COMPUTING

edge

- IoT Security
- Blockchain's Rise
- Healthcare Technology
- Social Media Sovereignty

APRIL 2018

www.computer.org

IEEE

IEEE computer society

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

COMPUTING edge

IEEE COMPUTER SOCIETY computer.org • +1 714 821 8380

Printed with inks containing soy and/or vegetable oils

IEEE

IEEE computer society

STAFF

Editor
Meghan O'Dell

Contributing Staff
Christine Anthony, Lori Cameron, Lee Garber, Cathy Martin, Chris Nelson, Dennis Taylor, Rebecca Torres, Bonnie Wylie

Production & Design
Carmen Flores-Garvey

Managers, Editorial Content
Brian Brannon, Carrie Clark

Publisher
Robin Baldwin

Director, Products and Services
Evan Butterfield

Senior Advertising Coordinator
Debbie Sims

Circulation: ComputingEdge (ISSN 2469-7087) is published monthly by the IEEE Computer Society. IEEE Headquarters, Three Park Avenue, 17th Floor, New York, NY 10016-5997; IEEE Computer Society Publications Office, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720; voice +1 714 821 8380; fax +1 714 821 4010; IEEE Computer Society Headquarters, 2001 L Street NW, Suite 700, Washington, DC 20036.

Postmaster: Send address changes to ComputingEdge-IEEE Membership Processing Dept., 445 Hoes Lane, Piscataway, NJ 08855. Periodicals Postage Paid at New York, New York, and at additional mailing offices. Printed in USA.

Editorial: Unless otherwise stated, bylined articles, as well as product and service descriptions, reflect the author's or firm's opinion. Inclusion in ComputingEdge does not necessarily constitute endorsement by the IEEE or the Computer Society. All submissions are subject to editing for style, clarity, and space.

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 copy-editing, proofreading, and formatting added by IEEE. For more information, please go to: http://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 08854-4141 or pubs-permissions@ieee.org. Copyright © 2018 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.

Unsubscribe: If you no longer wish to receive this ComputingEdge mailing, please email IEEE Computer Society Customer Service at help@computer.org and type "unsubscribe ComputingEdge" in your subject line.

IEEE prohibits discrimination, harassment, and bullying. For more information, visit www.ieee.org/web/aboutus/whatis/policies/p9-26.html.

IEEE Computer Society Magazine Editors in Chief

Computer
Sumi Helal, *Lancaster University*

IEEE Software
Diomidis Spinellis, *Athens University of Economics and Business*

IEEE Internet Computing
M. Brian Blake, *University of Miami*

IT Professional
Irena Bojanova, *NIST*

IEEE Security & Privacy
David M. Nicol, *University of Illinois at Urbana-Champaign*

IEEE Micro
Lieven Eeckhout, *Ghent University*

IEEE Computer Graphics and Applications
Torsten Möller, *Universität Wien*

IEEE Pervasive Computing
Marc Langheinrich, *University of Vienna*

Computing in Science & Engineering
Jim X. Chen, *George Mason University*

IEEE Intelligent Systems
V.S. Subrahmanian, *University of Maryland*

IEEE MultiMedia
Shu-Ching Chen, *Florida International University*

IEEE Annals of the History of Computing
Nathan Ensmenger, *Indiana University Bloomington*

IEEE Cloud Computing
Mazin Yousif, *T-Systems International*

MARCH 2018 • VOLUME 4, NUMBER 4

COMPUTING edge

26

Beyond Bitcoin: The Rise of the Blockchain World

Internet of Things

Blockchain

Healthcare

Social Media

Security

Departments

Subscribe to ***ComputingEdge*** for free at **www.computer.org/computingedge.**

Magazine Roundup

Editor: Lori Cameron

The IEEE Computer Society's lineup of 13 peer-reviewed technical magazines covers cutting-edge topics ranging from software design and computer graphics to Internet computing and security, from scientific applications and machine intelligence to cloud migration and microchip design. Here are highlights from recent issues.

Computer

Wise Computing: Toward Endowing System Development with Proactive Wisdom

In this article from the February 2018 issue of *Computer*, a broad, long-term research project is described, which will lead to the computer becoming an equal member of the system-development team. The computer will continuously make positive contributions, akin to those expected from an experienced and knowledgeable customer or user, a conscientious QA engineer, a strict regulatory auditor, an engineering-team leader, or the organization's CTO.

Computing in Science & Engineering

Math.Js: An Advanced Mathematics Library For JavaScript

Math.js is a JavaScript library that brings advanced mathematics to the web browser and server. The case study presented in this article from the January/February 2018 issue of *CiSE* demonstrates its flexibility by extending the library with custom functions to solve

 Published by the IEEE Computer Society 2469-7087/18/$33.00 © 2018 IEEE

and optimize a rocket trajectory. Several benchmark comparisons with other JavaScript libraries and state-of-the-art mathematics software are presented, as well as the current challenges facing math.js, including performance and size.

IEEE Annals of the History of Computing

Imagining the Personal Computer: Conceptualizations of the Homebrew Computer Club 1975–1977

The Homebrew Computer Club was a hobbyist group in the San Francisco Bay Area dedicated to helping people build their own home personal computers. In this article from the October–December 2017 issue of *IEEE Annals,* Elizabeth Petrick from the New Jersey Institute of Technology analyzes their writings between 1975 and 1977 to understand how their values became embedded in the technology they built, establishing how the personal computer should be used and thought of. These values were based on ideals of open information, access to computers, and the computer as a universal tool, while also allowing for development of entrepreneurial ambitions to market the computer as a consumer product.

IEEE Cloud Computing

Context Aware Ubiquitous Biometrics in Edge of Military Things

Edge computing can play a crucial role in enabling user authentication and monitoring through context-aware biometrics in military and battlefield applications. For example, in the Internet of Military Things or Internet of Battlefield Things, an increasing number of ubiquitous sensing and computing devices worn by military personnel and embedded within military equipment—such as combat suits, instrumented helmets, and weapons systems—are capable of acquiring a variety of static and dynamic biometrics like visual features, fingerprints, heart rate, gait, gestures, and facial expressions. Such devices might also be capable of collecting operational context data that can be used to perform context-adaptive authentication in the wild and continuous monitoring of soldiers' mental and physical conditions in a dedicated edge-computing architecture. Learn more in this article from the November/December 2017 issue of *IEEE Cloud Computing*.

IEEE Computer Graphics and Applications

A Generative Audio-Visual Prosodic Model for Virtual Actors

An important problem in the animation of virtual characters is the expression of complex mental states using the coordinated prosody of voice, rhythm, facial expressions, and head and gaze motion. The authors of this article from the November/December 2017 issue of *IEEE CG&A* propose a method for generating natural speech and facial animation in various attitudes using neutral speech and animation as input.

IEEE Intelligent Systems

Robots in Retirement Homes: Person Search and Task Planning for a Group of Residents by a Team of Assistive Robots

In this article from the November/December 2017 issue of *IEEE Intelligent Systems*, researchers from the University of Toronto present a general multi-robot task planning and execution architecture for a team of heterogeneous mobile robots that interact with multiple human users. The architecture is implemented in an environment where such robots provide daily assistance to residents in a retirement home setting. The robots are able to allocate and schedule activities throughout the day and find the appropriate residents with whom to engage in assistive activities.

IEEE Internet Computing

Internet of Things Enhanced User Experience for Smart Water and Energy Management

Smart environments can engage a wide range of end users with different interests and priorities, from corporate managers looking to improve the performance of their business to school children who want to explore and learn more about the world around them. Creating an effective user experience within a smart environment (from smart buildings to smart cities) is an important factor to success. In this article from the January/February 2018 issue of *IEEE Internet*

Computing, researchers reflect on their experience of developing Internet-of-Things-enabled applications within a smart home, school, office building, university, and airport, where the goal has been to engage a wide range of users (from building managers to business travelers) to increase water and energy awareness, management, and conservation.

IEEE Micro

High-Integrity Performance Monitoring Units in Automotive Chips for Reliable Timing V&V

As software continues to control more system-critical functions in cars, its timing is becoming an integral element in functional safety. Timing validation and verification (V&V) assesses software's end-to-end timing measurements against given budgets. The advent of multicore processors with massive resource sharing reduces the significance of end-to-end execution times for timing V&V and requires reasoning on worst-case access delays on contention-prone hardware resources. While Performance Monitoring Units (PMUs) support this finer-grained reasoning, their design has never been a prime consideration in high-performance processors. In this article from the January/February 2018 issue of *IEEE Micro*, researchers advocate for PMUs in automotive chips that explicitly track activities related to worst-case software behavior, are recognized as a mandatory high-integrity hardware service, and are accompanied with detailed documentation that enables their effective use to derive reliable timing estimates.

IEEE MultiMedia

Word of Mouth Mobile Crowdsourcing: Increasing Awareness of Physical, Cyber, and Social Interactions

By fully exploring various sensing capabilities and multiple wireless interfaces of mobile devices and integrating them with human power and intelligence, mobile crowdsourcing (MCS) is emerging as an effective paradigm for large-scale multimedia-related applications. However, most MCS schemes use a direct mode, in which crowd workers passively or actively select tasks and contribute without interacting and collaborating with each other. This can hamper some time-constrained crowdsourced tasks. In this article from the October–December 2017 issue of *IEEE MultiMedia*, researchers from universities in China, Japan, and Sweden execute a different approach: MCS based on word of mouth (WoM), in which crowd workers, apart from executing tasks, exploit their mobile social networks and/or physical encounters to actively recruit other appropriate individuals to work on the task.

IEEE Pervasive Computing

Designing Line-Based Shape-Changing Interfaces

In this article from the October–December 2017 issue of *IEEE Pervasive Computing*, researchers from Stanford and the MIT Media Lab present an overview of work on shape-changing line interfaces in the field of human–computer interaction (HCI), including their previous work on actuated-line interfaces (LineFORM and ChainFORM). They compare several potential implementation methods, discuss their potential for future research and applications, investigate the interaction design space around actuated line interfaces, and present potential applications and demonstrate their use with the LineFORM and ChainFORM prototypes. Envisioning a future where shape-changing lines are woven into daily life, this article aims to explore and initiate a broad research space around line-based shape-changing interfaces and to encourage future researchers and designers to investigate these novel directions.

IEEE Security & Privacy

Enhancing Selectivity in Big Data

Today's companies collect immense amounts of personal data and enable wide access to it within the company. This exposes the data to external hackers and privacy-transgressing employees. In this article from the January/February 2018 issue of *IEEE S&P*, researchers show that, for a wide and important class of workloads, only a fraction of the data is needed to approach state-of-the-art accuracy. They propose selective data systems that are designed to pinpoint the data that is valuable for a company's

current and evolving workloads. These systems limit data exposure by setting aside the data that is not truly valuable.

IEEE Software

Actionable Analytics for Strategic Maintenance of Critical Software: An Industry Experience Report

NASA has been successfully sustaining the continuous operation of its critical navigation software systems for over 12 years. To accomplish this, NASA scientists must continuously monitor their process, report on current system quality, forecast maintenance effort, and sustain required staffing levels. In this article from the January/February 2018 issue of *IEEE Software*, the authors present some examples of the use of a robust software metrics and analytics program that enables actionable strategic maintenance management of a critical system (Monte) in a timely, economical, and risk-controlled fashion.

