planning and decision-making process, with a deep reliance on CISs.

AMICA consists of four main components: a mission process model, a cyberadversary process model, a cyberdefender process model, and a CIS infrastructure model (see Figure 4). The mission, cyberadversary, and cyberdefender are modeled in terms of their respective business (operational) processes using the Business

Process Model and Notation standard (www.bpmn.org). The CIS infrastructure is modeled through the use of both directed and topological graphs. Behavioral and temporal aspects of the AOC processes (workload, workflow, timing constraints, required resources, decisions, and so on) are implemented through executable process models and stochastic discrete event simulation. Structural and functional aspects related to the AOC infrastructure (environmental constraints, mission and system dependencies, vulnerabilities, and so on) are maintained through databases of graph models. Each component of AMICA is decoupled from the rest to provide modularity and independence, and interacts via shared interfaces with the CIS infrastructure model. This allows inputs at both the operational and cyber layers to influence the CIS layer's behavior and produce a combined effect on mission performance.

AMICA models the progress of each aircraft flight through the AOC's planning process, dependent on the state of the cyberinfrastructure. Cyberattacks themselves are modeled as stochastic steps within the attacker's workflow (modeled using the cyber kill chain¹³) and follow a pattern of gaining access to the network, lateral movement, and exploitation of the target device. Gaining access is done via spearphishing, where end-user nodes have a probability of falling victim to this attack. Lateral movement through the network is based on scanning the network for vulnerable devices that the adversary can exploit. Once the target device has been reached, the attacker creates a confidentiality, integrity, or availability impact on that system. Availability attacks slow down or stop execution of impacted mission activities. Confidentiality and integrity attacks don't slow execution but might cause AOC personnel to perform rework (if they're aware of the attack) or allow corrupted data to appear on flight plans. The attack's duration depends on how quickly the defender detects the presence of an attack, performs forensics, finds all the machines that have been compromised, and completes the remediation process. The consequences of these cyberattacks are reported in terms of mission-level metrics. In the case of the AOC, the number of mission plans developed and the number of flights flown are typical metrics of interest.

Using simulation to explore cyberattacks' effects, over time, on a specific AOC mission scenario revealed results that ranged from no impact at all (because of low workload intensity and redundant systems), to mission plans being delayed by days (because personnel needed to revalidate data after discovering a breach), to unknowingly having an entire day's worth of flights modified by an adversary (because an attack took place after the last consistency checks were made).

Randomly timed attacks against critical nodes in the CIS infrastructure, while disruptive, weren't devastating to the mission. On the other hand, attacks against key process steps conducted by attacking the same portion of CIS at the right moment during the process caused severe mission impacts. Thus, cyberdefenders need to assume that an advanced cyberadversary will likely target process vulnerabilities using cybervulnerabilities as a vector.

Several findings of this modeling effort were surprising and wouldn't have been possible to obtain without a comprehensive, systemwide modeling approach. In one example, two simulated attacks were run against systems supporting an important planning process, and the duration of each was varied (from hours to weeks) to determine the maximum tolerable attack duration. In one attack, system performance degraded by 50 percent (see Figure 5). This showed no impact because the users were able to keep pace with their workload even when using slow systems. In the second attack, the external network connection was disrupted, cutting off access to team members located at remote locations. In this case, the local users were nearly able to keep up with the workload, and only for outages exceeding three weeks did the reduction in completed flight plans reach a significant level (defined as a 10 percent or greater reduction). Because outages of such a duration are unlikely, this combination of process, workforce, and systems can be viewed as tolerant of attacks.

In another example, attacks resulting in data modification were targeted at varying times against systems supporting a critical flight-planning process. Static dependency analysis would correctly show that attacks against these systems could severely impact operations. However, simulation results showed that only attacks that took place after the last consistency checks were made (approximately halfway into the day) impacted the mission, providing a discrete window of vulnerability for defenders to focus on each day. The impact of attacks—expressed as a fraction of plans with undetected malicious modifications—ranged from the minimum of 3 percent to 100 percent on the agility and proficiency of the attacker modeled.

Such comprehensive modeling doesn't come cheaply. The effort involved in applying these approaches with current techniques to a new mission area requires months of effort by SMEs. Once modeled, scenarios and simulation runs can be done quickly, but a rigorous analysis of the data might take a few weeks. Attack trees, dependency graphs, and vulnerability scan data can each provide partial solutions identifying where or how a cyberattack might take place. But none can quantify or place bounds on the mission impact because those approaches don't include time. Governments and large

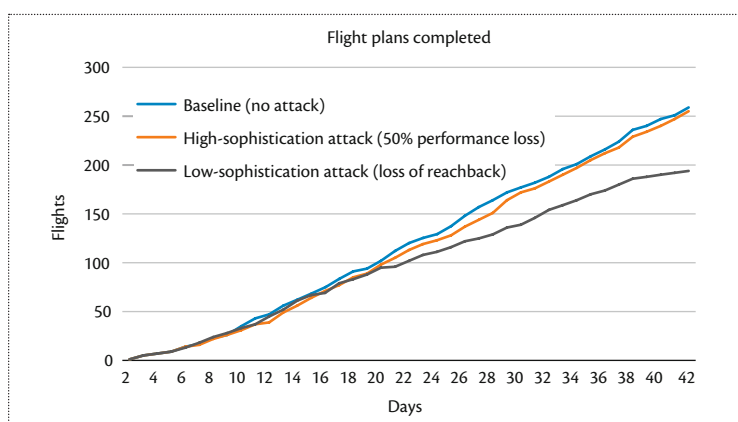


Figure 5. The model shows cyberattacks' unexpectedly limited impact if the users' workload is sufficiently low.

enterprises facing substantial cyberthreats might benefit from the additional detail that simulation can provide; however, this isn't an approach that is well-suited to casual users given the present state of technology.

The current prototype took three person-years to develop and contains only the features and datasets needed to support demonstration of the approach. A production-quality system would require further significant enhancements to data import, execution speed, scenario design, and user interfaces to make the software suitable for nondevelopers. The majority of the recurring effort in applying this approach is in collecting the necessary data about systems and processes, and the pertinent dependencies.

As another example of a recent experience, developing cyberdependency graphs for a large military organization with users, missions, systems, and networks spanning multiple locations took a team of developers more than seven person-years to complete, with further ongoing effort required to maintain the dependency graphs as changes are made to the infrastructure over time. A manual, SME-based mapping approach had to be used because an AMICA-related development team found that automated tools were insufficiently reliable in finding true dependency relationships—contrary to Panoptesec's experience. The automated tools showed poor recall and precision—that is, too many missed dependencies and too many false dependencies.

MIA Problem Formulation

Having considered two illustrative examples, we now discuss major remaining challenges.

To begin, appropriate formulation of a problem is the key to its successful solution. What constitutes a successful solution depends in turn on the solution's users. For example, decision makers need decision support at an appropriately abstracted level; they're much less

interested in technical details. For these reasons, future MIA techniques might specifically focus on supporting the cybersecurity decision-making process, particularly on tools that teach, train, and support decision makers.

Determining the correct users, however, depends on knowing where MIA belongs in the broader scheme of things. One way is to consider MIA as a part of the big control loop that strives to keep the controlled “plant”—the mission—within the prescribed space of secure states.¹⁴ This controller’s output is a set of corrective actions designed to keep the plant in the secure state.

In this formulation, MIA is the component of the control system that measures how much the plant has deviated or will deviate from the desired state. Once we say “how much,” a formal quantification of a utility function is needed. Some current approaches to MIA are based on heuristic scoring. To oversimplify, the assessor sums up the “impact points” and declares that the total impact on the mission is, let’s say, 73 points. Similarly, saying that “the mission impact is 70 percent failure” is very difficult to interpret. For example, even with 70 percent of the mission failing (whatever that means), the operator might still be able to reach a key goal.

One way to express quantitative MIA output would be to measure mission impact as a reduction in tangible system attributes, such as the network bandwidth, delay, or power use. Yet another approach would be to quantify the distance from the achievable states to desired states, for instance, via the cost of the corrective actions that would bring the plant to the desired, secure state. All this suggests that a formal language—a formal mathematics of mission security—would be highly desirable to give MIA a solid quantitative foundation.

Appropriate formulation of the MIA problem also requires choosing the right level of abstraction. When formulated and solved at a very abstract level, the solution might not give adequate insights into what actions—often very specific and detailed—must be considered. On the other hand, when formulated at a very detailed level, the problem demands a very intricate model that is far too expensive to construct. Arguably, a shift must occur from the enterprise-scale problem to a more meaningful tactical scale. One argument for more detailed formulations is that seemingly small attacks on mission activities can have large effects, as confirmed by simulation studies.¹¹

In addition to the control-theoretic style of MIA problem formulation, we shouldn’t overlook the game-theoretic (or game-playing simulation) perspective. A related and appropriate style of problem formulation could be robust control with adversarial inputs. Because full information isn’t normally

available, the problem should be formulated as a partial information game.

Model Content

The fundamental components of a model required for MIA include the organization models, its business processes (often decomposed into functions and tasks), the missions executed through the business processes, and the CISs that support the missions. Relations, influences, and dependencies—quantitatively characterized—among these entities and their subentities need to be modeled. Even the physical environment of a mission might need to be modeled as well as the sensors and actuators that sense and affect the environment, because they also can be subjects of cyber- or deception attacks.

Because the MIA problem is fundamentally adversarial in nature, we need a comprehensive model for adversary characterization and behavior understanding and prediction; the model should also include environment, attacks, and target properties, including modeling of these three elements and the relations and interactions among them.

In addition to describing the problem’s structure, models must capture its dynamics. There are several very different meanings of dynamics in MIA models. First, the structure itself changes rapidly. For example, the servers supporting a mission might be taken down for maintenance and then brought up online again, or reassigned to another mission. The model would need to be updated continually to reflect such changes. Second, when a cyberattack impacts a mission, the defenders and operators of the mission and supporting systems often show remarkable ability to work around the established process, that is, to redesign the business process rapidly and radically. Third, even in a very static business structure, actions are dynamic—they start, proceed, and stop in time. This dynamic must also be captured in a model. Fourth, the model’s characteristics of components and relations might change depending on the context. For example, systems’ criticality changes during different missions sharing the same systems.

Models of Adversary

Mission impact must be considered in the context of what impact the adversary desires. If we know or can estimate the adversary’s intents, motivations, and anticipations, the adversary’s impact or intended impact on our missions would be easier to assess. For example, in the AMICA system, the attacker model (agility and skills level) was shown to strongly influence the mission impact. It should be noted that here we consider the adversary rather abstractly; in particular, we don’t assume that the adversary can be modeled as an

individual human or a collection of individual humans. We discuss this perspective a bit later.

A model for adversary characterization, behavior understanding, and prediction should be sufficiently comprehensive. In particular, the model should include properties of the environment in which the cyberconflict occurs, the properties of the attacks and targets available to the adversary, and relations and interactions among all such elements—all this in addition to the adversary's properties. Naturally, the adversary's properties and characteristics are often unknown or uncertain. Modeling tools should allow representing such uncertainties.

A powerful determinant of adversaries' behavior is their expectations of our response. Thus, it's important to understand the role of *deterrence*—the measures we can take to prevent hostile actions by an adversary—in a cyberconflict. An adversary model should help answer questions like: What do the adversaries want to do? And what they expect us to do?

Models of Other Entities

When the adversary is an individual human, or a group of individuals that we find appropriate to model individually, we should consider techniques of cognitive modeling of individual human minds. Such models can help predict how cyberattackers formulate their goals and thereby tell us about the intended or actual mission impact of the adversaries' actions. Cognitive modeling tools like ACT-R (Adaptive Control of Thought—Rational) are beginning to be used for modeling cyberattackers' behaviors.¹⁵ However, this research area is rather immature.

Defender models shouldn't be overlooked either. To assess the likely mission impact, we need to know how a human cyberdefender reacts to cyberattacks. Errors committed by defenders determine the extent of mission impact. A defender might fail to recognize a threat and to take appropriate actions (or take the wrong action based on imperfect information), thereby enabling a greater impact on the mission. A defender might fall victim to an attacker's deception¹⁶ or fail to undertake a suitable workaround when a mission is impacted. A defender might also misinterpret the impact when it occurs. All this is highly relevant to the MIA problem.

Whether we model an attacker or a defender, the model needs to be rich enough to reflect "irrational" aspects of human cognition, such as cognitive biases. These aspects are particularly important in the high-pressure, high-tempo, nonintuitive world of cyberoperations. The impact of dynamic learning must be considered to account for rapid evolution of knowledge in cyberconflicts. Game-theoretic approaches should be included to account for the highly adversarial nature

of cyberoperations. Because both the attacker and the defender often operate with very limited awareness of one another's actions, situational awareness of both should be modeled. The importance of situational awareness in achieving impact on the opponent's mission can't be overlooked.

However, in many cases, both the defender and the attacker are best modeled not as individual human cognitive actors, but rather as organizations. Organizational modeling is studied by a community of researchers in the political science field that is distinct from the community of cognitive modelers. It would be worth exploring how that community might help solving the MIA problem.

Model Construction

The current practice of constructing models for MIA is almost entirely manual in nature. As such, model construction is very time consuming, expensive, and difficult to document, inspect, and validate. Maintaining such models—also manual—is likewise expensive. Quantitative characterization of dependencies between, for example, business functions and supporting technical assets, is largely a matter of asking the presumed SMEs for a number, such as a conditional probability. However, this can be expensive, and the verity of numbers is doubtful.

Still, manual construction of models for MIA problems appears feasible, even if expensive. For example, both Panoptesec and AMICA are comprehensive MIA modeling and simulation systems with a relatively fully implemented business model. To a degree, both rely on manually crafted models (AMICA more so).

Some tools allow essentially manual yet computer-aided construction of business models. Widely available business process management tools fall into this category. Ideally, however, we would like to see the bulk of MIA models constructed automatically, perhaps by observing a business process and its cyberdefense operations, and automatically learning or inferring a model.

Business effectiveness of cyberdefense depends to a large degree on the business's ability to assess—systematically and quantitatively—the impact of cyberattacks on the mission. This MIA problem won't be solved by an ad hoc muddle of compliance checklists, forensic investigations, and expert opinions. As in most mature technical and management fields, the problem will require comprehensive models.

To be sure, the challenges of building such a model are formidable. They range from formulating a model around the right decision-making needs and at the right

level of detail, finding effective means of representing complex adversarial and human cognition aspects of the domain, and developing cost-effective approaches to constructing and validating the model. Although experience with recent research projects suggests feasibility and potential utility of developing such models, decisive evidence of their benefits awaits further research and practical deployments. ■

Acknowledgments

This article was inspired in part by the NATO-sponsored workshop Cyber Attack Detection, Forensics and Attribution for Assessment of Mission Impact, held in Istanbul, Turkey, in July 2015.

References

1. A. Kott, C. Wang, and R. Erbacher, eds., *Cyber Defense and Situational Awareness*, Springer, 2014.
2. *Information Technology—Security Techniques—Information Security Risk Management ISO/IEC FIDIS 27005*, ISO/IEC, 2008; www.iso.org/standard/56742.html.
3. S. Musman et al., “Evaluating the Impact of Cyber Attacks on Missions,” MITRE, 2010; pdfs.semanticscholar.org/70fa/a7b3afa6f3afad7e64b92391d6e968b201ca.pdf.
4. A. Motzek and R. Möller, “Context- and Bias-Free Probabilistic Mission,” *Computers and Security*, vol. 65, issue C, 2017, pp. 166–186.
5. “Quadrennial Technology Review,” US Dept. Energy, 2015; energy.gov/under-secretary-science-and-energy/quadrennial-technology-review-2015.
6. S. Sridhar, A. Hahn, and M. Govindarasu, “Cyber-Physical System Security for the Electric Power Grid,” *Proc. IEEE*, vol. 100, no. 1, 2012, pp. 210–224.
7. M. Lange et al., “Event Prioritization and Correlation Based on Pattern Mining Techniques,” *Proc. 14th Int’l Conf. Machine Learning and Applications*, 2015; doi:10.1109/ICMLA.2015.76.
8. A. Carcano et al., “A Multidimensional Critical State Analysis for Detecting Intrusions in SCADA Systems,” *IEEE Trans. Industrial Informatics*, vol. 7, no. 2, 2011, pp. 179–186.
9. X. Chen et al., “Automating Network Application Dependency Discovery: Experiences, Limitations, and New Solutions,” *Proc. USENIX Symp. Operating Systems Design and Implementation (OSDI 08)*, 2008, pp. 117–130.
10. M. Lange, F. Kuhr, and R. Möller, “Using a Deep Understanding of Network Activities for Network Vulnerability Assessment,” *Proc. 1st Int’l Workshop AI for Privacy and Security*, 2016, article 6.
11. S. Noel et al., “Analyzing Mission Impacts of Cyber Actions,” *Proc. NATO IST-128 Workshop Cyber Attack Detection, Forensics and Attribution for Assessment of Mission Impact*, 2015; www.arl.army.mil/arlreports/2015/ARL-SR-0349.pdf.
12. R. Thompson, “Realizing Operational Planning and Assessment in the Twenty-First-Century Operations Center,” *Air and Space Power J.*, vol. 27, no. 2, 2013, p. 107.
13. E. Hutchins, M. Cloppert, and R. Amin, *Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Kill Chains*, white paper, Lockheed Martin, 2010; www.lockheedmartin.com/content/dam/lockheed/data/corporate/documents/LM-White-Paper-Intel-Driven-Defense.pdf.
14. A. Kott, “Towards Fundamental Science of Cyber Security,” *Network Science and Cybersecurity*, Springer, 2014, pp. 1–13.
15. C. Gonzalez et al., “Cognition and Technology,” *Cyber Defense and Situational Awareness*, Springer, 2014, pp. 93–117.
16. A. Kott, *Information Warfare and Organizational Decision-Making*, Artech House, 2006.

Alexander Kott is the chief of the Network Science Division, Computational and Information Sciences Directorate, US Army Research Laboratory. His research interests include applied development in network science and science for cyberdefense. Kott received a PhD in mechanical engineering from the University of Pittsburgh. Contact him at alexander.kott1.civ@mail.mil.

Jackson Ludwig is a principal cybersecurity engineer with the MITRE Corporation. His research interests include cybersecurity and business process analysis. Ludwig received MSs in systems engineering and cybersecurity from George Mason University. He’s a member of IEEE. Contact him at ludwig@mitre.org.

Mona Lange is a research assistant at the University of Lübeck, where she’s employed as a scientist in the European Union Project Panoptesec and is pursuing her PhD. Her research interests include formal methods for ensuring security. Lange received an MS in computer science from Friedrich-Alexander University, Erlangen. Contact her at monalange@gmail.com.

myCS

Read your subscriptions through the myCS publications portal at <http://mycs.computer.org>

Take the CS Library wherever you go!



IEEE Computer Society magazines and Transactions are available to subscribers in the portable ePub format.

Just download the articles from the IEEE Computer Society Digital Library, and you can read them on any device that supports ePub, including:

- Adobe Digital Editions (PC, MAC)
- iBooks (iPad, iPhone, iPod touch)
- Nook (Nook, PC, MAC, Android, iPad, iPhone, iPod, other devices)
- EPUBReader (Firefox Add-on)
- Stanza (iPad, iPhone, iPod touch)
- ibis Reader (Online)
- Sony Reader Library (Sony Reader devices, PC, Mac)
- Aldiko (Android)
- Bluefire Reader (iPad, iPhone, iPod touch)
- Calibre (PC, MAC, Linux)
(Can convert EPUB to MOBI format for Kindle)

www.computer.org/epub



IEEE  computer society

Security and Privacy Experiences and Practices of Survivors of Intimate Partner Abuse

Tara Matthews, Kathleen O’Leary, Anna Turner, Manya Sleeper, Jill Palzkill Woelfer, and Martin Shelton | Google
 Cori Manthorne | CORA (Community Overcoming Relationship Abuse)
 Elizabeth F. Churchill and Sunny Consolvo | Google



and practices.³ Our study builds on prior research outlining different phases of IPA (for example, Lenore Walker’s three phases of abuse⁴ and Shirley Patton’s five phases of leaving⁵). We also draw on research focused on improving the usability of general online privacy and security technologies (for example, Lorrie Cranor and Simson Garfinkel’s work⁶). We believe that understanding the experiences of survivors of IPA can improve digital privacy and security for the general population.

Formative Study

Our study sought to answer the following research questions:

- How do survivors of IPA experience digital privacy and security?
- What are survivors’ motivations, practices, and challenges when protecting their privacy and security online and on their devices?

Survivors of intimate partner abuse (IPA) are people who’ve experienced emotional abuse, threats of physical or sexual violence, or actual physical or sexual violence from an intimate partner—typically a current or former spouse, spouse during the process of separating, or dating partner. Approximately one in four women and one in 10 men in the US have experienced negative impacts from sexual violence, physical violence, or stalking by an intimate partner;¹ approximately one in three women worldwide have experienced

physical or sexual violence by an intimate partner.² Because IPA affects so many people globally, the survivors and abusers vary in gender, culture, wealth, education, tech literacy, and other attributes.

Survivors of IPA would greatly benefit from a technology community that understands and continues to address their unique challenges. To help technology creators better support survivors of IPA, we share findings from a study aimed at understanding this population’s digital privacy and security experiences

To this end, we conducted one-hour semistructured interviews with 15 survivors of IPA (14 female, one male). All participants were of low socioeconomic status, receiving services from two US nonprofit agencies. We worked with agency staff to co-create a study plan. Agency staff recruited participants who were at least 18 years of age and had a digital privacy or security

concern, such as experiencing an account breach. For more information about our study’s methodology, please see “Stories from Survivors: Privacy and Security Practices When Coping with Intimate Partner Abuse.”³

Ethical Considerations

This research entailed important ethical considerations. Throughout the process, we referred to pre-existing literature and consulted with more than a dozen experts in domains including survivors of IPA, human subjects research, legal, ethics, security, privacy, and anonymization.

Participant well-being shaped our study design. Our agency collaborators recruited using our criteria so that participants wouldn’t need to communicate with us until they decided to participate. We conducted interviews at the agencies to help participants feel more comfortable. Our interviews focused on technology-related abuse, not general stories of abuse unrelated to technology. We also made aftercare arrangements to communicate with the agencies if anything problematic came up during interviews.