IT Professional

Automatic Annotation of Text with Pictures

The vast array of information available on the Internet makes it challenging to quickly determine the importance and relevance of content. Text picturing is a cognitive aid that can help with text understanding, as it helps users decide if the text deserves a closer look by showing relevant pictures along with the text. Learn more in this article from the January/February 2018 issue of *IT Professional*.

Computing Now

The Computing Now website (computingnow.computer.org) features up-to-the-minute computing news and blogs, along with articles ranging from peer-reviewed research to opinion pieces by industry leaders. Read the latest Guest Editors' Introduction on current trends in visualization at www.computer.org/web/computingnow/archive/snapshot-trends-visualization-february-2018-introduction.

TECHNOLOGY

LinkedIn Corp.

has openings in our **Mountain View, CA** location for:

Software Engineer (All Levels/Types) (SWE0318MV) Design, develop & integrate cutting-edge software technologies.

LinkedIn Corp. has openings in our **Sunnyvale, CA** location for:

Software Engineer (All Levels/Types) (SWE0318SV) Design, develop & integrate cutting-edge software technologies; **Site Reliability Engineer (6597.2261)** Design, develop & integrate cutting-edge software technologies; **Senior Integration Developer, HR Technology (6597.2467)** Design, develop, configure, test, deploy, support & monitor HR reports & integrations; **Senior Technical Program Manager (Development Operations) (6597.2480)** Lead & develop site reliability engineering (SRE) & operations plans including application releases & updates, data & hardware migrations, & site stability & performance improvements; **Staff Engineer, Site Reliability (6597.2258)** Design, develop & integrate cutting-edge software technologies.

LinkedIn Corp. has openings in our **San Francisco, CA** location for:

Software Engineer (All Levels/Types) (SWE0318SF) Design, develop & integrate cutting-edge software technologies.

LinkedIn Corp. has openings in our **New York, NY** location for:

Machine Learning and Relevance Engineer (6597.2005) Work on massive semi-structured text, graph & user activity data sets to build highly scalable &performant social graph engines, state-of-the-art full-text search engines, & platforms for recommendation & applied data analytics; **Manager, Site Reliability Engineering (6597.2478)** Oversee engineers responsible for the overall health & operational design of company systems. Limited domestic & international travel required.

Please email resume to: 6597@linkedin.com. Must ref. job code above when applying.

EDITOR'S NOTE

Change Is in the Air

Now that springtime is here, we're turning over a new leaf—or petal, perhaps?—and shaking things up a bit here at *ComputingEdge*. Starting with this issue, instead of highlighting one theme per month, we'll be featuring as many as five themes per issue to bring our readers a broader sampling of the cutting-edge research from all 13 Computer Society magazines. In this issue, you'll read about the latest in the Internet of Things (IoT), blockchain, healthcare, social media, and security.

In *Computer*'s "How Do You Command an Army of Intelligent Things?," the authors look ahead to a future workforce made up of both humans and intelligent things, and argue that we'll need to understand and leverage the strengths and weaknesses of both to command and control this new organizational form.

The authors of *IEEE Security & Privacy*'s "Internet of Things Security Research: A Rehash of Old Ideas or New Intellectual Challenges?" discuss the challenges of building security into the new computing paradigm of IoT. They look at what IoT security problems can be solved using existing security principles and what new problems and challenges will require new security mechanisms.

Computing in Science & Engineering's "What Is the Blockchain?" reviews the basic ideas of this new technology—which is at the foundation of platforms for trading cryptocurrencies and executing smart contracts—and provides a sample minimalist implementation in Python.

The author of *Computer*'s "Beyond Bitcoin: The Rise of the Blockchain World" examines what a decentralized, secure financial system will mean for our society.

In *IEEE Software*'s "Toward Evidence-Based Software Engineering: Lessons Learned in Healthcare Application Development," the authors look back at the decisions made when designing, implementing, and evolving a collaboration tool to support evidence-based decisions in healthcare, and reflect on how software engineers could benefit from similar methods.

Using a set of unlabeled training examples representing occurrences of an event type such as disease outbreak, the authors of *IEEE Intelligent Systems*' "Graph Structure Learning from Unlabeled Data for Early Outbreak Detection" propose a novel framework for learning a graph structure that can be used to quickly and accurately detect—and thus prevent—future events of that type.

In *IEEE Internet Computing*'s "Social Media Won't Free Us," the author says that social media has been lauded as a democracy catalyzer while overlooking the adversarial opportunities it offers. He warns that while social media isn't going to overthrow authoritarianism, it might help pave the way for authoritarian manners in democratic countries.

The authors of *IT Professional*'s "It Doesn't Have to Be Like This: Cybersecurity Vulnerability Trends" look at the large and impactful data breaches making headlines in recent years and review trends in vulnerabilities.

Finally, from *IEEE Security & Privacy*, Gary McGraw interviews the chief information security officer of Eli Lilly and Company in "Silver Bullet Talks with Wafaa Mamilli," and the author of "Freedom of Encryption" reviews the main events in history that have shaped the legislative landscape that encompasses the use of encryption (paying particular attention to post-Snowden developments).

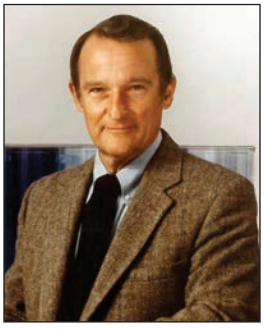
Published by the IEEE Computer Society 2469-7087/18/$33.00 © 2018 IEEE

Recognizing Excellence in High Performance Computing

Nominations are Solicited for the

SEYMOUR CRAY SIDNEY FERNBACH & KEN KENNEDY AWARDS

SEYMOUR CRAY COMPUTER ENGINEERING AWARD



Established in late 1997 in memory of Seymour Cray, the Seymour Cray Award is awarded to recognize innovative contributions to high performance computing systems that best exemplify the creative spirit demonstrated by Seymour Cray. The award consists of a crystal memento and honorarium of US$10,000. **This award requires 3 endorsements.**

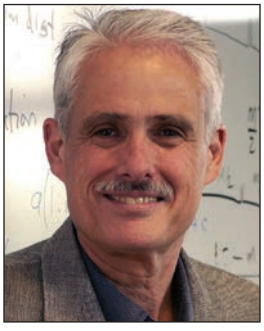
Sponsored by: IEEE computer society

SIDNEY FERNBACH MEMORIAL AWARD

Established in 1992 by the Board of Governors of the IEEE Computer Society. It honors the memory of the late Dr. Sidney Fernbach, one of the pioneers on the development and application of high performance computers for the solution of large computational problems. The award, which consists of a certificate and a US$2,000 honorarium, is presented annually to an individual for "an outstanding contribution in the application of high performance computers using innova-tive approaches." **This award requires 3 endorsements.**

Sponsored by: IEEE computer society

ACM/IEEE-CS KEN KENNEDY AWARD



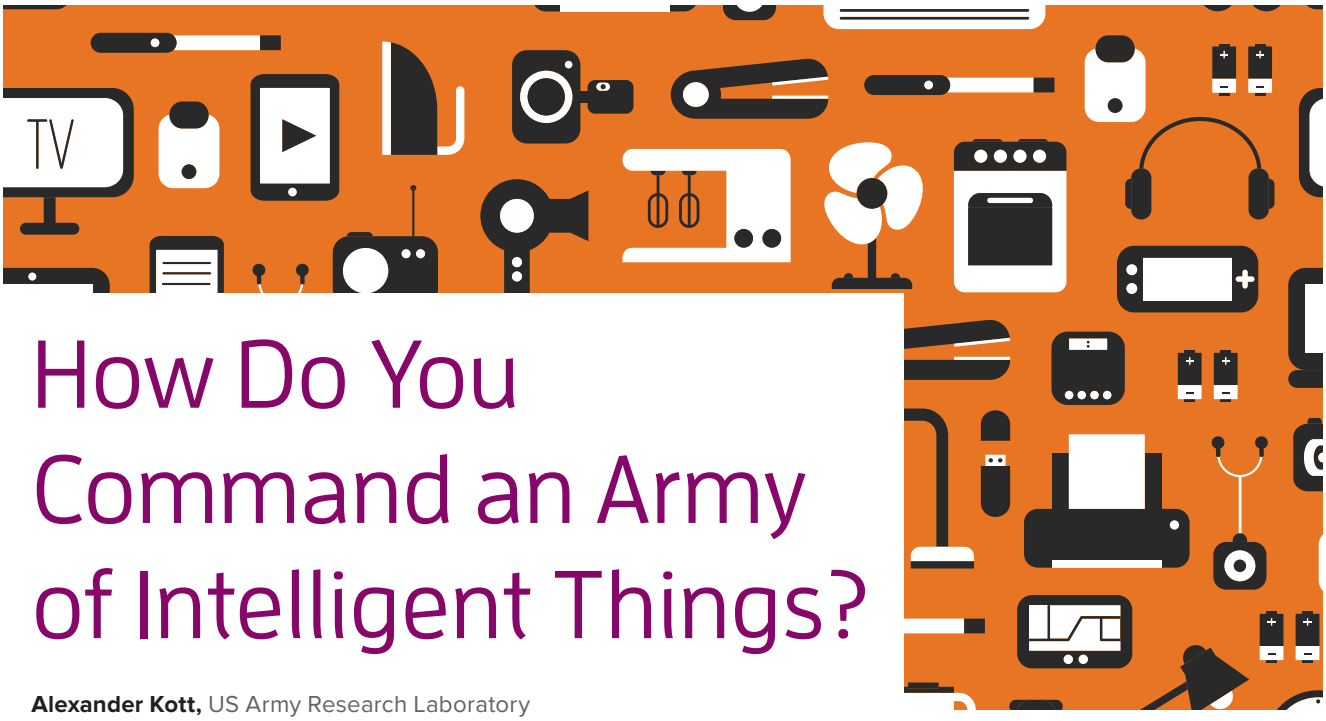
Established in memory of Ken Kennedy, the founder of Rice University's nationally ranked computer science program and one of the world's foremost experts on high-performance computing. A certificate and US$5,000 honorarium are awarded jointly by the ACM and the IEEE Computer Society for outstanding contributions to programmability or productivity in high performance computing together with significant community service or mentoring contributions. **This award requires 2 endorsements.**

Cosponsored by: IEEE computer society acm Association for Computing Machinery

Deadline: 1 July 2018

All nomination details available at http://awards.computer.org

IEEE



How Do You Command an Army of Intelligent Things?

Alexander Kott, US Army Research Laboratory
David S. Alberts, Institute for Defense Analysis

The future workforce will be made up of both humans and intelligent things. We'll need to understand and leverage the strengths and weaknesses of human cognition and machine intelligence to command and control this new organizational form.

FROM THE EDITOR

One of the challenges of working with the Internet of Things (IoT) is the sheer scale of the systems that need to be coordinated and controlled. This article provides a closer look at what this means in practice, and the challenges that result from having many IoT devices that also exhibit smart behaviors. Just as with human organizations, this becomes an issue of skillful management rather than absolute control.–*Roy Want*

The rapid development of information and communications technologies has led to the Internet of Things (IoT). As these "things" become more intelligent and, in some cases, capable of locomotion, the IoT will evolve into networks of advanced forms of machine intelligence that are capable of, and will possess, a degree of autonomy. Within the next few years, humans will need to find ways to work effectively with the ever-growing number of intelligent things that are appearing in our homes and places of work, including robots and intelligent agents.

The networked workforce of the near future will consist of not only interconnected and interdependent humans but also of intelligent things. The humans among them will find themselves to be merely a particular species of intelligent entities, and in fewer and fewer numbers in relation to other intelligent things. At least some of these intelligent things need to be considered, from a management perspective, as entities with decision-making responsibilities, similar to human individuals to be accounted for in the design and operations of our business organizations.

 Published by the IEEE Computer Society 2469-7087/18/$33.00 © 2018 IEEE

EDITOR **ROY WANT**
Google; roywant@google.com

This raises a number of challenging issues, none more compelling and urgent than finding an answer to the question, "How will we manage this new organizational form?" It's hard enough to effectively command or manage a purely human organization. It's even harder to do so with the agility required for the complex and dynamic environments in which we must operate. Can we rise to the occasion?

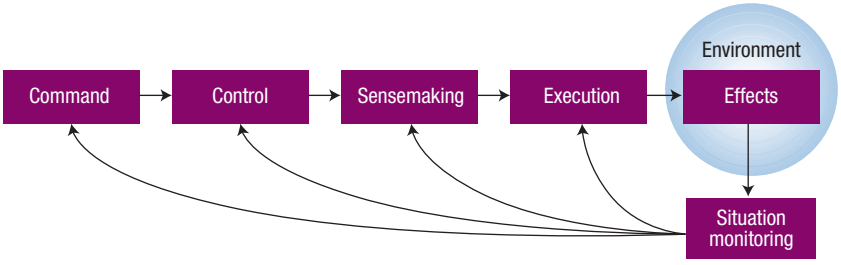


Figure 1. To achieve the desired effects in a complex, dynamic environment, an organization must execute five functions: command, control, sensemaking, execution, and situation monitoring.

COMMAND AND CONTROL: AN OVERVIEW

Let's consider these issues in a particularly challenging domain of human endeavor: warfare.[1] Command and Control (C2) is the term applied to the management or governance of military organizations and endeavors. The question, then, is, "How can we command and control an army of intelligent things?" Humans and other intelligent entities will each bring to the table different strengths to accomplish key C2 or management functions. We'll need to understand and leverage the comparative strengths and weaknesses of human cognition and machine intelligence to design an approach to C2 that's appropriate for this new organizational form, one that is agile enough to deal with unexpected developments.

Figure 1 depicts the five essential C2 functions (command, control, sensemaking, execution, and situation monitoring) necessary to achieve the desired effects in a dynamic context.[2] Successfully accomplishing these functions requires the ability to collect, process, and share information. We must consider how human and other intelligent entities can best ensure that the decision makers, whether human or machine, have the information they require and can make good use of this information to accomplish C2 functions.

Command

The function of command includes establishing intent and creating the conditions that enable an organization to achieve its desired results. Intent consists of setting goals and priorities, establishing rules, and setting constraints. Creating the conditions for success includes assigning roles and responsibilities and defining relationships.

In establishing and expressing intent, humans exhibit their share of shortcomings. Humans tend to be vague and unspecific about their intent. They also tend to make numerous assumptions, more often implicit and unspoken than explicitly articulated, and can engage in self-serving interpretations regarding priorities. Personal agendas and groupthink often dominate. Unlike humans, intelligent things are far more likely to stick to well-defined, formally stated goals and priorities, with no personal agendas. Inconsistent goals will be identified and highlighted. Humans, however, will find it difficult and infuriating to provide the level of formality and specificity in matters of intent, goals, and priorities that intelligent things require.

Every organization strives to determine roles and responsibilities, and to develop and nurture the necessary relationships within the organization to better match the requirements of the problems and solutions at hand. Recent research supports the thesis that there isn't a one-size-fits-all approach to C2 that's appropriate for all missions and circumstances, and that military organizations need to be prepared with C2 approach options; that is, different ways of allocating decision rights, interacting with one another, and accessing information.[3]

In a less formal and more ad hoc manner, humans within a decision-making network continually modify their own roles, tasks, and relationships with other nodes in the network. They also attempt to modify the roles, tasks, and relations of others, either by ordering organizational changes to subordinates or trying to influence other nodes. Humans also try to establish trust and cooperative relations with other nodes. Informal task forces or ad hoc groups are created, new experts or champions are discovered, and so on. In many situations, this results in adjustments to the structure of the organization that exhibits a better match to the problem at hand. In less satisfactory cases, the organization drives itself into an unproductive mode. How well would intelligent things cope with these dynamics? We suspect the current generation of intelligent things would fare rather poorly.

In the foreseeable future, intelligent things will remain largely incomprehensible to humans, without a natural, intuitively understandable set of

personalities. Establishing human-like trust between humans and things will be challenging at best (especially because intelligent things are susceptible to cyber-intrusions that might compromise their perceptions and decision making).[4] Intelligent things will also be relatively unsuccessful in negotiating with humans (but more successful when negotiating with other things),[5] and will find it hard to participate in defining a suitable role and task allocation in a mixed human–thing team. They'll find it even harder to adjust to changes in roles, task assignments, and relationships.

Control

Control involves making adjustments to actions, which take into consideration changes in the situation and the progress, or lack of progress, that has been achieved. This function of C2 provides the agility necessary to be successful in fluid and dynamic situations.

Control involves all other elements of the process in Figure 1, performed again and again—with agility—as the execution of the solution faces inevitable breakdowns and unexpected stumbling blocks. Humans are prone to information overload as they attempt to collect and absorb relevant new information. Often, they feel disoriented when presented with unexpected or unfamiliar situations—they take time to shift their mental models and might not calculate or recalculate quickly enough to accomplish the necessary coordinating actions. In some cases, humans are hobbled by an actual or perceived need to obtain higher-level approvals for necessary changes.

This is where intelligent things might shine. They can and should perform the control-related actions much faster than humans. They're less likely to experience information overload or psychological and cognitive barriers to "overturning" past decisions to make a more appropriate decision for the situation at hand. At the same time, because of this incredibly fast tempo, humans will find it difficult to understand and trust the recommendations or actions of intelligent things during an agile adaptation.

Sensemaking

Making sense of the situation involves a number of iterative activities in the information, cognitive, and social domains. Sensemaking involves information seeking and analysis used to construct a story that represents our understanding of the situation and frames our consideration of how to respond. The components of sensemaking include obtaining relevant data, developing situation understanding, prediction, and decision making.

In the case of a team of humans and intelligent things (see Figure 2), humans and machine intelligence must be able to work effectively together to develop shared understanding of the situation.

Obtaining relevant data. Participants of a decision-making organization need to know what information is available, how to access it, and how to process and filter it. Gathering high-quality information to build situation awareness involves sifting through large amounts of data to select the most relevant (and timely) information, resolving conflicting information, and rejecting false information.

Humans are relatively slow in these processes and can be easily overwhelmed when the quantity of information is extremely large. Facing the possibility of information overload, humans often focus on sources and methods that they know well and trust. They have a tendency to collect, select, and prefer the information that supports preexisting biases. With information overload also comes increased error rate. Conflicting information can come from multiple nodes within the organization, and reconciliation can become a matter of debate, differing personalities, and office politics.

This is another area where intelligent things will shine. They can acquire and process information quickly, deal with very large volumes of information, and handle information in a consistent, rigorous, and unbiased way. Errors and omissions will be less likely, and conflicting data will be noticed. On the other hand, intelligent things rely largely on preexisting models and algorithms (although a degree of learning will be expected), and are less likely to show creativity or to detect clever deceptions that might be hidden in the data.

Situation understanding. Understanding a situation involves far more than developing a description of the situation. To support sensemaking, our understanding of a situation needs to include perceptions of cause and effect, as well as temporal dynamics (how the situation is likely to unfold or change over time).[2] As humans, we say that we understand something when the result seems reasonable to us, and we say that we don't understand something when the result is unexpected or is without a logical explanation.

Understanding resides in the cognitive domain and—like everything in the minds of humans—is subjective, influenced by perceptual filters and biases. However, one's understanding might not be "correct," meaning it doesn't conform to objective reality. What does "understanding" mean to intelligent things? It likely involves the alignment of available data with models residing in the intelligent thing's "brain," ensuring that observations of the environment can be explained by the available models of the environment. If so, intelligent things will be successful to the extent that a drastically novel model of reality is not required.

Prediction. Prediction requires more than understanding—even if one understands a phenomenon, one might not be able to predict the effects of that phenomenon. Prediction requires actionable knowledge, specifically the values of the variables that determine (or influence) the outcome in question. Operationally, the most that can be expected is to identify meaningfully different alternative futures and

Figure 2. A team of humans and intelligent things. The team must work together to make effective, distributed decisions in a dynamic, disorienting, dangerous environment. (Source: Tien Pham [concept] and Evan Jensen [art], US Army Research Laboratory; used with permission.)

indicators that those alternatives are becoming more or less likely over time.

Humans have a sophisticated base of experiential knowledge of what might happen in different situations. They have intuitive vision—the ability to project events forward. At the same time, each individual human member of a decision-making organization is limited in their knowledge. Alternative futures can be debated, and a formal analysis is often distrusted or is too difficult to apply. As always, personalities and biases play important roles.

Intelligent things will be far better at applying formal or computational models for predictions,[6] considering all pertinent details in an unbiased, exhaustive manner and drawing rigorous conclusions. However, they'll have a hard time explaining the rationale, the chain of reasoning, and articulating a compelling narrative that would illuminate their analysis of the situation.[7]

Decision making. A number of diverse processes are used in decision making: assess the pros and cons of the solutions, compare them, and articulate and defend the recommended solution. Too often, humans' assessments of solutions are subconsciously biased toward the desired features or priorities of the problem or the presumed preferences of a more senior decision maker. For humans, disagreements about alternative solutions and their potential outcomes can devolve into a battle of personalities.

Intelligent things, on the other hand, will rigorously explore a broad space of potential solutions and match them to the requirements of the problem in an unbiased way, albeit within the scope of available machine knowledge, models, and algorithms. If suitable analytical techniques exist, they'll be used to characterize the advantages and disadvantages of the proposed solutions. However, as we previously mentioned, an intelligent thing is likely to be challenged in explaining its chain of reasoning, to find "out-of-the-box" solutions, or to recognize and manage the human emotional and political aspects of alternative solutions.

Execution

Aside from taking physical actions or issuing orders for actions by subordinate entities, execution involves a complex process of coordinating details with other nodes within the decision-making network: coordinating actions in time and space, managing interdependencies between and among actions, and resolving conflicts.

This article originally appeared in Computer, *vol. 50, no. 12, 2017.*

Humans achieve effective coordination by utilizing a common understanding of goals and priorities (even if they're vaguely articulated); following established, trusted links; and negotiating if challenged by competition over priorities or resources. Humans are also capable of agile interpretations of orders if the right environment has been created.

These are skills that intelligent things have yet to develop. Things are not, at this point in their development, good at effective negotiation skills, especially when negotiating with humans. They're currently unable to understand unstated preferences and priorities of humans or to build trust and mutual dependence.