We anonymized the data we report following the guidance of multiple privacy experts and agency collaborators. Also, our findings focus on informing the technology community. We assessed prior literature and confirmed that our findings organized known abuser attacks and survivor practices^{7,8} into a framework with corresponding recommendations for technology designers. We describe additional ethical considerations in “Stories from Survivors: Privacy and Security Practices When Coping with Intimate Partner Abuse.”³

Three IPA Phases Affecting Technology Use

We observed three phases of IPA—physical control, escape, and life

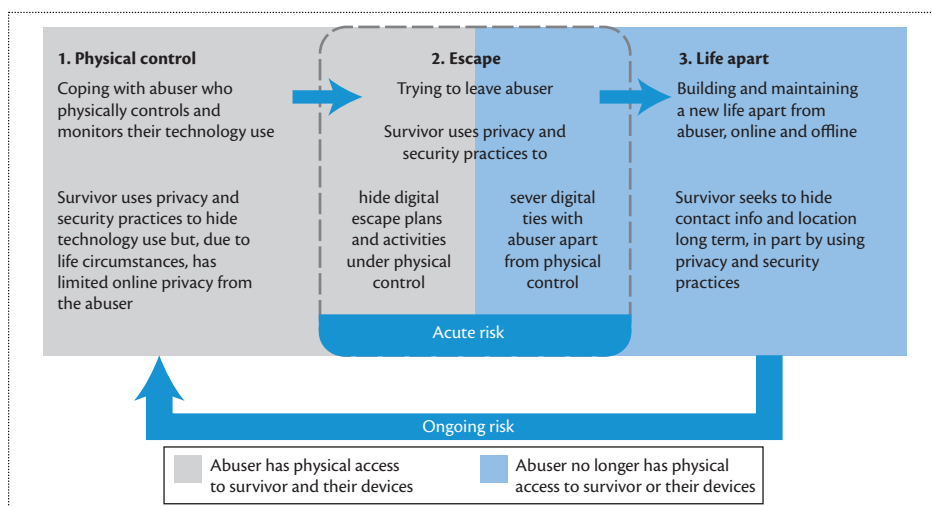


Figure 1. Three phases of intimate partner abuse that affected survivors’ technology use, focusing on privacy and security practices.

apart (see Figure 1)—that affected how survivors used technology, focusing on their digital privacy and security practices. This framework gives technology creators a lens through which to consider how survivors of IPA might experience or leverage new and existing technologies. Note that this framework describes how survivors’ experiences and use of technology are likely to change across the different phases.

Physical Control

He’s really controlling, and he doesn’t want me to even have anything online. . . . Like, he wants me to be alone and have nobody. So I could just call him whenever I need him, just so he’s the only one.
—Participant (P) 12

Participants first faced the physical control phase, during which their abuser had regular physical access to them and their devices. Abusers used this physical proximity to control and monitor participants’ devices and accounts. Although some participants found ways to use technology rather than avoid it completely, all described challenges maintaining autonomy and privacy

while using technology because of their abusive relationships. For multiple participants, the abuser’s physical control of their technology contributed to social isolation, device loss or damage, financial hardship, and psychological distress.

Survivor experiences. During the physical control phase, abusers violated participants’ digital privacy and security by

- physically controlling and monitoring their devices and accounts,
- destroying their devices, and
- installing spyware on their devices.

Abusers also hijacked participants’ accounts and harassed them online during the physical control phase. Such hijacking also occurred during the other two phases.

P3 explained how her abuser monitored her digital activity by forcing her to give him physical access to her phone. She said: “When we were together, he would always have my phone. Whoever would text me, he had to see who it was first. Or who was calling me, he had to check to make sure it wasn’t another guy.”

P4’s abuser had destroyed several of her phones. She told us “he would

just, like, stomp on it.” This isolated her from her social relations. In her words, “people have my old numbers ... there was no way for me to get a hold of other people.” P4 had to deal with being phoneless for some time due to the difficulty of purchasing a new phone in her situation.

P2 observed unusual behavior on her phone and, with help from an expert at a store, found spyware. She explained that the expert told her that “somebody put something in the phone ... [so that they] can see ... where you call, who you talk to, all the logs ...”

Participants experienced threats online in all three phases. For example, P4 described: “He would just talk about me with my name, my family’s name ... all this information, he would just put it on [the social network]. ... Because he was basically being a bully, as well, through [the] Internet, saying he was gonna kill me, kill my mom, kill my dad, kill my [sibling].”

Account hijacking was another abuser attack that participants experienced in all three phases. For example, P7’s abuser hijacked her email account and impersonated her. He also deleted emails about potential jobs for her. She told us: “He read personal emails and responded to personal emails in my voice. And he deleted job information.” She found it to be “rather personal and damaging.”

Survivor practices. To cope with their abusers’ physical control of their devices and accounts, participants reported

- limiting or avoiding use of devices and accounts their abusers could access,
- using alternate devices and accounts their abusers didn’t know about, and
- deleting material from their devices and accounts (such as messages and browsing histories).

For example, after P2 found spyware on her phone and laptop, she said: “I simply stopped using the laptop at home. And the phone. That’s why I went to the library to use the computer.”

Escape

I was trying to figure out a way to get out. And so I was moving stuff out of our house a little at a time while he was at work. ... I got that little prepaid phone and then I called from there. ... I was just in the middle of the street. —P11

During the escape phase, participants’ main goal was to leave and sever ties with their abuser. The escape phase overlapped with the other two phases, so it inherited the same abuser attacks and survivor practices as those phases. However, it added new privacy and security challenges due to the survivors’ life circumstances. The National Domestic Violence Hotline estimates that it takes an average of seven escape attempts to succeed.⁹ Research shows that abusers escalate their attempts to regain control over survivors during this time, resulting in an increased likelihood of violence and even death.⁷ Thus, in Figure 1, we mark escape as being an acute risk and depict the phases as a cycle.

Survivor practices: escape during physical control. During the physical control portion of the escape phase, participants focused on hiding their digital escape activities, for example, learning how to escape, setting up social support, and finding new housing and jobs. They used the same practices described for the physical control phase, but possibly more often because, as noted, abusers might escalate their efforts.

As an example of escape during physical control, P8 told us how she used an alternate account on her work computer: “I was trying to

look for elsewhere to live and trying to find resources out there and trying to apply to, you know, just housing and things like that. And I didn’t want it to go to where he would find it. So ... I’d go into [my separate email account] at work only, I didn’t want [that account] on my phone or anything.”

As another example of escape during physical control, P2 explained how she deleted her browsing history from her home computer to hide some of her search activities from her abuser and her child: “[I delete my browsing history because] my [child] sometimes [uses] the computer; I don’t want [my child] to know that I am like searching how to get a restraining order, ... how to kick my husband out of the house. How to help my [child] cope with separate parents, how to help your [child] in school with those kind of issues. ... But I don’t want [my child] to see what I am searching; [my child] will start asking questions and I am not ready.”

Survivor practices: escape during life apart. During the life apart portion of escape, participants needed to sever digital ties with their abusers. To do so, participants

- deactivated or abandoned accounts known to the abuser;
- destroyed, discarded, or wiped devices; and
- strengthened authentication.

After leaving, P3 deactivated her social media account in an effort to hide her new location. She suspected her abuser had located her through the account during a previous escape attempt, saying “he’d find so many ways to find out where I was.” The decision to deactivate an account often involved balancing digital privacy and security with access to social support, both of which were important during escape and

life apart. After deactivating her social media account, P3 risked reactivating it to contact her mother: “My mom didn’t have a phone back then. So I had to ... use the [social media account] to talk to her. So it was scary.”

P6, whose abuser had installed spyware on her phone, didn’t trust that a reset would completely fix the problem. So after leaving her abuser, she destroyed her phone, saying: “Bye-bye phone. SIM card through the shredder. ... The phone unit, painstakingly ran over by a car a couple of times. I mean, it’s in pieces.”

Several participants decided to keep their online accounts, but reported strengthening how they authenticated. P11 recalled: “[A software product] let me know when someone’s trying to hack into my account. Then I used the [two-factor authentication] method, and I changed the password. So that is so cool for me. It’s a couple times. I think the last time was my ex. You know he thought he could just check my email and see what I’m doing.”

Life Apart

I had given up my home, left my job, relocated to another county and not this one that we’re sitting in. My [children] had to go through this. ... I had spent a lot of money, lost a lot of money, and had gone through a lot of tech devices. —P6

During the life apart phase, participants described having to start over—often with a new home, job, schools for their children, devices, and accounts—while also dealing with the immediate and long-term risk of their abuser finding information about them. After severing digital ties as part of escape, participants had lifelong privacy work to do, ensuring that they, their children, and other people took great care when

sharing their personal information online.

Survivor practices. Participants exerted special care to protect their location (anywhere they or their family go) and contact information (new email addresses, phone numbers, online identities, and so on). They did this to prevent abusers from harassing them or reestablishing physical control. To protect their personal information, participants

- limited or avoided sharing information online,
- monitored and restricted their children’s online activities,
- strengthened the privacy and security settings for their online accounts, and
- severed ties with social relations they had in common with their abuser.

Several participants reported limiting or avoiding sharing information online. But this limited job opportunities for some, as described by P15, who was self-employed but could no longer advertise her services and thus had to change careers: “I have my [small business], but when I was actively working, so you have your email on [the advertisement]. And then you have your phone number. You [include] when you’re going to [be there]. ... They know right where to find you, and sometimes, you’re there by yourself. You’re just a sitting duck.”

An important challenge in staying hidden was that the abuser could use other people—such as the participant’s children, family, friends, and colleagues—to find the participant. This concern greatly complicated participants’ online privacy and security work, because it required them to enlist the cooperation of other people who might not fully understand or appreciate their situation. For example, P11 told us that she doesn’t allow her teenager

to post on social media. She said, “I just don’t want [my teenager] posting something out there that could be threatening to [him/her] or to our entire family, [he/she] doesn’t even realize it. Like if [he/she] puts ... where you go to school. ... That means [my abuser] could be sitting outside waiting in the car-pool lane, or in the morning when they get to school, there he is.”

Some participants decided to sever ties with social relations they shared with their abuser. For example, P5 said: “I’ve gotten rid of a lot of friends. ... They’re mutual friends [with the abuser]. ... People can flip-flop, play one side, or [talk] to me and then go give him information. I just don’t trust anybody.”

What’s Working and What Can the Tech Community Do Next?

Our study highlights ways in which technology is already working well for survivors across the three IPA phases that affect technology use. It also highlights opportunities for technology creators seeking to support survivors of IPA.

Using Controls to Delete or Hide Online Activities in High-Stress Situations

Deleting or hiding online activities such as messages and browsing was an important strategy for survivors in all phases, especially physical control. Fortunately, there are already ways survivors can do this; for example, many technologies allow users to have more than one account or device, access multiple accounts on a device, and delete content.

However, as noted, participants in our study reported occasionally making mistakes when deleting or clearing information, perhaps due to the high levels of stress and risk they were facing. Future work could make further improvements by studying such tools’ usability during high-stress, high-risk situations.

Educating Survivors about Security Features

Account hijacking was an issue our participants dealt with in all phases. For those who knew how to use them, security features and controls like two-factor authentication and unusual activity alerts were very empowering, particularly in the life apart phase. However, confidence using these features, especially in high-risk situations, was an issue. Future work could focus on providing instructional materials for survivors and their service providers about how to use privacy and security features and controls.