Situation monitoring

Situation monitoring focuses on ascertaining whether the actions being taken are producing the desired effects. In dynamic situations, situation monitoring also needs to recognize when circumstances have changed and whether these changes require altering the plan and whether the current approach to C2 remains appropriate. The same attributes of humans and intelligent things that affect their sensemaking abilities come into play here as well. As a result, humans will be slower and more reluctant to recognize when change is required, while being better able to think "out of the box."

THE C2 CHALLENGE

This transformation of the workforce promises to bolster our capabilities in a number of ways, perhaps most importantly by enhancing our agility. However, realizing these benefits will require developing appropriate ways to command this new organizational form. The fundamental challenge is one of C2 design. As more decisions and tasks migrate from humans to other intelligent entities, these entities need to be carefully integrated into our approaches to C2 in a manner that takes advantage of their unique qualities. Not doing this could lead to situations in which entities prevent one another from functioning as intended.

Best C2 practices call for the allocation of decision rights to individuals to be based upon not only their competence to make the decision, but also the ability to ensure that the decision maker can interact with others appropriately and has access to the required information. Adding other intelligent entities into the mix will require the explicit consideration of both the requirements of the decision task and the cognitive attributes of these entities. The C2 design challenge is to find the most appropriate allocation of decision tasks between and among humans and things for the mission and circumstances. Although there's a desire to delegate responsibilities to the lowest practical level,[8,9] given that trust is necessary to achieve this in practice, how will this apply to intelligent things?

Commanders or managers of mixed human–thing organizations will face several of the challenges we've highlighted here. Intelligent things will also be challenged in a number of areas and will need humans to help develop their capabilities, including the ability to explain; build trust; bond; understand personal agendas, emotions, and politics; and negotiate. Things and humans both can have difficulty anticipating and coping with the unusual and unexpected, and both can struggle with finding "out-of-the-box" solutions.

Welcome aboard, intelligent things. No matter what our respective shortcomings are, we'll be stronger and more agile when working together in decision-making organizations.

DISCLAIMER

This article does not reflect the positions or views of the authors' employers.

REFERENCES

1. A. Kott, A. Swami, and B.J. West, "The Internet of Battle Things," *Computer*, vol. 49, no. 12, pp. 70–75.
2. D.S. Alberts and R.E. Hayes, *Understanding Command and Control*, US Department of Defense Command and Control Research Program, 2006; www.dodccrp.org/files/Alberts_UC2.pdf.
3. *SAS-085 Final Report on C2 Agility*, tech. report STO-TR-SAS-085, NATO Research and Technology Organization, 2013; dodccrp-testorg.squarespace.com/sas-085.
4. A. Kott, D.S. Alberts, and C. Wang, "Will Cybersecurity Dictate the Outcome of Future Wars?," *Computer*, vol. 48, no. 12, 2015, pp. 98–101.
5. R. Want, T. Pering, and Y. Agarwal, "Multidevice Interaction," *Computer*, vol. 49, no. 12, 2016, pp. 16–20.
6. R. Rasch, A. Kott, and K.D. Forbus, "Incorporating AI into Military Decision Making: An Experiment," *IEEE Intelligent Systems*, vol. 18, no. 4, 2003, pp. 18–26.
7. D. Gunning, "Explainable Artificial Intelligence (XAI)," DARPA; www.darpa.mil/program/explainable-artificial-intelligence.
8. J.D. Sharpe Jr. and T.E. Creviston, "Understanding Mission Command," US Army, 10 Jul. 2013; www.army.mil/article/106872.
9. K.G. Stewart, "The Evolution of Command Approach," *Proc. 15th Int'l Command and Control Research and Technology Symp.* (ICCRTS 10), 2010; www.dodccrp.org/events/15th_iccrts_2010/papers/192.pdf.

ALEXANDER KOTT is the Chief Scientist at the US Army Research Laboratory. Contact him at alexander.kott1.civ@mail.mil.

DAVID S. ALBERTS is a Senior Fellow at the Institute for Defense Analysis, where he explores issues related to command and control in a networked environment. Contact him at dalberts@ida.org.

Editors: D. Balzarotti, davide.balzarotti@eurecom.fr | W. Enck, whenck@ncsu.edu | T. Holz, thorsten.holz@rub.de | A. Stavrou, astavrou@gmu.edu

Internet of Things Security Research:

A Rehash of Old Ideas or New Intellectual Challenges?

Earlence Fernandes, Amir Rahmati, Kevin Eykholt, and Atul Prakash | University of Michigan

Devices with computational and networking capabilities are enhancing our homes, hospitals, cities, and industries. This emerging network of connected devices—or Internet of Things (IoT)—promises better safety, enhanced management of patients, improved energy efficiency, and optimized manufacturing processes. Despite these many benefits, security vulnerabilities in these systems can lead to user dissatisfaction (for instance, random bugs), privacy violation (for instance, eavesdropping), monetary loss (for instance, denial-of-service attacks or ransomware), or even loss of life (for instance, attackers controlling vehicles[1]). Therefore, it is critical to secure this emerging technology revolution in a timely manner.

Although the research community has begun tackling challenges in securing the IoT, an often-asked question is: What are the new intellectual challenges in the science of security when we talk about the IoT, and what problems can we solve using currently known security techniques? This article summarizes some similarities and differences between IoT security research and classic IT security research.

We take a broad view of the IoT, touching on consumer-grade, industrial control systems and autonomous vehicles. Other IoT areas, such as smart cities, are outside the scope of this article. A whole set of privacy issues might arise from always-connected devices in the physical environment—this article doesn't go into depth on these challenges, but Nigel Davies and his colleagues discuss possible challenges and solutions in "Privacy Mediators: Helping IoT Cross the Chasm."[2] Our focus is on security and safety issues.

Similarities and Differences

We classify the similarities and differences based on the standard computing stack: hardware, system software, network, and application layer. The IoT computing stack is structured similarly:

- At the lowest layer are devices that can sense and effect physical change in the environment.
- The next layer comprises IoT platforms—software systems that aggregate multiple devices and controlling software to perform useful tasks.
- Next, various connectivity/network protocols enable software and physical devices to communicate with one another.
- Finally, the application layer runs custom code to control physical processes.

We discuss areas of similarities and differences next. We note that it is not our goal to be exhaustive in our discussion.

2469-7087/18/$33.00 © 2018 IEEE Published by the IEEE Computer Society

Hardware Layer

The hardware layer often forms a root of trust in modern computing systems, and we expect that hardware security research results developed in the context of desktop, mobile, and cloud systems will transfer in some form to IoT systems. We focus on two themes: security for hardware and hardware for security.

Security for hardware. Recent work has shown the possibility of hardware-level Trojans—malicious components or instruction sequences that, when triggered, circumvent security guarantees. Kaiyuan Yang and his colleagues recently showed how fabrication-time attackers can inject analog components that force a flip-flop, which maintains the processor's privilege bit, to a target value.[3] With a large percentage of IoT devices being manufactured by third parties (often overseas), hardware-level attacks are an increasing point of concern.

Given the relative simplicity of IoT devices, such as sensors and microcontrollers, in comparison to general-purpose computer processors, open questions are whether such attacks can remain stealthy, and whether postfabrication testing can be more effective in determining whether hardware Trojans exist in a chip.

Hardware for security. Galen Hunt and his colleagues recently discussed this topic in *The Seven Properties of Highly Secure Devices*. Two properties directly concern hardware security techniques: a hardware root of trust and hardware-supported software isolation.[4]

Although the ideas of using hardware mechanisms to securely store cryptographic keys (for example, trusted platform modules and one-time fuses) and to create isolation units (for example, memory management units and Intel Software Guard Extension enclaves) are similar to those in classic IT research, we envision many challenges arising in *applying* these notions of hardware security to IoT systems due to their limited computational and energy constraints.

These computational and energy limitations can affect higher-layer security primitives—some IoT devices might not have very precise real-time clocks, making it harder to implement even the most basic of network security protocols that assume the presence of reliable clocks. For example, Amir Rahmati and his colleagues showed how the natural decay rate of static RAM can be used as a timekeeper for embedded devices without clocks (for instance, smart cards).[5]

Hardware security research results developed in desktop, mobile, and cloud systems will likely transfer to IoT systems.

In general, we observe that although the core notions of creating hardware to support security primitives is similar to other computing paradigms, the computational and energy limitations at the hardware layer can impact security mechanisms at higher layers in the context of the IoT computing paradigm. We also observe that, conversely, higher-layer security properties might have to be tuned to the specific limitations of the IoT device through a hardware–software codesign approach.

System Software Layer

The system software layer consists of firmware, OS code, and any privileged system applications or programming frameworks. This layer builds on hardware mechanisms for establishing trust and isolation. We believe that many security principles developed in the context of mobile, desktop, and cloud computing will be applicable to IoT platforms—software systems that are similar in function to OSs for other computing paradigms. We discuss a few areas of similarities and differences, categorized by security principle.

Process isolation. Current OSs provide a basic primitive: a fault in one process doesn't affect other processes on the system. These isolation guarantees depend on the presence of a hardware memory management unit (MMU). In small IoT devices (for instance, devices with 64 Kbytes of RAM), such an MMU is generally absent. A challenge here is to support the classic notion of process isolation without an MMU. The Tock OS is currently exploring a combination of language-based isolation features and memory protection units to provide a process isolation abstraction.[6]

In general, although the notion of process isolation is well-known, enabling it for OSs of resource-constrained IoT devices can require new techniques, whereas enabling it for IoT devices with more resources, for example, Nest thermostats or Amazon Alexa, likely won't be a challenge.

Access control. OSs protect resources from untrusted code using access control. A piece of code is either given a token (as in a capability-based system) or assigned an unforgeable unique identity on which access control rules are expressed. Building an access control system for a particular domain is often challenging. Our prior work in analyzing consumer

IoT platforms revealed access control design errors as a security flaw.[7] We performed an empirical security analysis of the SmartThings platform and found that access control granularity wasn't designed appropriately, and it led to exploitable overprivilege. A fundamental reason for such granularity design errors in access control systems stems from the tension between usability and security. This tension has manifested itself before, in mobile OSs[8] and, before them, in desktop OSs.[9]

Although the notion of access control still applies to IoT platforms, there are new challenges in the usability aspect of designing such systems. For example, most prior access control systems dealt with virtual objects such as files and processes. In the IoT space, the objects of access control are physical devices and intuitive physical operations. An interesting challenge is how to exploit our natural intuitions about physical objects while designing an access control system for IoT platforms. For example, Earlence Fernandes and his colleagues recently discussed the notion of a user-perceived-risk-based access control system for IoT platforms.[7]

Information flow control. Access control is a gatekeeper—once the code obtains access to sensitive resources, access control doesn't provide any further protection. We analyzed a set of smart home platforms and found that current platforms use only access control. Information flow control (IFC) is a promising technique to control how untrusted code uses its access to sensitive resources.[10]

Although IFC isn't a new concept, as evidenced by the multitude of proposed systems for various domains, the challenge lies in applying it meaningfully to a specific domain. For example, FlowFence is a recent proposal for consumer IoT frameworks that enables a dataflow graph approach to IFC due to the structure of IoT apps.[11] Furthermore, the kinds of confidentiality properties for environments such as homes are well-studied; however, the kinds of integrity properties that we might need, which are arguably more important in the IoT, have been less well-studied.

Software updates. Updating software is a fundamental security practice to patch security bugs and include additional features once devices are deployed. For smartphones, PCs, and cloud services, updating software is a well-understood, secure, and common practice. However, for physical devices in the IoT, several challenges arise:

- Upgrading software might require a shutdown of the physical processes under control,[12] which could have an economic impact.
- Updates might require reverification of compliance policies for safety-critical devices in sensitive installations like factories and hospitals.
- Updates on computers in tertiary network functions (for instance, a business network) can have unintended effects on a physical process. A prominent example of a negative effect of this kind was the shutdown of a nuclear reactor due to a software update on a computer in the plant's business network.[13]
- Many IoT devices deployed in the field (such as in concrete bridges) can be difficult to physically access and might be intermittently powered (by harvesting power from vibrations). Updating the software on such intermittently powered devices is a challenge that classical computing systems generally don't face.
- IoT devices might not be updatable fundamentally because the manufacturers didn't build an update channel. In this case, we need to revisit our notion of a software update of the host (the device) and include notions of network-based patches.[14]

Although software updates for security are a well-understood concept, designing update systems for the IoT poses new challenges because of the unique properties of the physical processes that are under the control of software.

Authentication. Passwords are currently the most widely used mechanism to authenticate users to their IoT devices, platforms, and services. But, they are also a major point of concern because weak passwords are pervasive and have recently enabled large denial-of-service attacks from botnets.[15] Although there are lightweight techniques to obtain statistical estimations of password strength (github.com/dropbox/zxcvbn), weak passwords are still rampant. We don't view *enforcing* reasonable strength passwords (nondefault) as a technical difference from IT security, but rather we view it as a usability challenge. Some proposals suggest moving away from password-based authentication schemes.[4]

Open challenges in authenticating users to IoT devices include answering the following:

- Are activity-based biometrics (for instance, gait and heart-rate) a better alternative to passwords given that IoT devices interact with physical phenomena?
- IoT devices don't necessarily have classic I/O (for instance, no display in Google Home)—this can affect authentication schemes like passwords. Can we design authentication schemes of equivalent security for different interaction modalities?

Network Layer

As in a classic computing stack, the network layer in the IoT stack

enables devices and software to communicate with each other. However, different from classic networking, IoT networking is marked by a multitude of protocols and is generally populated with fixed-function devices. We elaborate how these differences results in new security challenges and mechanisms.

Connectivity protocol diversity. The network layer in the IoT is marked by various physical media and communication protocols. Part of this connectivity protocol diversity stems from the relative infancy of this technology, and part of it stems from the constraints imposed by devices or from the physical spaces that host these devices. For intermittently powered devices, short-range protocols like Bluetooth Low Energy (BLE) and near-field communication (NFC) are vital in conserving energy. For devices located in existing infrastructure, protocols like physical-line communications avoid expensive infrastructural costs. Similarly, visible-light communication can be useful because lights are ubiquitous in physical spaces. This protocol diversity disrupts the operation of network scanning—a fundamental security practice. We highlight this using a BLE port–scanning case study, described below.

In BLE, a rough analog of a TCP port is a service UUID (Universally Unique Identifier). A device can support multiple UUIDs that define the kinds of functionality it provides. There are UUIDs for fitness machines, heart monitors, and so on (see www.bluetooth.com/specifications/gatt/services). When a BLE device is disconnected, it sends out advertisements that can help controllers (or scanners) discover the device, and attempt connections. Advertisements contain rudimentary information, so connections are required to get a full list of the services a device supports. Therefore, for a scanner to work reliably, a device would have to be in a disconnected state as a BLE device accepts only a single connection for its services, unlike TCP ports, where multiple simultaneous connections can be serviced on the same port. This introduces randomness into the scanning process as the scanner will have to "try again" at a later point in time in the hope that the BLE device is in the disconnected state. Furthermore, if a BLE device is connected, it doesn't send advertisements, further complicating scanner operation. (Sophisticated scanners could try to jam existing connections to force them to drop.)

Challenges arise in adapting known security principles to make them work for the unique IoT computing paradigm.

Therefore, scanners for IoT protocols are currently very network specific and offer only limited coverage (BLE scanners will be useful only for BLE devices, but it's common for physical spaces such as a home to contain devices using different connectivity protocols). This contrasts starkly with the Internet in general, in which TCP/IP is a constant presence for online services where network scanning is typically used. Port scanning is further complicated in the consumer IoT space due to the practice of placing devices behind a hub or router. Network scanners situated outside such a network won't be able to conduct internal scans.

Because each protocol has its own notions of how two peers communicate with each other, it's unclear how network security practices such as port scanning translate to networks of devices that use various IoT protocols.

Repurposing networking technologies in unforeseen ways. Again, a common IoT system architecture for smart homes is to connect multiple devices to a hub. If all the home IoT devices use Wi-Fi as a connectivity protocol, then a Wi-Fi router can be a hub. This kind of configuration poses new security challenges that Wi-Fi wasn't designed to support. For example, it's very difficult to ensure that only a Wi-Fi-enabled presence detector affects a door lock. Such an isolation boundary is useful because there could be multiple devices on a network, some of which might be malicious or compromised through bugs. The isolation unit would serve as defense in depth against such a situation. Furthermore, as we discussed, some devices might not have update channels, necessitating other means of updates. A central hub like a Wi-Fi router is in a good position to apply updates in the form of filters for known malicious traffic patterns. Anna Simpson and her colleagues discuss the design of a Wi-Fi home hub that can perform such security functions.[16]

In the context of smart homes, we observe that hubs like Wi-Fi routers are being increasingly used to support IoT device networks. Adapting these hubs to natively support security properties such as isolation is an open challenge.

Anomaly detection in the network. As defense in depth, detecting misbehaving devices on the network is a common and well-deployed security practice in many computing areas. The main challenge in obtaining useful results from anomaly detectors is tuning them to produce a low number of errors—that

is, to minimize how often they either raise a flag for benign behavior or don't raise a flag for malicious behavior. This challenge arises due to the fundamental complexity of the devices we typically connect to a network—general-purpose computers like mobile phones, desktops, and servers. These devices perform multiple functions and lead to complicated network traces that make it difficult to characterize "normal" behavior. In contrast, IoT devices are simple and have a single purpose (that is, they have fixed functions). This can translate to simpler network dynamics and, hence, easier-to-model behaviors, ultimately leading to fewer errors in anomaly detectors. Recent work in the context of industrial control systems have yielded promising results—David Formby and his colleagues show how predictable network characteristics of relays and circuit breakers can be used to reliably fingerprint them.[17]

A physical process evolves as per physical laws in a generally predictable fashion. For example, a garage door of a certain mass takes a specific amount of time to close, and an oven of a certain volume heats up to a specific temperature in a predictable amount of time. We envision that models of these physical processes can be used to reduce the errors in anomaly detectors. In contrast, general-purpose computers, by definition, don't have well-defined behavior models when applications running on them are taken into account.

Application Layer

The application layer in the IoT is no different from other computing paradigms—it runs customized code for end-user scenarios. We consider two ways in which IoT application behavior can affect security.

Physical co-relations. Consider a simple if-this-then-that rule that closes a garage door after 9:00 p.m. If a speaker were placed in the vicinity of the motors controlling the door, it would record a specific acoustic pattern for a specific amount of time whenever the door closes. There is a natural physical co-relation between this acoustic pattern and the closing of the garage doors.

The natural co-relations between physical phenomena could act as feedback channels that IoT platforms could then use to approximately monitor physical processes for deviations from expected behavior. If deviations exist, then it would mean that a failure or security issue occurred.

Machine learning and control of physical processes. In recent years, machine learning (ML) and deep learning have found wide applicability to many computing domains—deep-learning robots can learn to grasp objects, and the Nest thermostat can learn and then control HVAC settings automatically. However, recent work has shown that deep-learning algorithms are susceptible to adversarial manipulations of their input—attackers can craft input that looks indistinguishable from benign input to humans, but can be interpreted in a completely different way by machines. For example, tampered images that are fed into a vision algorithm running on an autonomous vehicle can make the vehicle believe a stop sign was a yield sign, causing a possible crash at an intersection. Building robustness into ML algorithms against such attacks is an active area of research whose details are beyond this article's scope. We refer readers to "Towards the Science of Security and Privacy in Machine Learning" for a more thorough treatment of the topic.[18]

As more physical processes come under the control of ML algorithms, their vulnerabilities in adversarial settings will become pressing security and safety issues. Classic IT security has often applied ML to security problems (for instance, malware detection); however, only recently has work begun on securing the ML algorithms.

Broadly, classic IT security research and IoT security research share the basic secure software and hardware construction principles that have been developed in other computing paradigms. The differences form a spectrum of new intellectual challenges. On one end of this spectrum, challenges arise in applying and adapting known security principles to make them work for the unique challenges posed by the IoT computing paradigm. We believe that overcoming many of these challenges will involve a cross-layer codesign approach. For example, due to limited energy availability, hardware security mechanisms might need to be purpose-built depending on the specific higher-level security property we want to enforce—it's not possible to efficiently accommodate a one-size-fits-all security mechanism.

At the other end of the spectrum, the nature of both physical processes and IoT devices lend themselves to the construction of new security mechanisms. As discussed, natural co-relations between physical phenomena can be exploited to detect security and safety failures. Similarly, the predictability of physical processes is another avenue that can be used to detect anomalous events. Finally, introducing ideas from the control engineering world into IoT platform construction (for instance, specialized feedback loops) could lead to a safer and more secure IoT. ■

References

1. A. Greenberg, "Hackers Remotely Kill a Jeep on the Highway with Me in It," *WIRED*, 21 July 2015; www.wired.com/2015/07/hackers-remotely-kill-jeep-highway.