Managing Digital Evidence of Abuse

It might come as a surprise, but there was an upside to some of the harassing messages that participants received from their abusers—these messages were sometimes provided as evidence to law enforcement (for example, to help obtain restraining orders). Prior work has also shown that digital channels can provide an outlet for an abusers' desire to exert control, which might reduce their motivation to exert control in other ways.⁷

However, survivors tend to experience emotional trauma as a result of such harassment. Future work could explore solutions that capture digital evidence and provide abusers with an outlet while minimizing survivors' emotional trauma.

Maintaining Online Social Lives

Some participants chose to avoid technology to limit the information an abuser could find about them online; however, this also socially isolated them at a time when they needed support and access to resources such as housing and jobs. Several practices commonly used by our participants—avoiding technology, deactivating accounts, and destroying devices—added to

their social isolation. Future work could educate survivors about existing technologies and explore new technological solutions to help them maintain social ties without leaking important personal information.

Providing Options and Ambiguity

Survivors must deal with highly motivated attackers who have intimate knowledge of their lives. In the physical control phase, attackers also have physical access to the survivors' devices and accounts. There is no one-size-fits-all solution to this type of threat. To cope, survivors benefit from having multiple privacy and security options to deal with their highly individual situations. In addition to the features we discussed, survivors benefit from the ability to maintain ambiguity in their technology-mediated interactions with others.¹⁰ One way to do this is to give them the space to tell stories about those interactions. For example, survivors could explain that they missed a call from their abuser by claiming that they didn't hear the call (for example, by saying that their phone ringer was off or they left their phone behind). Another way to give survivors space is to provide granular controls such as the ability to delete specific content, ignore messages, or temporarily turn features off; such controls can also help users manage this type of ambiguity.

We shared results from a formative study on the digital privacy and security experiences and practices of survivors of IPA. Our aim is to help technology creators consider how new and existing technologies and features can be designed to help survivors of IPA as well as identify opportunities to continue to improve support for this user population. ■

Acknowledgments

This article summarizes a paper previously published in the proceedings of the 2017 SIGCHI Conference on Human Factors in Computing Systems.³ We thank Ali Lange, Allison Woodruff, Ben Petrosky, Clara Sherley-Appel, Elie Bursztein, Erin Simon, Heather Lipford, James Tarquin, Jessica Staddon, Keith Enright, Lawrence You, Lea Kissner, Matt Moore, Michael Falgoust, Michael Janosko, Parisa Tabriz, Patrick McDonald, and Thomas Roessler for their support, feedback, and ideas regarding this research and article. We thank our many colleagues at Google and our collaborators from the agencies for their fantastic support and ideas as we planned and conducted this research. We thank our reviewers and other members of the HCI and usable privacy and security communities who provided input along the way. Most importantly, we thank our participants for their willingness to share their stories, and for their courage.

References

1. M.J. Breiding et al., "Prevalence and Characteristics of Sexual Violence, Stalking, and Intimate Partner Violence Victimization—National Intimate Partner and Sexual Violence Survey, United States, 2011," *Am. J. Public Health*, vol. 105, no. 4, 2014, pp. e11–e12; doi.org/10.2105/AJPH.2015.302634.
2. *Global and Regional Estimates of Violence against Women: Prevalence and Health Effects of Intimate Partner Violence and Non-partner Sexual Violence*, executive summary, World Health Org., 2013; www.who.int/iris/handle/10665/85239.
3. T. Matthews et al., "Stories from Survivors: Privacy and Security Practices When Coping with Intimate Partner Abuse," *Proc. SIGCHI Conf. Human Factors in Computing Systems (CHI 17)*, 2017, pp. 2189–2201.
4. L.E. Walker, "Battered Women and Learned Helplessness," *Victimology*, vol. 2, nos. 3–4, 1977, pp. 525–534.
5. S. Patton, *Pathways: How Women Leave Violent Men*, Government of Tasmania, 2003; trove.nla.gov.au

/work/7886258?selectedversion=NBD25304195.

6. L.F. Cranor and S. Garfinkel, *Security and Usability: Designing Secure Systems that People Can Use*, O'Reilly Media, 2005.
7. C. Fraser et al., "The New Age of Stalking: Technological Implications for Stalking," *Juvenile and Family Court J.*, vol. 61, no. 4, 2010, pp. 39–55; doi.org/10.1111/j.1755-6988.2010.01051.x.
8. "Tech Safety App," Nat'l Network to End Domestic Violence, 2016; techsafetyapp.org.
9. "50 Obstacles to Leaving: 1–10," Nat'l Domestic Violence Hotline, 10 June 2013; www.thehotline.org/2013/06/50-obstacles-to-leaving-1-10.
10. P.M. Aoki and A. Woodruff, "Making Space for Stories: Ambiguity in the Design of Personal Communication

Systems," *Proc. SIGCHI Conf. Human Factors in Computing Systems (CHI 05)*, 2005, pp. 181–190; doi.org/10.1145/1054972.1054998.

Tara Matthews is a user experience researcher at Google. Contact her at taramatthews@google.com.

Kathleen O'Leary is a PhD student at the University of Washington. She performed this work while a user experience research intern at Google. Contact her at katieole@gmail.com.

Anna Turner is a user experience researcher at Google. Contact her at annaturn@google.com.

Manya Sleeper is a user experience researcher at Google. Contact her at manya@google.com.

Jill Palzkill Woelfer is a user experience researcher at Google. Contact her at jillwoelfer@google.com.

Martin Shelton is a user experience researcher at Google. Contact him at martinselton@google.com.

Cori Manthorne is a director of programs at CORA (Community Overcoming Relationship Abuse). Contact her at corim@corasupport.org.

Elizabeth F. Churchill is a director of user experience at Google. Contact her at churchill@acm.org.

Sunny Consolvo is a user experience researcher at Google. Contact her at sconsolvo@google.com.

IEEE  computer society

Read all your IEEE magazines and journals your **WAY** on

myCS

Introducing **myCS**, the digital magazine portal from IEEE Computer Society. Go beyond static, hard-to-read PDFs with an easily accessible, customizable, and adaptive experience.

There's No Additional Cost!

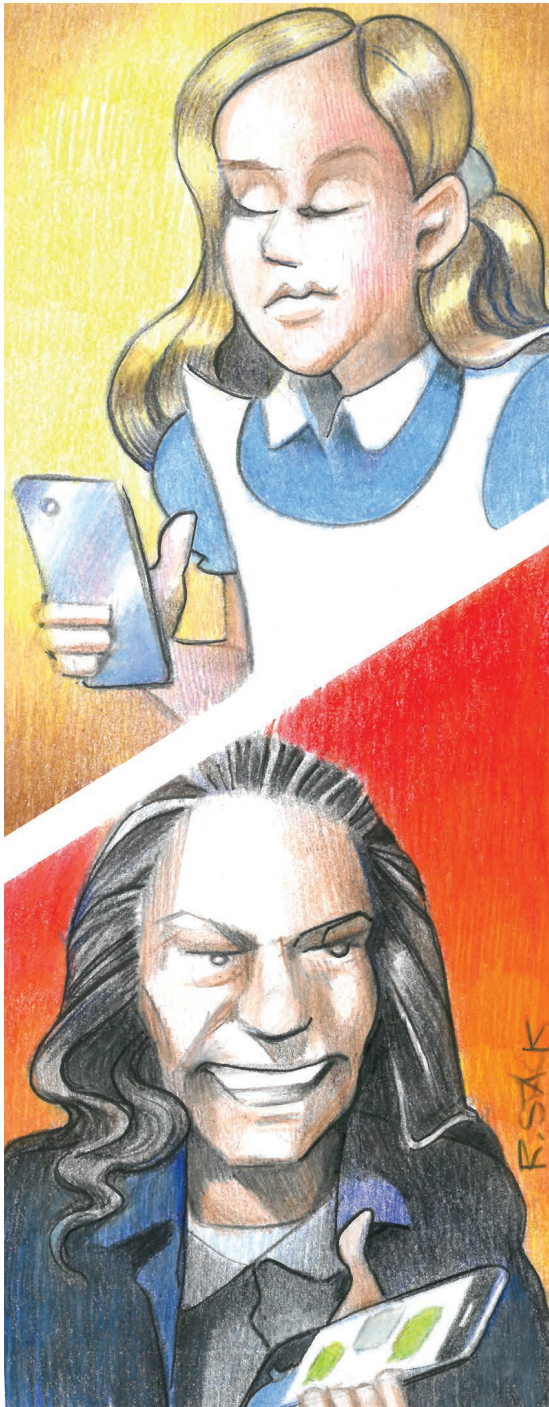
Now there's even more to love about your membership...



▶ LEARN MORE AT: mycs.computer.org

Disillusioning Alice and Bob

Rolf Oppliger | eSECURITY Technologies



In their seminal paper,¹ Ronald Rivest, Adi Shamir, and Len Adleman not only introduced the RSA public-key cryptosystem but also cast “Alice” and “Bob” as replacements for the A and B symbols used to refer to the participants of a cryptographic protocol. Since then, cryptographers and security professionals have cast additional characters to refer to protocol participants, such as Carl or Dave, or adversaries, such as Eve or Mallory. Originally viewed as a side product of the RSA paper, the notion of Alice and Bob prevailed and is now the de facto terminological standard and notation for arguing about cryptographic protocols—be it in informal descriptions or semiformal specifications.

In this column, I challenge this notation and argue against its further use. I think it’s more appropriate to use symbols such as A and B rather than human names like Alice, Bob, and the rest of the gang, because human names tend to oversimplify—and therefore obfuscate—the situation. When we say, “Alice sends a message to Bob,” we suggest that Alice and Bob

- are human,
- personally interact, and
- fully control the messages they send and receive.

In reality, however, the situation is more involved, and none of the above suggestions is true: neither Alice nor Bob is human, they don’t personally interact, and they don’t fully control the messages they exchange.

Figure 1 illustrates Alice and Bob communicating electronically. They both use a device (such as a computer system or a smartphone) that consists of multiple layers of hardware and software. More specifically, the device consists of hardware modules that run an OS, which hosts application software. For Alice to send a message to Bob, there must be messaging software available on either side of the communication channel. Alice interacts with this software on the sending device (the user interaction marked in black). The message is transport-encoded and sent over some networking facility empowered by some hardware and OS functionality (the network interaction marked in gray). The same is true on the recipient side: Bob isn’t personally receiving messages. Instead, he’s interacting with the messaging software installed on the receiving device and operated on some hardware and OS. The picture is highly fractal—it gets more involved as you zoom in on the details.

Keeping Figure 1 in mind, let’s revisit the sentence “Alice sends a message to Bob.” Note how it oversimplifies the situation. Instead of sending a message to Bob, Alice prepares the message using application software. She clicks a button to alert the software that the message is ready to be sent. This click is all Alice does; from that moment on, the message is transmitted by the appropriate software and hardware components of the sending device. Alice can hardly control these

operations, and she must trust that all components play by the rules and behave as specified. Obviously, many things can go wrong, and many components can misbehave and cheat in various ways. Having Alice (and Bob) follow the protocol is necessary, but not sufficient, to deliver the message from sender to recipient. Many other components are involved that must also follow the protocol rules.