2. N. Davies et al., "Privacy Mediators: Helping IoT Cross the Chasm," *Proc. 17th Int'l Workshop Hot Topics in Mobile Computing* (Hot Mobile 16), 2016, pp. 39–44.
3. K. Yang et al., "A2: Analog Malicious Hardware," *IEEE Symp. Security and Privacy* (SP 16), 2016, pp. 18–37; dx.doi.org/10.1109/SP.2016.10.
4. G. Hunt, G. Letey, and E. Nightingale, *The Seven Properties of Highly Secure Devices*, tech. report MSR-TR-2017-16, Microsoft, 31 Mar. 2017; www.microsoft.com/en-us/research/publication/seven-properties-highly-secure-devices.
5. A. Rahmati et al., "Tardis: Time and Remanence Decay in SRAM to Implement Secure Protocols on Embedded Devices without Clocks," *21st USENIX Security Symp.* (USENIX Security 12), 2012, pp. 221–236; www.usenix.org/conference/usenixsecurity12/technical-sessions/presentation/rahmati.
6. A. Levy et al., "Ownership Is Theft: Experiences Building an Embedded OS in Rust," *Proc. 8th Workshop Programming Languages and Operating Systems* (PLOS 15), 2015, pp. 21–26; doi.acm.org/10.1145/2818302.2818306.
7. E. Fernandes, J. Jung, and A. Prakash, "Security Analysis of Emerging Smart Home Applications," *Proc. 37th IEEE Symp. Security and Privacy* (SP 16), 2016; doi:10.1109/SP.2016.44.
8. A.P. Felt et al., "Android Permissions: User Attention, Comprehension, and Behavior," *Proc. 8th Symp. Usable Privacy and Security* (SOUPS 12), 2012, pp. 3:1–3:14; doi.acm.org/10.1145/2335356.2335360.
9. E. Bertino et al., "Some Usability Considerations in Access Control Systems," *Proc. Symp. Usable Security and Privacy* (SOUPS 08), 2008; cups.cs.cmu.edu/soups/2008/USM/bertino.pdf.
10. E. Fernandes et al., "Security Implications of Permission Models in Smart-Home Application Frameworks," *IEEE Security & Privacy*, vol. 15, no. 2, 2017, pp. 24–30; dx.doi.org/10.1109/MSP.2017.43.
11. E. Fernandes et al., "FlowFence: Practical Data Protection for Emerging IoT Application Frameworks," *Proc. 25th USENIX Security Symposium* (USENIX Security 16), 2016; www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/fernandes.
12. A. Cardenas et al., "Challenges for Securing Cyber Physical Systems," *Proc. Workshop Future Directions in Cyber-Physical Systems Security*, Dept. Homeland Security, 2009; chess.eecs.berkeley.edu/pubs/601.html.
13. B. Krebs, "Cyber Incident Blamed for Nuclear Power Plant Shutdown," *Washington Post*, 5 June 2008; www.washingtonpost.com/wp-dyn/content/article/2008/06/05/AR2008060501958.html.
14. T. Yu et al., "Handling a Trillion (Unfixable) Flaws on a Billion Devices: Rethinking Network Security for the Internet-of-Things," *Proc. 14th ACM Workshop Hot Topics in Networks* (HotNets 14), 2015, pp. 5:1–5:7; doi.acm.org/10.1145/2834050.2834095.
15. B. Krebs, "Did the Mirai Botnet Really Take Liberia Offline?," Krebs on Security, 4 Nov. 2016; krebsonsecurity.com/tag/mirai-botnet.
16. A. Simpson et al., *Securing Vulnerable Home IoT Devices with an In-Hub Security Manager*, tech. report UW-CSE-17-01-01, Univ. Washington, Jan. 2017.
17. D. Formby et al., "Who's in Control of Your Control System? Device Fingerprinting for Cyber-Physical Systems," *Network and Distributed System Symp.* (NDSS 16), 2016; www.internetsociety.org/sites/default/files/blogs-media/who-control-your-control-system-device-fingerprinting-cyber-physical-systems.pdf.
18. N. Papernot, "Towards the Science of Security and Privacy in Machine Learning," *Computing Research Repository*, vol. abs/1611.03814, 2016; arxiv.org/abs/1611.03814.

Earlence Fernandes is a research associate at the University of Washington. At the time of this writing, he was a PhD candidate at the University of Michigan. Contact him at earlence@cs.washington.edu.

Amir Rahmati is a security research engineer at Samsung Research America. He will be joining the Computer Science Department at Stony Brook University in 2018. At the time of this writing, he was a PhD candidate at the University of Michigan. Contact him at amir@rahmati.com.

Kevin Eykholt is a PhD candidate at the University of Michigan. Contact him at keykholt@umich.edu.

Atul Prakash is a professor of computer science at the University of Michigan. Contact him at aprakash@umich.edu.

This article originally appeared in
IEEE Security & Privacy, *vol. 15, no. 4, 2017.*

myCS
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.
COMPUTER SOCIETY WEBSITE: www.computer.org
OMBUDSMAN: Direct unresolved complaints to ombudsman@computer.org.
CHAPTERS: Regular and student chapters worldwide provide the opportunity to interact with colleagues, hear technical experts, and serve the local professional community.
AVAILABLE INFORMATION: To check membership status, report an address change, or obtain more information on any of the following, email Customer Service at help@computer.org or call +1 714 821 8380 (international) or our toll-free number, +1 800 272 6657 (US):

- Membership applications
- Publications catalog
- Draft standards and order forms
- Technical committee list
- Technical committee application
- Chapter start-up procedures
- Student scholarship information
- Volunteer leaders/staff directory
- IEEE senior member grade application (requires 10 years practice and significant performance in five of those 10)

PUBLICATIONS AND ACTIVITIES
***Computer*:** The flagship publication of the IEEE Computer Society, *Computer*, publishes peer-reviewed technical content that covers all aspects of computer science, computer engineering, technology, and applications.
Periodicals: The society publishes 13 magazines, 19 transactions, and one letters. Refer to membership application or request information as noted above.
Conference Proceedings & Books: Conference Publishing Services publishes more than 275 titles every year.
Standards Working Groups: More than 150 groups produce IEEE standards used throughout the world.
Technical Committees: TCs provide professional interaction in more than 30 technical areas and directly influence computer engineering conferences and publications.
Conferences/Education: The society holds about 200 conferences each year and sponsors many educational activities, including computing science accreditation.
Certifications: The society offers two software developer credentials. For more information, visit www.computer.org/certification.

NEXT BOARD MEETING

7-8 June 2018, Phoenix, AZ, USA

EXECUTIVE COMMITTEE

President: Hironori Kasahara
President-Elect: Cecilia Metra; **Past President:** Jean-Luc Gaudiot; **First VP, Publication:** Gregory T. Byrd; **Second VP, Secretary:** Dennis J. Frailey; **VP, Member & Geographic Activities: Forrest Shull**; **VP, Professional & Educational Activities: Andy Chen**; **VP, Standards Activities:** Jon Rosdahl; **VP, Technical & Conference Activities:** Hausi Muller; **2018-2019 IEEE Division V Director: John Walz**; **2017-2018 IEEE Division VIII Director:** Dejan Milojicic; **2018 IEEE Division VIII Director-Elect:** Elizabeth L. Burd

BOARD OF GOVERNORS

Term Expiring 2018: Ann DeMarle, Sven Dietrich, Fred Douglis, Vladimir Getov, Bruce M. McMillin, Kunio Uchiyama, Stefano Zanero
Term Expiring 2019: Saurabh Bagchi, Leila DeFloriani, David S. Ebert, Jill I. Gostin, William Gropp, Sumi Helal, Avi Mendelson
Term Expiring 2020: Andy Chen, John D. Johnson, Sy-Yen Kuo, David Lomet, Dimitrios Serpanos, Forrest Shull, Hayato Yamana

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

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: James Jefferies
President-Elect: Jose M.F. Moura
Past President: Karen Bartleson
Secretary: William P. Walsh
Treasurer: Joseph V. Lillie
Director & President, IEEE-USA: Sandra "Candy" Robinson
Director & President, Standards Association: Forrest D. Wright
Director & VP, Educational Activities: Witold M. Kinsner
Director & VP, Membership and Geographic Activities: Martin Bastiaans
Director & VP, Publication Services and Products: Samir M. El-Ghazaly
Director & VP, Technical Activities: Susan "Kathy" Land
Director & Delegate Division V: John W. Walz
Director & Delegate Division VIII: Dejan Milojičić

IEEE

revised 7 Feb. 2018

Editors: Francis Sullivan, fran@super.org | Ernst Mucke, ernst.mucke@gmail.com

What Is the Blockchain?

Massimo Di Pierro | DePaul University

The technology known as the blockchain was first revealed by Satoshi Nakamoto in his paper "Bitcoin: A Peer-to-Peer Electronic Cash System" (https://bitcoin.org/bitcoin.pdf), which laid out the mathematical foundation for the bitcoin cryptocurrency. Although this was a groundbreaking paper, it was never actually submitted to a traditional peer-reviewed journal, and the author's true identity is unknown. Blockchain technology is not only at the foundation of all crytocurrencies, but it has found wide application in the more traditional financial industry. It also opened the door to new applications such as smart contracts.

It's a Matter of Trust

The problem that Nakamoto solved with the blockchain was that of establishing trust in a distributed system. More specifically, the problem of creating a distributed storage of timestamped documents where no party can tamper with the content of the data or the timestamps without detection.

Note that this problem is orthogonal to the problems of authentication, integrity, and nonrepudiation, which are solved by digital signatures. If a party creates a digital signature for a document, it establishes only a verifiable link between the party and the document. The existence of a valid digital signature proves that the party indeed intended to sign the document and that the document hasn't been altered. Yet the digital signature guarantees nothing about the time when the document was signed: the timestamp requires trust in the party that signed it. In the case of financial transactions and other forms of legal contracts, time is of the essence, and the order of those financial transactions needs to be independently certified to be auditable.

Consider the case of house sales. The owner can be defined as the party to whom the house was last sold to, but ownership can only be verified from the full paper trail of all transactions related to the house, a paper trail that's

 Published by the IEEE Computer Society 2469-7087/18/$33.00 © 2018 IEEE

usually kept and verified by title companies. Note this system doesn't completely prevent fraudulent transactions (such as a person selling a house that he or she doesn't own or selling the same property to more than one party), but fraudulent activities eventually get detected, and true ownership is established. The same ownership verification problem arises in financial transactions—for sure, in the sale of cryptocurrency, but also in the sale of any other traditional financial instrument. The problem is normally solved by recording all transactions in a single trusted centralized ledger, but a ledger isn't always a practical solution because it doesn't scale to large numbers of frequent transactions and because it requires all parties to trust the ledger's maintainer. In the same way you need to trust your bank with your money (and bank employees stealing customer funds is not unheard of). To address this, the blockchain provides a distributed trust mechanism: multiple parties keep a record of transactions, and every party can verify that the order and timestamps of the transactions haven't been tampered with.

A unit of bitcoin is nothing other than a number, but only some numbers are valid bitcoins. These numbers are solutions of a well-defined equation, and whoever finds a new solution owns it (this process is called mining). Once a bitcoin is discovered, it can be traded, with transactions stored in a ledger. Transactions are digitally signed with the credentials of the seller to avoid nonrepudiation. There is no centralized ledger because users wouldn't trust one and because there are too many transactions to store them all in one place. Hence bitcoin and other cryptocurrencies provide a distributed ledger in which every computer involved in the transaction of a specific coin (or fraction of a coin) keeps a copy of the history of that coin's transactions. The blockchain technology makes sure that no party storing this history can tamper with it without being detected.

Hash Functions

Transactions are units of data containing the transaction details plus a timestamp. Both can be represented as computer numbers or strings. A blockchain can be thought of as a table with three columns, where each row represents a distinct transaction, the first column stores the transaction's timestamp, the second column stores the transaction's details, and the third column stores a hash of the current transaction plus its details plus the hash of the previous transaction. When a new record is inserted into a blockchain, the last computed hash is broadcasted to every interested party. It isn't necessary for every party to keep a copy of the entire transaction history—it's sufficient that a few parties do. Because everyone knows the last hash, anyone can verify that the data hasn't been altered since it would be impossible without obtaining a different and thus invalid hash. The only way to tamper with the data while preserving the hash would be to find a collision in the data, and that's computationally impossible. It would require so much computing power that it's practically uneconomical.

A hash can be thought of as an encrypted version of the original string from which it is impossible to derive the original string. In fact, one way to compute the hash of a string is by encrypting it and performing some scrambling and xoring of the output bits. Mathematically, a hash is produced by a hash function, f, which must have two important properties: the size of the input space and the output space must be large; it must be practically impossible to find collisions, that is, two inputs x_1 and x_2 that produce the same output $f(x_1) = f(x_2)$. A typical application of hash functions is in password storage—when you register on a website, you don't want the site to store your password p in its database, otherwise anyone with access to the database could read it. The website should store the hash of the password, $f(p) = y$. When you login, the input password p is hashed again and compared with the stored value, $f(p) = y$. The probability of an incorrect password producing the same hash value y as the actual password is zero for practical purposes.

Examples of hash functions are the Secure Hash Algorithms (SHA1, SHA128, SHA512, and so on), which are implemented in the standard Python module `hashlib`. They can take any string as input and always produce an output string that's a hexadecimal representation of the output number of the function with a fixed number of digits:

```
>>> print hashlib.sha1('hello world').hexdigest()
2aae6c35c94fcfb415dbe95f408b9ce91ee846ed
```

Let's look at a simple implementation of a blockchain in Python. First, we define a function that we call *bhash* that, given the timestamp and details (a string or other serializable object) of a new transaction along with the hash of the previous transaction, computes a new hash using the SHA1 algorithm:

```
import hashlib, json, time

def bhash (timestamp, details, prev_hash):
    token = json.dumps([timestamp, details, prev_hash])
    return hashlib.sha1(details).hexdigest()
```

Notice that we used the json serializer to combine the elements together into a hashable string that we then pass to the hash SHA1 hash function. Our choice of serializing in json is an implementation detail and not the only way to achieve the goal.

Next we create a `Blockchain` class to encapsulate a list of blocks:

```
class Blockchain(object):
 def __init__(self, details='new-chain'):
     self.blocks = [(time.time(), details, '')]
 def record(self, details, timestamp = None):
     timestamp = timestamp or time.time()
     prev_hash = self.blocks[-1][2]
     new_hash = bhash(timestamp, details, prev_hash)
     self.blocks.append((timestamp, details, new_hash))
```

The class has a constructor, "`init`", which creates a list of blocks and stores the first block in the list. This first block contains an initial timestamp and details but no hash. In the case of a bitcoin, this would store information about the discovery of a new unit and its owner.

The class also has a second method, "`record`", that, given the details of a new transaction and an optional timestamp (otherwise automatically computed), stores them in a new block. This is done by retrieving the hash of the previous block from `self.blocks[-1][2]`, calling the `bhash` function, and appending the triplet (`timestamp`, `details`, `new_hash`) to the list of blocks. Notice that `self.blocks[i][j]` represents a cell in the blockchain table where `i` is the row number starting from 0, and `j` is the column number also starting from 0.

We use our `Blockchain` class by creating an instance of it, which we call "bc", and recording transactions represented as self-descriptive strings:

```
>>> bc = Blockchain('A found $1')
>>> bc.record('A gives $1 to B')
>>> bc.record('B gives $1 to C')
>>> bc.record('C gives $1 to D')
```

Then we can print the blocks in the blockchain:

```
>>> print bc.blocks
[(1495941516.704196, 'A found $1', ''),
 (1495941516.704201, 'A gives $1 to B', 'a75a9227f...'),
 (1495941516.704277, 'B gives $1 to C', 'ca911be27...'),
 (1495941516.704290, 'C gived $1 to D', 'cb462885e...')]
```

The last hash is `'cb462885e...'`. For this technology to work, we must make sure we broadcast the last hash and that there a few copies of the full chain stored by different parties. The parties in this context are the computing nodes in the peer-to-peer network in charge of recording and storing the transactions. This is a network problem and beyond this article's scope.

It's also important that every party can verify the chain's integrity. This can easily be done by using the function below:

```
def verify(blockchain):
    prev = blockchain.blocks[0]
    for block in blockchain.blocks[1:]:
        new_hash = bhash(block[0], block[1], prev[2])
        if block[2] != new_hash: return False
        prev = block
 return True
```

In the code, above we loop over all the blocks starting from the second one, recompute each hash, and then compare it with the stored one in `block[2]`, the third column. If the code finds any hash that doesn't match, it returns `False`, or else it returns `True`. We can call this code on our blockchain with

```
>>> print verify(bc)
True
```

From a technology viewpoint, there's a lot more than this to the bitcoin network. There are algorithms for data distribution, for syncing nodes, for efficient storage and querying, for conflict resolutions, and so on, yet the blockchain technology is at the heart of it.

Cryptocurrencies and Beyond

It's important to observe that different cryptocurrencies run on different platforms and make different storage and hashing choices. In addition, for the same type of cryptocurrency, for example, bitcoin, there are different implementations of the algorithm, even though they're all compatible and can communicate with each other. Moreover, for each unit of coin, there's one set of blocks (replicated in multiple locations).

Its use for cryptocurrencies is the first and best-known application of the blockchain, but it isn't the only one, and probably not the most important. Many companies provide proprietary implementations of the blockchain technology and sell their solutions to the financial industry, which uses them to record various types of transactions. These proprietary solutions are integrated into the authentication infrastructure of financial institutions and allow different agents to record transactions in a distributed fashion, thereby allowing different institutions (or parts of the same institution) to transact without reciprocal trust.

Because a transaction is basically a string, it can contain arbitrary information. It should be evident to the reader at this point that this technology can be used for any kind of notarization, and not necessarily involving money. For example,

Chicago's Cook County has been experimenting with using the bitcoin network to record house titles (https://bitcoinmagazine.com/articles/chicago-s-cook-county-to-test-bitcoin-blockchain-based-public-records-1475768860). Similarly, someone could store an idea for a patent in the blockchain to later prove a first-to-invent claim. You could also store a promise to do something at a later time, with the promise stored in the form of code that would execute the promise in an automated manner. This is what's called a smart contract; for example, let's say we have this promise: "Alice promises to pay Bob $1 if on 1 January 2028 it rains in Chicago." As long as the promise is in the blockchain, and an API can check whether the conditions are met, the system can automatically execute the transaction should the condition be fulfilled.

The bitcoin network was the first, but new ones are emerging all the time to trade and specifically handle smart contracts, "applications that run exactly as programmed without any possibility of downtime, censorship, fraud, or third-party interference" (thereum.com).

On one hand, the idea of trading cryptocurrencies might be nothing more than stamp collecting, but the other, the underlying technology has only started to revolutionize contracts and human interactions. It will displace many white collar jobs the same way robots have displaced blue collar ones. It will also create new jobs that we can't even imagine today. Only time will tell if cryptocurrencies can soar and prosper because of the increasing trust people put into blockchain technology. ■

Massimo Di Pierro is a professor in the School of Computing at DePaul University and co-director of the MS program in computational finance. Contact him at massimo.dipierro@depaul.edu.

This article originally appeared in Computing in Science & Engineering, *vol. 19, no. 5, 2017.*

Beyond Bitcoin: The Rise of Blockchain World

Roman Beck, IT University of Copenhagen

The brave new world of blockchain potentially transforms the financial structures we have come to know and feel ambivalent about. What does a decentralized, secure system mean for our society?

Bitcoin—a cryptocurrency built on blockchain technology—was the first currency not controlled by a single entity.[1] Initially known to a few nerds and criminals,[2] bitcoin is now involved in hundreds of thousands of transactions daily. Bitcoin has achieved values of more than US$15,000 per coin (at the end of 2017), and this rising value has attracted attention. For some, bitcoin is digital fool's gold. For others, its underlying blockchain technology heralds the dawn of a new digital era. Both views could be right.

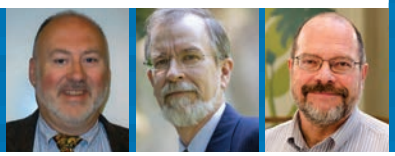
The fortunes of cryptocurrencies don't define blockchain. Indeed, the biggest effects of blockchain might lie beyond bitcoin, cryptocurrencies, or even the economy. Of course, the technical questions about blockchain have not all been answered. We still struggle to overcome the high levels of processing intensity and energy use. These questions will no doubt be confronted over time. If the technology fails, the future of blockchain will be different. In this article, I'll assume technical challenges will be solved, and although I'll cover some technical issues, these aren't the main focus of this paper.

In a 2015 article, "The Trust Machine," it was argued that the biggest effects of blockchain are on trust.[1] The article referred to public trust in economic institutions, that is, that such organizations and intermediaries will act as expected. When they don't, trust deteriorates. Trust in economic institutions hasn't recovered from the recession of 2008.[3] Technology can exacerbate distrust: online trades with distant counterparties can make it hard to settle disputes face to face. Trusted intermediaries can be hard to find, and that's where blockchain can play a part. Permanent record-keeping that can be sequentially updated but not erased creates visible footprints of all activities conducted on the chain. This reduces the uncertainty of alternative facts or truths, thus creating the "trust machine" *The Economist* describes. As trust changes, so too does governance.[4]

Vitalik Buterin of the Ethereum blockchain platform calls blockchain "a magic computer" to which anyone can upload self-executing programs.[5] All states of every

 Published by the IEEE Computer Society 2469-7087/18/$33.00 © 2018 IEEE

EDITORS
HAL BERGHEL University of Nevada, Las Vegas; hlb@computer.org
ROBERT N. CHARETTE ITABHI Corp.; rncharette@ieee.org
JOHN L. KING University of Michigan; jlking@umich.edu



program are publicly visible, with cryptographic guarantees that programs will execute as specified by the blockchain protocol. (Buterin later abandons the term *magic* in favor of *Turing-complete*.) Blockchain might, as the subtitle of this article suggests, usher in a new world. Some refer to blockchain as the most promising new technology since the Internet.[4] The gods of powerful institutions (for example, central banks), are challenged by blockchain. Whether this technology will force these gods into the twilight is unclear, but it's big enough and powerful enough to bring major changes.

WHAT IS BLOCKCHAIN?

Blockchain, as it is used today, is a tamper-resistant database of transactions consistent across a large number of nodes. The blockchain is cryptographically secured against retrospective manipulations, and it uses a consensus mechanism to keep the database consistent whenever new transactions need to be validated. Data storage on the blockchain is secured by cryptographic hashes in which data being hashed return a fingerprint that verifies the authenticity of the data. Alteration of the original data causes the hash of the altered data to no longer match the original fingerprint. Transactions on the blockchain are grouped and stored in blocks. The combined hash of these transactions is also stored, and each subsequent block saves the combined hash of the previous block. This creates a chain of cryptographically secured and linked blocks containing the information—the blockchain.

Any attempt to change information necessitates rehashing, not only the block relevant to the transaction, but all subsequent blocks. This is possible theoretically, but it's impractical since the blocks grow continuously as other nodes add blocks to the blockchain.[6] Technical details are summarized in a paper by Ethereum's Gavin Wood.[7] The Ethereum blockchain goes beyond bitcoin to allow user-created smart contracts executed on a generic, programmable blockchain under decentralized control, using a built-in Turing-complete programming language. This allows smart contracts and customized (even arbitrary) rules for ownership, transaction formats, and state transition functions. These smart contracts enable the distributed user community to resolve some issues without depending on trusted centralized authorities.

> Blockchain, as it is used today, is a tamper-resistant database of transactions consistent across a large number of nodes.