Alice and Bob have been cast to explain cryptographic protocols. Using such a protocol, Alice doesn't typically send a message in the clear. Instead, she authenticates and/or encrypts it. But it's very likely *not* Alice who does the cryptographic computation but rather some hardware or software module that operates on her behalf (it can be a hardware security module such as a smartcard, or a cryptographic library that runs in software). The same is true for the cryptographic keys that control the cryptographic computation. Very likely, it's not Alice who provides these keys but a software module that either stores the keys or generates them on the fly by using an automated key exchange and management protocol. The bottom line is that cryptographic computations are never done by human users but by supporting modules implemented in hardware or software and specialized for these tasks (note that these modules aren't even illustrated in Figure 1).

The same line of argumentation that applies to a message's sender (Alice) and receiver (Bob) also applies to the adversary: it's almost never human users who eavesdrop and try to manipulate messages but rather highly specialized attack software. If adversaries try to mount a pass-the-hash attack, for example, the attack software extracts users' credentials from the local cache. If they try to mount a BEAST-like attack against the SSL/TLS protocols, the attack software is delivered

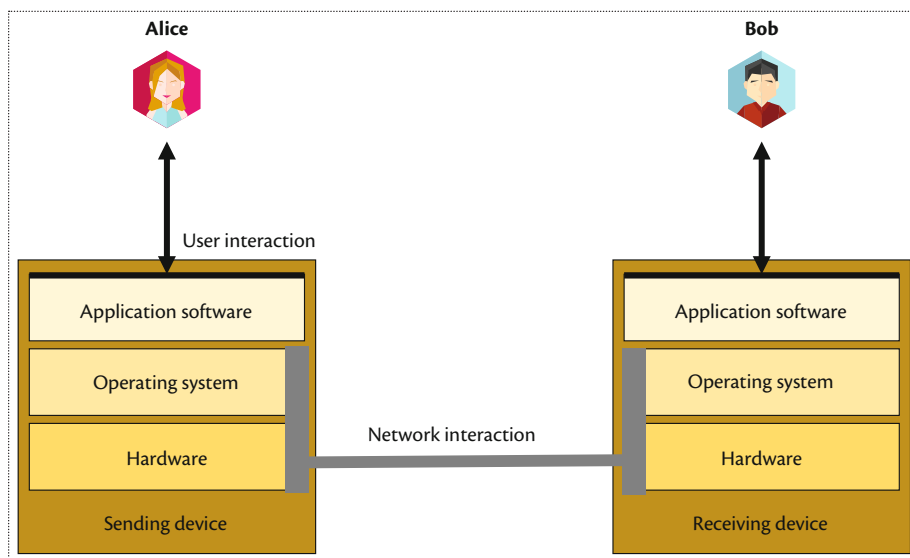


Figure 1. Alice and Bob communicate electronically, using devices with multiple layers of hardware and software.

as active content (for example, malicious JavaScript code) to launch a man-in-the-middle attack and choose ciphertexts that are sent to the server. Here, we're talking about thousands or even millions of ciphertexts that need to be compiled in a specific way and sent to the server in a reasonable amount of time. Adversaries must use highly specialized software to automate such an attack.

So, "Alice sends a message to Bob" sounds friendly but is illusive. Above all, it misses the point when it comes to a technical discussion, as is always the case in applied cryptography. Most of the components that must be in place and cooperate are inherently nonhuman. In fact, human users' roles in such protocols should be as small as possible—the more things users can do, the more likely something is to go wrong. Therefore, a rule of thumb in cryptographic protocol and system design is to make the user interface as small and intuitive as possible. This contradicts the role human names play in such protocols' description and specification.

The realm of remote Internet voting further clarifies my point. By clicking a button, Alice might think she's casting a vote for a particular candidate—but this isn't always true. If the software managing the voting process on the client side is flawed or somehow compromised, anything is possible and there's no real way for Alice to determine whether her vote was cast-as-intended and counted-as-cast. Many voting systems work that way and don't provide any guarantee. But there are cryptographic techniques that can empower Alice to verify her vote end to end (E2E). Technologies that provide E2E verifiability are going to be important in the future to mitigate the threats and respective risks in remote Internet voting.

So although it might seem a little pedantic (and most people working in the field likely appreciate the difference between a notation and reality), I still think it's more appropriate to use symbols like A and B instead of human

IEEE  computer society

Looking for the BEST Tech Job for You?

Come to the **Computer Society Jobs Board** to meet the best employers in the industry—Apple, Google, Intel, NSA, Cisco, US Army Research, Oracle, Juniper...

Take advantage of the special resources for job seekers—job alerts, career advice, webinars, templates, and resumes viewed by top employers.

www.computer.org/jobs

names like Alice and Bob. If you agree, then consider joining me in getting rid of the cast of characters and using symbols to describe and specify cryptographic protocols. A symbol is better suited to be associated with a multiple-component technical device than is a human name. Using such symbols might help bring discussions back into the realm of technology, where they really belong. ■

Reference

1. R.L. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Comm. ACM*, vol. 21, no. 2, 1978, pp. 120–126.

Rolf Oppliger is the founder and owner of eSECURITY Technologies. Contact him at rolf.oppliger@esecurity.ch.



myCS

Read your subscriptions through the myCS publications portal at <http://mycs.computer.org>

FinTechSec: Addressing the Security Challenges of Digital Financial Services

Patrick Traynor, Kevin Butler, and Jasmine Bowers | University of Florida
Bradley Reaves | North Carolina State University

You probably don't think about traditional banking very often. Many of us would be hard-pressed to say why we picked one financial institution over another—it could be one that partners with an alma mater, one used by friends and neighbors, or simply the one closest to home. Most important, most of us also have options and could easily take our business (and our money) down the street to another institution should we fail to receive the terms, service, and security we feel necessary to grow and protect our assets.

Now that we have you thinking about traditional banking, it's easy to begin enumerating the many things it enables: our employers electronically deposit our paychecks (and they become immediately available), we make payments using credit and debit cards (reducing the need to physically carry and protect cash), and we even have an array of protections against fraud. This infrastructure extends far beyond national borders and is now so pervasive in the developed world that travelers think nothing of withdrawing money at foreign ATMs. In short, traditional banking makes payments (and most of the challenges around it) largely frictionless to the consumer.

It would be easy to assume that everyone has access to traditional banking given its seeming ubiquity. Unfortunately, that assumption is simply wrong.

Billions around the world lack access to even the most basic



banking services for many reasons. Many simply lack physical access. Even more lack the ability to maintain the relatively high minimum balances required by traditional financial institutions. The practical impacts are significant. In the US alone, only 68 percent of homes were “fully banked” in 2015, meaning that the remaining 32 percent required the use of so-called “alternative financial services” including check cashing and payday loans.¹ Worldwide, some two billion people remain unbanked.

The lack of basic banking services makes tasks most of us take for granted, such as saving, electronic payment, and short-term loans,

essentially out of reach for huge portions of the population. Technology might provide a real path to so-called financial inclusion; however, as our research shows, security and privacy remain significant impediments to future progress in this space.

Our goal in this article is to discuss our experience in securing *mobile money*, a digital financial system that uses mobile phones to transfer currency without the need for a bank. Our efforts began in 2011 and have resulted in extensive collaboration with organizations including the US Department of State, the International Telecommunications Union (ITU), the GSM

Association (GSMA), the World Bank, and many individual providers and vendors. These transformative systems have already demonstrated the power to raise populations out of poverty, and we believe that they will soon be deeply intertwined with the traditional global financial infrastructure. This means that we have a chance to get security and privacy correct now instead of looking back with regret when the above systems are made manifest.

What Is Mobile Money?

In the mid-2000s, the Kenyan cellular provider Safaricom noticed an interesting trend. For some time, customers in its network could send minutes to their friends and families, and often did so to ensure that those with access to funds could talk to those without. However, a few enterprising customers began sending minutes in exchange for goods and services. This was no small innovation—at the time, the vast majority of Kenyans didn't have a bank account, and electronic payment was beyond most citizens' reach. In contrast, nearly eight out of 10 citizens had mobile phones. Seeing this tremendously unmet need for electronic payment being approximated with “top up” minutes, Safaricom launched M-Pesa in 2007 and allowed subscribers to send actual money to one another via SMS.

M-Pesa was an overnight success. Urban residents who would often travel long distances to physically transport money to their rural family members (often at the literal risk of highway robbery) could simply transfer those funds at the press of a few buttons. Moreover, M-Pesa overcame the problem of physical access by making virtually every vendor the equivalent of an ATM—capable of both depositing funds to and withdrawing funds from the network. Finally, M-Pesa charged

extremely low transaction rates, further enticing those unable to use traditional banking services to join.

M-Pesa now claims more than two-thirds of the Kenyan population as its customers. Moreover, this model has been copied and attempted widely across the globe (especially in the developing world). In 2016, there were more than half a billion mobile money accounts around the world, and the industry processed an estimated US\$22 billion.² These numbers continue to increase by staggering amounts each year.

What we've described here might sound somewhat familiar. After all, the past few years have seen the rise of peer-to-peer payment systems such as Apple Pay, Google Wallet, Samsung Pay, Venmo, and a handful of others. However, none of these are mobile money because they're all backed by the traditional banking infrastructure. That means that unless you acquire a credit or debit card, you really can't use these systems. Think of mobile money instead in the following way: rather than Bank of America or HSBC, AT&T or Orange now becomes your “bank,” and you deposit money or checks at your local gas station, corner market, or grocery store.

Mobile money is also not the same thing as cryptocurrency (for example, Bitcoin and Ethereum). Speaking very broadly, these two systems solve decidedly different problems. Whereas cryptocurrencies strive to create alternative money outside of centralized control, mobile money systems operate using traditional nation-state-backed fiat currency. While some researchers and start-ups have attempted to deploy cryptocurrencies in the context of mobile money, they haven't met much success. Moreover, mobile money is being used by a far greater number of people: M-Pesa alone reported 6 billion transactions in 2016,³ compared to Bitcoin's 184 million over

its entire lifetime (blockchain.info/charts/n-transactions-total). Given this number of transactions, we believe that those who care about cryptocurrencies should also understand mobile money.

What Is the State of Security?

Most first-generation mobile money systems were built on widely deployed 2G GSM cellular networks. These services relied on either SMS or Unstructured Supplementary Service Data (USSD) channels for communication. These channels are ideal from the perspective of rapid deployment in that they're nearly omnipresent. However, they're problematic from the perspective of security. First, 2G networks generally rely on cipher suites that are known to be weak. Specifically, the A5/1 and A5/2 algorithms protecting the wireless portion of GSM networks can both be cracked with relatively little effort by an adversary. Although A5/1, the stronger of the two ciphers, was believed to provide significantly improved protection, software-defined radio systems capable of cracking this cipher in real time are now available in backpack-sized setups. That means these first-generation services are vulnerable. To make matters worse, many providers instead rely on the A5/0 (that is, no encryption) standard, removing the already low barrier to attack.

Second, even if providers were to universally upgrade their over-the-air cipher suite to A5/3 (a stronger cipher also known as KASUMI, with known theoretical weaknesses but no practical attacks at this time), encryption protecting data in the SMS and USSD channels ends at the base station. That means that in the core network (and potentially over wireless backhauled used to connect remote towers to that core network), an attacker can easily observe and modify transactions

without detection. Moreover, because authentication in GSM networks is unidirectional (that is, device to network, but not network to device), an adversary could easily deploy a so-called “rogue base station” in a busy area and force all connections to pass unencrypted through it.

The most discussed solution in this space has been the SIM Application Toolkit (also known as SIM Toolkit). SIM Toolkit lets providers develop applications directly on SIM cards, thereby overcoming the need to build applications for a massive set of feature phone platforms. Many have proposed adding application-layer encryption to mobile money via SIM Toolkit, but these efforts have largely failed in practice. Providers privately express frustration in ensuring the correct operation of such a solution. Moreover, there’s great difficulty in replacing the massively deployed number of SIM cards, and over-the-air updates haven’t proven to be a successful path for upgrade.

Network and device upgrades represent a second, more viable path to security. The use of 3G and 4G cellular standards (with better encryption options) and smartphones offer the potential for strong protections from both core network and end-to-end perspectives. The first suggestion, while slowly happening, is unlikely to be universal in the near future. The return on investment for ripping out the massively deployed infrastructure and replacing it with an expensive new network is low. That’s not to say that more 3G and 4G infrastructures aren’t being deployed; rather, the pace at which they’re being rolled out is slow in the developing world. More critically, these networks don’t provide end-to-end cryptographic protection of user data flows, meaning that a total replacement of all 2G networks alone wouldn’t solve the security problems discussed earlier.

Much of our research has focused on mobile money applications for smartphones because they represent the most practical and rapid path to security. Smartphones come equipped with libraries containing an array of strong encryption algorithms, making it possible for developers to quickly and correctly provide end-to-end security for their applications. In 2015, we undertook a major effort to measure how well such mechanisms were being used.⁴ What we found was disheartening. Using a combination of automated and manual analysis, we discovered widespread misuse of insecure protocols, failure to properly authenticate users and mobile money entities, and poor SSL/TLS configuration on back-end servers (among many other issues). Our comprehensive teardown of seven applications revealed that we could steal money from six of them with ease. Moreover, the terms of service in all these applications made customers responsible for all fraud, even though we demonstrated that funds could be stolen without any negligence (for instance, giving out their PIN) on the consumers’ part.

These weaknesses were covered in news outlets including the *Wall Street Journal*, and we worked diligently behind the scenes to provide each of the at-risk companies with detailed vulnerability reports. We also worked with the GSMA and the ITU to spread word of the problems as well as how they could be addressed at low cost (for instance, correct configuration or code updates). However, when we remeasured the applications a year later, we saw not only that the majority of vulnerabilities hadn’t been fixed (in spite of promises to the contrary) but also that development of new features and interfaces had proceeded significantly.⁵

Much remains to be done by the research community. We need

to make it harder to design applications that use insecure communications. Although Android took significant steps forward in this space, the amount of insecure code and security bypasses discovered in the recovered code means that we aren’t there yet. Mechanisms that prevent the submission of applications that fail to properly use TLS would be great, but creating tools to do this will require extremely careful design. Moreover, because of the lack of an obvious push to replace feature phones and 2G networks, easy-to-deploy protocols and solutions are critical. Too many academics view GSM networks and feature phones as “solved” problems, but the reality is that like any massively deployed infrastructure (think COBOL in banking or the magnetic stripe on credit cards), they will never fully be removed from service, especially in the developing world.

What Is the State of Privacy?

Mobile money creates new privacy challenges. Whereas traditional banks are limited to seeing exchanges between their customers and vendors, the peer-to-peer nature of mobile money systems means that providers can observe additional social interactions. For instance, a group of people eating a meal together might send money to one another. Whereas traditional payment systems would have allowed a bank to see that all such attendees were at a restaurant at the same time, mobile money transaction data could be used to definitively link these attendees. Smartphone platforms also offer mobile money applications access to a wealth of additional information, including GPS location.

We don’t believe that collecting such data is inherently problematic. In fact, it’s being used as a means of bootstrapping emerging

credit offerings. In settings in which traditional metrics for determining credit-worthiness aren't available (for example, citizens might not file tax returns or have an official address, a mortgage, or an official history of payments), such data is beginning to act as a substitute. M-Shwari, which offers interest-bearing savings and loans to M-Pesa customers in Kenya, uses M-Pesa usage history to develop credit scores. Such loans have proven critical to merchants, who can eliminate the cash flow issues that traditionally made fully stocking their shelves a challenge.

We believe that consumers should be made aware of how their data is being collected and used and, therefore, be able to make informed decisions when selecting a mobile money or digital credit service. As such, our most recent research efforts have focused on a comprehensive study of privacy policies for mobile money applications.⁶ We collected privacy policies for all 54 mobile Android-based money applications listed by the GSMA and compared these policies to those of the top 50 US financial institutions as listed by the Federal Deposit Insurance Corporation (FDIC; an independent government body in the US responsible for providing regulation for the nation's banks, insurance for deposits, and consumer protection). Although many in the privacy community have opined about what financial privacies should look like ideally, in our evaluation, we relied instead on GSMA and FDIC recommendations. This was important because it let us measure compliance with their communities' published standards.

The results of this were similarly discouraging. Of the 54 studied mobile money applications, only 30 (54 percent) had privacy policies at all. A full third of those that had policies weren't written in either of the two most common languages spoken

in the country, meaning that many in the targeted customer demographics would simply be unable to read such terms. Finally, in the cases in which privacy policies were available, many were too short to contain meaningful content (for instance, EcoCash and TigoPesa's policies were 68 and 268 words long, respectively), or they lacked any mention of critical issues (for instance, fewer than half of those with policies had definitions of terms, mentions of accountability and enforcement, or data retention policies). Finally, mobile money privacy policies also tended to be more difficult to read according to several grade-level readability tests (for example, the Gunning-Fogg index). Given lower literacy rates in many of the populations served by mobile money applications, these results were troubling.

These results were in stark comparison to the traditional financial institutions, which were directly regulated by the FDIC. Mobile money systems, however, generally don't fall under the same regulatory bodies as financial institutions. Adding regulations isn't a simple solution. Many mobile money applications offer low transaction costs because their compliance costs are low. Moreover, these systems exist across a wide array of countries, each with cultures that hold different values to individual data privacy. Accordingly, creating a single set of strong privacy standards that meet universal approval is unlikely to be successful. We instead recommend that the industry push for stronger enforcement of the ideals put forth by the GSMA. Methods and tools for ensuring such compliance, however, remain a research challenge.

The rate at which mobile money systems are bringing traditionally unbanked populations into the global financial infrastructure is unprecedented and absolutely requires new ways of reasoning about and enforcing consumer protection.

We're firm believers in the transformative power of mobile money systems. We also believe that they will connect the finances of the developed and developing worlds in the most meaningful way yet in human history. Accordingly, the price for getting security and privacy wrong is extremely high.

Meaningfully addressing these challenges will require the efforts of our large community. We're trying to expand our engagement through an upcoming NSF-sponsored workshop entitled "Addressing the Technical Security Challenges of Emerging Digital Financial Services." Here, we hope to engage some of the top academic and industrial minds in the details of the challenges we've listed here. Other issues are also critical to address, including how to establish programming interfaces that let developers securely perform critical financial functions in mobile applications; how to ensure the security of legacy 2G infrastructure; and how to address the usability gap when populations with limited literacy and exposure to finance, who represent some of the populations most vulnerable to fraud, are using mobile money. Successfully addressing these problems will require a unique and sustained effort among academia, industry, and nongovernmental organizations. ■

Acknowledgments

This work was supported in part by the NSF under grants CNS-1526718 and CNS-1540217. Any opinions, findings, and conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

References

1. "2015 FDIC National Survey of Unbanked and Underbanked Households," Fed. Deposit Insurance Corporation, Oct. 2016; www.fdic.gov/householdsurvey/2015/2015execsumm.pdf.

2. "GSMA State of the Industry Report on Mobile Money, Decade Edition: 2006–2016," GSM Assoc., 2017; www.gsma.com/mobilefordevelopment/wp-content/uploads/2017/03/GSMA_State-of-the-Industry-Report-on-Mobile-Money_2016.pdf.
3. B. Ngugi, "M-Pesa Global Transactions Hit Six Billion in 2016," Business Daily, 26 Feb. 2017; www.businessdailyafrica.com/markets/MPesa-global-transactions-hit-six-billion-2016/539552-3828662-7h9g3xz/index.html.
4. B. Reaves et al., "Mo(bile) Money, Mo(bile) Problems: Analysis of Branchless Banking Applications in the Developing World," *Proc. USENIX Security Symp. (SEC 15)*, 2015, pp. 17–32; www.cise.ufl.edu/~traynor/papers/reaves-usenix15a.pdf.
5. B. Reaves et al., "Mo(bile) Money, Mo(bile) Problems: Analysis of

Branchless Banking Applications in the Developing World," *ACM Trans. Privacy and Security*, vol. 20, no. 3, 2017 article 10.

6. J. Bowers et al., "Regulators, Mount Up! Analysis of Privacy Policies for Mobile Money Services," *Proc. USENIX Symp. Usable Privacy and Security (SOUPS 17)*, 2017; www.cise.ufl.edu/~traynor/papers/bowers-soups17.pdf.

Patrick Traynor is the John and Mary Lou Dasburg Preeminent Chair in Engineering and associate professor at the University of Florida. He is also a Fellow of the Center for Financial Inclusion at Accion. Contact him at traynor@ufl.edu.

Kevin Butler is an associate professor at the University of Florida. He also serves as the vice chairman and leader of Security Workstream

for the International Telecommunication Union's Focus Group on Digital Financial Services. Contact him at butler@ufl.edu.

Jasmine Bowers is a PhD student at the University of Florida. Contact her at jdbowers@ufl.edu.

Bradley Reaves is an assistant professor at North Carolina State University. Contact him at bgreaves@ncsu.edu.

myCS


Read your subscriptions through the myCS publications portal at <http://mycs.computer.org>



Call for Software Engineering Award Nominations

Established in memory of Harlan D. Mills to recognize researchers and practitioners who have demonstrated long-standing, sustained, and impactful contributions to software engineering practice and research through the development and application of sound theory. The award consists of a \$3,000 honorarium, plaque, and a possible invited talk during the week of the annual International Conference on Software Engineering (ICSE), co-sponsored by the IEEE Computer Society Technical Council on Software Engineering.

Deadline for 2018 Nominations:
1 October 2017

Nomination site:
awards.computer.org
IEEE  computer society

*The award nomination requires at least 3 endorsements.
Self-nominations are not accepted.
Nominees/nominators do not need
to be IEEE or IEEE Computer Society members.*

Availability of Required Data to Support Criminal Investigations Involving Large-Scale IP Address-Sharing Technologies

David O'Reilly | FTR Solutions

Large-scale IP address-sharing technologies (often collectively referred to as Carrier-Grade Network Address Translation [CG-NAT]¹) are a helpful tool for extending the life of IPv4 by allowing multiple endpoints to share a small number of IPv4 addresses. Several such technologies have been discussed and deployed.²⁻⁵ A related category of technologies, known as Address Plus Port, or A+P,⁶ is also used for large-scale IP address sharing, achieved in these cases by using some of the port number bits for addressing purposes. Multiple examples of this category of technologies are also available.⁷⁻⁹

All of these technologies involve extending the space of available IPv4 addresses by mapping

communication from multiple endpoints to a single address, or a small number of shared addresses, using port numbers. The details of how this is achieved in each technology vary, but the principle is the same in all cases.

From the perspective of an Internet server, endpoint traffic that has passed through an IP address-sharing infrastructure appears to originate from the IP of the address-sharing appliance. Today, common practice is for servers to log the connection time and source IP address of incoming connections. However, the IP address of the address-sharing appliance isn't sufficient to identify the true source of the traffic because potentially hundreds or thousands of individual endpoints were using that IP address at the same time. If a criminal

investigation requires identification of the source of a specific connection, the source port and exact connection time will also be required. Without this additional information, it's highly unlikely that law enforcement authorities will be able to perform their investigations.

Operators of large-scale IP address-sharing infrastructures, typically Internet service providers, are usually required by law to maintain records of which endpoint used a particular IP address and port at a particular time. The period of time for which these records must be retained is defined by national legislation. However, IP address sharing hampers the ability to trace network use and abuse, and this challenge is likely to become more severe and widespread with the increased use of large-scale address sharing.¹⁰ More recently, Europol highlighted the issue of large-scale IP address sharing as a threat to Internet governance, reporting that the problem of crime attribution related to the use of CG-NAT technologies was regularly encountered by 90 percent of respondents to a survey on the topic.¹¹

Previous work has already suggested that, as best practice, Internet-facing servers should log source IP address, source port, and exact connection time.¹² However, no detailed consideration has been given to possible approaches to and implications of this proposed logging practice.

In this article, I describe how to bring about Internet-facing servers' routine logging of the information needed to reestablish the ability to trace network abuse.

Centralized Connection Logging

RFC6269 (Issues with IP Address Sharing) describes two ways to record adequate information to identify the parties of a particular connection:⁹

- IP address-sharing infrastructure operators log mappings between their subscribers and external IP address-port combinations. Internet-facing server operators log the IP address and source port of incoming connections. This is referred to as source *port logging*.
- Instead of relying on server operators to log the source port of incoming connections, operators of IP address-sharing infrastructures additionally log all destination IP addresses for outgoing connections. This is referred to as *connection logging*. Server operators continue to log only the IP address of incoming connections, which is the common current practice.

RFC6269 presents two challenges to the routine use of connection logging.

The first issue is that the large volume of data makes centralized connection logging infeasible. Whether destination IP addresses are recorded or not, the volume of logs generated by a large-scale IP address-sharing infrastructure will be substantial. Some approaches have been proposed to address this hurdle and make central connection logging more feasible, such as deterministic allocation of ports^{10,13} or allocation of port ranges.^{6,14} RFC7422 includes some representative figures for the scales of data involved; it's estimated that the logging overhead would be on the order of 150 Mbytes per subscriber, per month.¹³ In addition to the technical overhead of storing such a large volume of data, searching and locating relevant records over

a legally mandated retention period would also present a significant technical challenge.

The second issue raised in RFC6269 against connection logging is that even if connection logs store all combinations of timestamp, source IP, source port, and destination IP, querying this information without a source port (because the service operator hasn't recorded the source port) wouldn't be sufficient to distinguish the activity of one individual from another in cases in which the destination IP is popular. This problem is further exacerbated in the case of protocols that make multiple connections per session (for example, HTTP/HTTPS) and in cases of criminal activity that involve deliberate generation of large volumes of traffic (for example, distributed denial of service). Thus, connection logging alone, despite potentially significant technical and operational overhead, can't guarantee that the retained information is sufficient to identify an individual suspect.

Separately, the privacy concerns arising from connection logging have also been repeatedly raised.^{15,16}

In summary, it's clear that large-scale IP address-sharing infrastructure operators need to retain records to enable the identification of suspects; however, there's no centralized solution that removes the need for Internet-facing server operators to retain source port information.

Challenges to Capturing Source Port

It's relatively easy to explain why an operator of an Internet-facing server would want to retain source port information for incoming connections. If server operators (or the users that they serve) find themselves the victim of a crime, having information to facilitate a criminal investigation is preferable. On the other hand, there are numerous reasons

why a server operator might not have the required source port information. In this section, I enumerate factors that could negatively influence server operators' ability and inclination to capture and record source port information.

Lack of Awareness

One of the main problems with the increasing use of large-scale IP address-sharing technologies is server operators' lack of awareness that there are direct implications for them should they (or their users) become the victim of a crime.

At the time of writing, a minimal amount of material is available online concerning this issue, even for those actively seeking to find information on source port logging. Where vendors have provided guidance or information concerning the logging of incoming source ports, no explanation is provided for why this is something that server operators might want to do.

There is, therefore, a considerable awareness gap between the importance of this issue for investigating criminal activity online and the awareness of those who need to act to ensure availability of the information needed to facilitate a criminal investigation.

Poor Software Support for Logging Source Ports

Before server operators can decide to log source port information, the software they are using must support logging of incoming connections' source ports. Many, but not all, major software distributions support such logging. Lack of support in server software is an insurmountable technical obstacle for a server operator.

In some cases, even where software supports logging the source port of incoming connections, it can only be achieved by enabling verbose logging in the software. This would substantially (and

unnecessarily) increase the size of the logs produced by the server and reduce the server's production performance.

Many major software distributions provide default log formats in their configuration files. A review of the default log format of the latest versions of some common server software has been carried out, and in only one case (OpenSSH 7.5) was the source port of incoming connections logged by default.¹⁷

Breaking Downstream Tooling

By default, commercial and free log analysis software expects logs to be in a particular format. Consider, for example, the ubiquity of the Apache Common and Extended Log Formats. Most software can be configured to parse arbitrary log formats, but this is additional configuration work for a server operator.^{18,19} Without migration planning, a change to default log formats would likely cause substantial disruption to a considerable amount of downstream processing of server log files. In addition to commercial and freely available software, many administrators have developed or downloaded scripts that expect logs to be in a particular log format.

Therefore, log processing software, and in particular custom scripts, might break if log formats change unexpectedly. The tooling might need to be updated to correctly process the additional fields now present in log files.

Accuracy of Recorded Time

In addition to recording the connection's IP address and source port, it's important to record the exact connection time. It's been suggested that there's a need to keep the exact time against some sort of global standard (for instance, Network Time Protocol [NTP]);¹² however, this might not be possible for practical, security, or legacy reasons. In practice, it's usually not necessary

to keep time against a global standard, as long as time is recorded consistently. Any discrepancies can be calculated and compensated for manually. Time offsets of this nature are commonly encountered and well understood in the digital forensics world.

Conclusions and Next Steps

There's clearly substantial work to be done to bring about the regular recording of source port information at Internet-facing servers, and there are undoubtedly criminals free right now because the information required to identify them from their online activity isn't available.

I present some possible courses of action based on the current state of source port logging.

Raise Awareness of Logging Source Port in Deployment Guidance

Both free and commercial software publishers should consider releasing deployment guidance or best practices that describe why server administrators need to record source port information, as well as instructions for how to do this. This will help to address the lack of awareness of this issue's importance.

Considering also the awareness of those building software applications, or otherwise involved with coding of Internet-facing applications, secure coding guidance should be updated to include reference to source port information, particularly where such guidance already touches on the issue of logging. For example the OWASP Secure Coding Practices specifies a list of important log event data.²⁰ However, at the time of writing, the "important log event data" list doesn't include source port.

Increase Software Support for Logging Source Port

Many software packages support logging of source port information,

but only 11 of 16 examined in a recent study support logging in a way that wouldn't significantly negatively impact server software operation.¹⁸ Therefore, software publishers must consider their level of support for logging source port. In particular, software should support the logging of source port without needing to enable a verbose logging level.

Change Default Log Formats to Include Source Port

In cases in which a particular software package supports logging of incoming source port, one possibility is to incorporate one or more log formats that include incoming source port as a field logged by default. Obviously, this won't impact deployments of software already in place, but for future deployments, incorporating source port into the log format means that administrators using the unaltered default log format will automatically store the required information.

Parallel Logging to a Connection Log

Configuring parallel logging of connection information to a separate log stream is a possible solution to address the fact that changes to log format might break downstream tooling. It's also a possible solution for server software types that log via syslog. In this case, software publishers could produce guidance on how to configure syslog to log connection information parallel to main log files.

Such a solution would help to ease the transition to an alternate log format: current log formats wouldn't need to be changed because the required source port information is stored separately but can still be correlated with the main log files.

Adequate Timestamp Accuracy in Logs

Operators of large-scale address-sharing infrastructure will likely

need connection times specified with the granularity of at least one second. Most server software will log times with this granularity by default, but there's no guarantee that this is the case.

Server operators should ensure that the times being recorded in their log files have sufficient accuracy to allow identification of the required records. As mentioned earlier, the times don't necessarily need to be recorded with reference to a centralized time source (for instance, NTP) as long as they're recorded consistently.

This factor must also be considered by software developers when they produce software; although time recording is mentioned in the OWASP Secure Coding Practices, the required accuracy/granularity of the recorded time is not discussed.

Much work needs to be done to bring about the routine logging of source port information, and there's no centralized solution to this problem. Ultimately what's required is a wide recognition that IP addresses don't necessarily represent individual users' activity on the Internet. A shift in understanding, combined with raising awareness of this issue, will hopefully lead to an increase in the availability of the information that needs to be logged and more online criminals being brought to justice. ■

References

1. S. Perreault, ed., "Common Requirements for Carrier-Grade NATs (CGNs)," RFC6888, Apr. 2013; tools.ietf.org/html/rfc6888.
2. A. Durand et al., "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion," RFC6333, Aug. 2011; www.rfc-editor.org/info/rfc6333.
3. M. Bagnulo, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers," RFC6164, Apr. 2011; www.rfc-editor.org/info/rfc6146.
4. I. Yamagata et al., "NAT 444," Internet Eng. Task Force, 5 Jan. 2013; tools.ietf.org/html/draft-shirasaki-nat444-06.
5. T. Anderson and A. Leiva Popper, "Explicit Address Mappings for Stateless IP/ICMP Translation," RFC7757, Feb. 2016; www.rfc-editor.org/info/rfc7757.
6. R. Bush, ed., "The Address Plus Port (A+P) Approach to the IPv4 Address Shortage," RFC6346, Aug. 2011; www.rfc-editor.org/info/rfc6346.
7. Y. Cui et al., "Lightweight 4over6: An Extension to the Dual-Stack Lite Architecture," RFC7596, July 2015; tools.ietf.org/html/rfc7596.
8. O. Troan, ed., "Mapping of Address Port with Encapsulation (MAP-E)," RFC7597, July 2015; tools.ietf.org/html/rfc7597.
9. X. Li et al., "Mapping of Address and Port Using Translation (MAP-T)," RFC7599, July 2015; tools.ietf.org/html/rfc7599.
10. M. Ford, ed., "Issues with IP Address Sharing," RFC6269, June 2011; www.rfc-editor.org/info/rfc6269.
11. "The Internet Organised Crime Threat Assessment (IOCTA) 2016," Europol, 2016; www.europol.europa.eu/activities-services/main-reports/internet-organised-crime-threat-assessment-iocta-2016.
12. A. Durand et al., "Logging Recommendations for Internet-Facing Servers," RFC6302, June 2011; www.rfc-editor.org/info/rfc6302.
13. C. Donley et al., "Deterministic Address Mapping to Reduce Logging in Carrier-Grade NAT Deployments," RFC7422, Dec. 2014; www.rfc-editor.org/info/rfc7422.
14. T. Tsou et al., "Port Management to Reduce Logging in Large-Scale NATs," RFC7768, Jan. 2016; www.rfc-editor.org/info/rfc7768.
15. S. Perrault, ed., "Common Requirements for Carrier-Grade NATs (CGNs)," RFC6888, Apr. 2013; www.rfc-editor.org/info/rfc6888.
16. S. Sivakumar and R. Penno, "IPFIX Information Elements for Logging NAT Events," draft, Internet Eng. Task Force, 9 Jan. 2017; tools.ietf.org/html/draft-ietf-behave-ipfix-nat-logging-13.
17. D. O'Reilly, "Approaches to Address the Availability of Information in Criminal Investigations Involving Large-Scale IP Address Sharing Technologies," draft, Internet Eng. Task Force, 21 Aug. 2017; www.ietf.org/id/draft-daveor-cgn-logging-00.txt.
18. S. Turner, "Analog 6.0: Log Formats," Mirror Reverse, 19 Dec. 2004; mirror.reverse.net/pub/analog/docs/logfmt.html.
19. L. Destailleur, "AWStats Logfile Analyzer 7.6 Documentation," AWStats, awstats.sourceforge.io/docs/awstats_setup.html.
20. "OWASP Secure Coding Practices Quick Reference Guide," OWASP, 2010; www.owasp.org/images/0/08/OWASP_SCP_Quick_Reference_Guide_v2.pdf.

David O'Reilly is chief technologist at FTR Solutions. Contact him at dave.oreilly@ftrsolutions.com.

myCS

Read your subscriptions through the myCS publications portal at <http://mycs.computer.org>

The Open Science Cyber Risk Profile: The Rosetta Stone for Open Science and Cybersecurity

Sean Peisert | Berkeley Lab
Von Welch | Indiana University

A common misconception—one often held even by scientists—is that open science is “open” by definition, so hackers wouldn’t target it. The reality is that even open science is rarely *entirely* open at all times. For example, it can often be misleading to the public or even other researchers to publish raw data before it’s been verified, validated, and interpreted. Beyond situations in which raw data is published almost immediately, there are certainly many circumstances in which raw data contains valuable intellectual property that could be at risk of theft—both domestically and internationally. Or data might contain personally identifiable information, such as during clinical drug trials.

Moreover, it would be a mistake to ignore security risks outside confidentiality, including integrity and availability. While scientists might not feel anyone wants to interfere with their results, any scientist developing or testing something of commercial value can certainly be

at risk of having their work tampered with in a way that causes it to behave unpredictably or to make something look more or less successful than it actually is. Consider the possibilities of tampering with science related to politically sensitive subjects or public safety, such as meteorology or public health.

The reality is that, aside from the “why me?” question, the most important issue is really the “what if” question. Producing scientific results takes months or years of careful labor of many people using expensive and often unique instruments. These results, in turn, are often built upon by others, again over months, years, or even decades. While the scientific process has done a good job of finding errors and inaccuracies in science, there are steps to help this process with regard to errors owing to computer attacks. The goal is to mitigate errors from the outset, or at least spend less time and money to identify them after they do happen.

Bringing cybersecurity to bear on open science often presents both a culture clash and a knowledge gap. Cybersecurity professionals don’t have much experience with rare, even unique, scientific instruments, and the sensitivities of their data, unlike say HIPAA (Health Insurance Portability and Accountability Act) regulatory data, aren’t defined. Scientists, believing themselves to not be targets, will often see cybersecurity as simply administrative hindrances to their work. The result is that the application of cybersecurity to open science can be off target—an impediment to science and less than optimally effective.

The Open Science Cyber Risk Profile (OSCRP) aims to help improve IT security for open science projects—that is, science that’s unclassified and often funded by US government agencies, such as the NSF, the Department of Energy’s Office of Science, and the National Institutes of Health. The OSCRP working group has created a document that motivates scientists by demonstrating how improving their security posture reduces the risks to their science, and enables them to have a conversation with IT security professionals regarding those risks so that appropriate mitigations can be discussed.

Given all the potential risks, the OSCRP working group examined a variety of different types of scientific computing-related assets and divided them into key categories, including various types of

- data (for instance, public data, embargoed data, and internal data),

- facilities (for instance, physical storage, power, and climate control),
- system and hardware assets (for instance, networks, front ends, servers, databases, and mobile devices),
- software assets (including both internal and third-party software),
- instruments (for instance, sensors or control systems), and
- intangible and human assets (ranging from project reputation to human staff to collaborative materials and financial assets).

Note that it's key that the working group focused on *assets*, which are things that a scientist knows and cares about, rather than specific *threat actors*, which are difficult for anyone to predict and whose motivations and tactics change over time (for example, the rise of ransomware over the past few years has greatly changed the threat landscape).

To accomplish this task, we assembled a group of security experts as well as domain scientists running large science projects, including particle physicists, oceanographers, genomic researchers, and more.

This group considered a set of common open science assets as well as how open science projects relied on each—and, hence, the risks associated with each asset's failures. We then mapped possible IT threats to these science risks. Scientists can use the OSCRP document to enumerate all the assets of importance and the risks each brings to their science mission. Using this information, they can prioritize the relevant IT threats. IT security professionals can then design and implement appropriate mitigations tuned specifically for the science risks, and scientists would understand the value of these mitigations.

It's our hope that this document helps scientists better understand reasons why they might be interested in pursuing further discussions with computer security experts and, conversely, help

institutional community efforts best convey important messages to domain scientists about the risks to open science.

The OSCRP can be found at trustedci.github.io/OSCRP. It reflects an initial set of assets and the group's early valuation of those assets' risks. Over time, assets will change and so will risks; hence, we envision it as a living document that will evolve over time. To this end, we followed a NIST practice and used the popular GitHub source code repository to author the OSCRP. This allows for the public's submission of proposed additions, changes, and comments on the document. Note that the lists of assets and their risks are not comprehensive; more contributions in either of these areas are welcome. We've already received some great community feedback and hope for not just more feedback but a community sense of ownership.

Although open science is indeed open, it's not exempt from the risks of computer-related attacks, and there are cultural and technical challenges to applying current cybersecurity approaches. We hope the OSCRP serves to bridge the communication gap between scientists and IT security professionals and allows for the effective management of risks to open science caused by IT security threats. ■

Sean Peisert is a staff scientist at Lawrence Berkeley National Laboratory, chief cybersecurity strategist at CENIC, and an associate adjunct professor at UC Davis. Contact him at speisert@lbl.gov.

Von Welch is director of the Center for Applied Cybersecurity Research and the NSF Cybersecurity Center of Excellence at Indiana University. Contact him at vwelch@iu.edu.

got flaws?



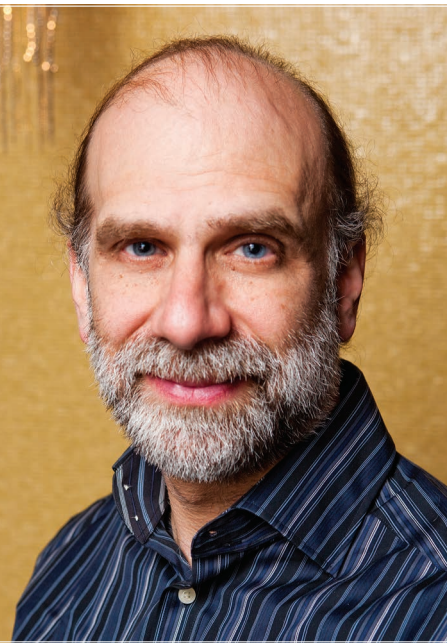
Find out more and get involved:

cybersecurity.ieee.org



IEEE computer society





Bruce Schneier
Harvard University

IoT Security: What's Plan B?

In August, four US Senators introduced a bill designed to improve Internet of Things (IoT) security. The IoT Cybersecurity Improvement Act of 2017 is a modest piece of legislation. It doesn't regulate the IoT market. It doesn't single out any industries for particular attention, or force any companies to do anything. It doesn't even modify the liability laws for embedded software. Companies can continue to sell IoT devices with whatever lousy security they want.

What the bill does do is leverage the government's buying power to nudge the market: any IoT product that the government buys must meet minimum security standards. It requires vendors to ensure that devices can not only *be* patched but *are* patched in an authenticated and timely manner, don't have unchangeable default passwords, and are free from known vulnerabilities. It's about as low a security bar as you can set, and that it would considerably improve security speaks volumes about the current state of IoT security. (Full disclosure: I helped draft some of the bill's security requirements.)

The bill would also modify the Computer Fraud and Abuse and the Digital Millennium Copyright Acts to allow security researchers to study the security of IoT devices purchased by the government. It's a far narrower exemption than our industry needs. But it's a good first step, which is probably the best thing you can say about this legislation.

However, it's unlikely this first step will even be taken. I am writing this column in August, and have no doubt that the bill will have gone nowhere by the time you read it in October or later. If hearings are held, they won't matter. The bill won't have been voted on by any committee, and it won't be on any legislative calendar. The odds of this becoming law are zero. And that's not just because of current politics—I'd be equally pessimistic under the Obama administration.

But the situation is critical. The Internet is dangerous—and the IoT gives it not just eyes and ears, but also hands and feet. Security vulnerabilities, exploits, and attacks that

once affected only bits and bytes now affect flesh and blood.

Markets, as we've repeatedly learned over the past century, are terrible mechanisms for improving the safety of products and services. It was true for automobile, food, restaurant, airplane, fire, and financial-instrument safety. The reasons are complicated, but basically, sellers don't compete on safety features because buyers can't efficiently differentiate products based on safety considerations. The race-to-the-bottom mechanism that markets use to minimize prices also minimizes quality. Without government intervention, the IoT remains dangerously insecure.

The US government has no appetite for intervention, so we won't see serious safety and security regulations, a new federal agency, or better liability laws. We might have a better chance in the EU. Depending on how the General Data Protection Regulation on data privacy pans out, the EU might pass a similar security law in five years. No other country has a large enough market share to make a difference.

Sometimes we can opt out of the IoT, but that option is becoming increasingly rare. Last year, I tried and failed to purchase a new car without an Internet connection. In a few years, it's going to be nearly impossible to not be multiply connected to the IoT. And our biggest IoT security risks will stem not from devices we have a market relationship with, but from everyone else's cars, cameras, routers, drones, and so on.

We can try to shop our ideals and demand more security, but companies don't compete on IoT safety—and we security experts aren't a large enough market force to make a difference.

We need a plan B, although I'm not sure what that is. Email me if you have any ideas. ■

Bruce Schneier is a security technologist and a Fellow at the Berkman Klein Center for Internet and Society at Harvard University. He's also the chief technology officer of IBM Resilient and special advisor to IBM Security. Contact him via www.schneier.com.



PREFERRED PLUS



TRAINING
& DEVELOPMENT



RESEARCH



BASIC



STUDENT

New Membership Options for a Better Fit

And a better match for your career goals. Now IEEE Computer Society lets you choose your membership — and the benefits it provides — to fit your specific career needs. With four professional membership categories and one student package, you can select the precise industry resources, offered exclusively through the Computer Society, that will help you achieve your goals.



IEEE  computer society

Learn more at www.computer.org/membership.



ELECTRONIC EDITION

IEEE

SECURITY & PRIVACY

SUBSCRIBE FOR \$39

- Protect your network
- Further your knowledge with in-depth interviews with thought leaders
- Access the latest trends and peer-reviewed research anywhere, anytime



\$69 Print Edition



\$39 Electronic Edition
for Computer Society and Reliability Society members

mycs.computer.org