Blocks, hashing, trees, and miners

The foundation of blockchain is the security of code and data in the blocks. Bitcoin uses a "Merkle tree" to store data from new transactions with pointers to original block locations for unchanged data. Transactions are repeatedly paired, merged, hashed and rehashed until only one hash—the Merkle root—remains. Each subsequent block saves the Merkle root of the previous block. Ethereum blocks contain the entire state of the Ethereum system stored in a "Patricia tree," an evolved Merkle tree. Chained hashing keeps blocks well formed and difficult to tamper with. This helps keep the blockchain secure and almost unbreakable. A blockchain isn't run from a single server, but on a network of computers that hold all data and changes to the data in the blockchain. These computers are called "miners," essential to a blockchain that uses a proof-of-work mechanism to achieve consensus.[4] Proof-of-work is the most common consensus mechanism, used by both bitcoin and Ethereum and dating back to 1992.[8]

Proof-of-work mathematically ensures validity as long as no single entity holds enough computing power to add an illegitimate block to the blockchain. Each miner competes with other miners to earn the reward of being able to add a block to the blockchain. This is accomplished by the miner doing computationally intense work. Bitcoin requires the miner to find a string that, when concatenated with a hash of the previous block header and then re-hashed, returns a particular string. Anyone trying to "spoof" the blockchain (for example, to change data on old transactions) must recalculate the proof of work for all subsequent blocks. Convincing the system to use a bogus chain would require continuously adding blocks to the chain faster than a legitimate chain would evolve. Ethereum is developing an alternative consensus scheme that uses proof of stake that doesn't require the computational resources of proof of work, largely in response to processing intensity and energy use as noted earlier.

Each miner that joins the blockchain increases the level of decentralization, and also strengthens the consensus mechanisms. Transactions on decentralized blockchains are transparent and visible to users, in contrast to centralized systems where the users typically don't enjoy such transparency or trust in the provider.[9] Miners who have been able to solve the

cryptographic puzzle are rewarded, so miners continuously try to create the next blocks that can be added to the chain. No central authority decides this. Miners that try to add different blocks than those agreed on through the consensus mechanism are disregarded by the rest of the system. This forces uniformity in the blockchain. It's nearly impossible to cheat the blockchain without circumventing the consensus scheme that dictates nodal agreement that a miner has a right to be a block in a given blockchain.

Smart contracts

Security and transparency helps the blockchain provide a single version of what is the case and how that case was achieved—what some call "the truth." In this, bitcoin and Ethereum are similar. Ethereum goes beyond by permitting smart contracts, a piece of code that enables the Ethereum Virtual Machine (EVM) to execute on the blockchain. The EVM is similar to other virtual machines, compiling instructions from a programming language into low level code for the computer on which it runs. The EVM is a large decentralized computer containing millions of objects called "accounts." Accounts can maintain internal databases, execute code, and talk to other accounts. A smart contract is itself an account. The EVM allows for externally owned accounts (EOAs) controlled by a private key through a user, allowing an account to send ether and messages from the EOA.

A smart contract can't be altered once the code is set, although storage of the smart contract can be altered. The piece of code acts as an agreement, available for anyone to use. Smart contracts are made possible by the Turing-complete programming languages compiled into EVM bytecode. Smart contracts have addresses and execute code based on the data they receive. Smart contracts can call other smart contracts through messages. To avoid malicious behavior, infinite loops or distributed denial of service attacks, execution and creation of smart contracts uses Ethereum's internal cryptocurrency. The amount needed for a contract is determined by the computations and storage entries of bytecode that the EVM compiles the smart contract into. Specific computation costs are calculated by the complexity of the computation, with basic computations (addition, subtraction, and multiplication) costing less and more complications costing more. Miners are paid for use of their computational power. As of 2015, the computing power available on blockchains was small, about equivalent to a 1999 smartphone.[10] However, with powerful smart contracts this could change quickly.

Access to a blockchain is for transaction validation or transaction entry. Transaction validation depends on whether the blockchain is permissionless (all nodes can validate transactions) or permissioned (only preregistered can validate transactions). Transaction entry is available to all nodes in public blockchains. Only preregistered nodes can submit new transactions in private blockchains. Public blockchains can be either permissioned or permissionless.[11]

Blockchain's core ideas are well established: fidelity and transparency. Fidelity is the truthful rendering of the state of things. People trust those things are as represented. The technical structure of the blockchain is that blocks containing requisite information are secured cryptographically, and consensus mechanisms ensure that blocks along the chain agree with the creation of and/or change in the information to be held. Transparency is the ability of anyone to examine the entire record of changes to determine when, how and why changes were made. The architecture of blockchain is such that any effort to "hide" information on the chain is obvious, causing other users of the chain to ask questions about why it's happening. The technology doesn't guarantee that a blockchain cannot be corrupted, but it makes corruption difficult enough to generate trust.

BLOCKCHAIN AND TRUST

Trust is complicated and difficult to define precisely. It has numerous meanings and many different forms. Yet trust is the underlying fabric of human interactions, of central importance to interpersonal and interorganizational relationships. Blockchain affects trust. People sometimes refer to blockchain as a technology that overcomes the need for trust in human interactions. It's unclear that overcoming the need for trust is possible; rather, it's more productive to assess blockchain's effect on the antecedents of trust, including confidence, integrity, reliability, responsibility, and predictability. If we can be confident that collaborations will be executed as intended, and that there's only one version constituting truth, integrity is guaranteed. When contracts are executed as coded, blockchain is seen to be reliable. Roles and responsibilities are determined in advance, and outcomes are predictable. When these trust antecedents are handled effectively by blockchain, certainty can replace uncertainty. This is a major hope for blockchain; time will tell if it can be realized.

Decentralized and autonomous

Much is made of blockchain's decentralization and autonomy. However, nothing in blockchain requires decentralization or autonomy. Decentralization and autonomy are enabled by blockchain, but a choice can be made based on the needs of the application. Authorities, such as central banks, can adopt and apply blockchain technology; but blockchain provides an alternative that might have implications for control, authority, power, and so on. Beyond this, it might be possible to implement previously unavailable solutions when requirements for centralized authority are lifted. As formerly impractical solutions become practical, blockchain's impacts might go beyond "least expected" to "not expected at all."

It's useful to look at R.H. Coase's work;[12] he questioned why, in a market-oriented economy, economic activity

isn't limited to individuals interacting on markets? Why are there firms? Coase was an economist, but his ideas reach beyond economics. Firms emerged to handle "transactions" (searching, negotiating, monitoring, enforcing, coordinating) required by markets. In his model, when transaction costs are high, the firm emerges as more efficient than the market. The choice is between market and firm, but Coase recognized that a third "hybrid" form can emerge around collaborations, alliances, or joint ventures. These hybrids didn't conform to the products and services of the 1930s, and Coase didn't elaborate on them.

Friction costs

In principle, blockchain allows for such hybrids, enabled by its decentralized mechanisms to make claims, attest to things, or enforce rights (such as property rights). Blockchain enables trust that a transaction will be completed even if there are slight variances in protocol, because it's possible to see that the ends are achieved. It can reduce friction that comprises all kinds of direct and indirect costs and efforts due to the lack of trust and bring certainty via transaction logic instantiated as code. Contracts and other forms of agreements can be electronically executed without trust-associated friction costs. Blockchain can be used for transparent and secure transactions between and among individuals, individuals and organizations, and organizations.

Blockchain might alleviate our dependency on central, hierarchical organizing and planning—previously the only way to reliably handle financial transactions—and thus allow for decentralized enforcement of transactions, in a manner similar to the way the Internet enabled changes in social relationships, commerce, and so on. The constraints that now lead to centralized solutions might evaporate if the transaction logic can be orchestrated and enforced without that central authority. Blockchain can generate real-time information flows of transactions to allow new approaches of digital auditing to ensure agreements are honored. This paradigm shift suggests that such systems organize transactions reliably—possibly without human interaction—following a protocol. It's akin to unstaffed, autonomously navigating vessels safely moving passengers from A to B using a protocol capable of minimizing exceptions (malicious and accidental) and getting humans out of the loop. In principle, blockchain could be an Internet of Things backbone, enabling tamper-proof coordination of activities, for example between delivery drones and their delivery stations.

Whether any given application is controlled in a centralized or decentralized manner becomes a matter of choice; it doesn't default to the centralized approach because that's the only way to do it. It's less important for requiring a particular solution than for enabling multiple solutions, thereby increasing the options of those who pay for, design, use, or otherwise interact with blockchain applications.

BLOCKCHAIN WORLD

The emerging blockchain world is the combination of traditional ways of doing things and those that are enabled by blockchain. Third parties might still ensure trustworthiness, but they don't have to do so, nor do those who seek assurance have to depend on third parties. A transaction might be conducted as agreed upon solely because blockchain enables those interested to monitor the status of the transaction, know what's going on, and remind others of their obligations. Or a party could turn to another system (such as the criminal justice system) for enforcement. Blockchain could support many codified agreements handled by traditional means, including stock trades, monitoring contract, managing land records, security of foodstuffs, preserving provenance, and maintaining the chain of custody. In this way, the technology will become part of the infrastructure of daily life, affecting commerce, social interaction, law, education, entertainment, nutrition, livelihood, housing, and so on.

Just because blockchain world is part of a larger infrastructure doesn't mean its effects are trivial. The Internet has had a profound effect on our culture, economy, and systems, and blockchain will be complementary to the kinds of changes that have already transpired, providing the means for decentralized governance in addition to centralized governance. By enabling so-called decentralized autonomous organizations (DAOs), blockchain empowers participants to implement agreements and transactions, without being their own legal entity. DAOs make transactions transparent to DAO members, which in turn makes fraudulent behavior difficult to hide.

The emerging blockchain world is the combination of traditional ways of doing things and those that are enabled by blockchain.

In principle, a DAO can run autonomously as a decentralized, transparent, and secure system for operation and governance among independent participants. Blockchains needn't be controlled by any of the participants as it is serving as a trusted third party to provide the role of proxy and enforcement of rules. To use Coase's insight, a DAO might reduce transaction costs while providing setup, maintenance, regulation, and supervision like traditional third parties. The results

This article originally appeared in Computer, *vol. 51, no. 2, 2018.*

wouldn't be trust-free, but would shift from trust in a counterparty or a third party to the blockchain system itself and the rules coded therein.

One might say the DAO is a shift from a socio-technical system to a techno-social system. Socio-technical systems handle control of transactions through social systems. Techno-social systems handle control of transactions through technical systems that can be autonomous.[13] How this would work exactly is as yet unclear in many ways, but through blockchain technology we have the chance to experiment with secure, decentralized systems, which could enable new social models that go well beyond the economy.

Realizing these blockchain-enabled models will require workers possessing process and management knowledge, as well as information technology skills including programming, design, and an ability to see the big picture. Blockchain world promises much, though many of the details are still being determined.

REFERENCES

1. "The Trust Machine: The Technology behind Bitcoin Could Transform How the Economy Works," The Economist, 31 Oct. 2015; www.economist.com/news/leaders/21677198-technology-behind-bitcoincould-transform-how-economy-works-trust-machine.
2. S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008; bitcoin.org/bitcoin.pdf.
3. "2016 Edelman Trust Barometer," annual report, Edelman, 2016; www.edelman.com/insights/intellectual-property/2016-edelman-trust-barometer.
4. W. Mougayar, *The Business Blockchain: Promise, Practice, and Application of the Next Internet Technology*, Wiley, 2016.
5. V. Buterin, "Ethereum," white paper, 2013; github.com/ethereum/wiki/wiki/White-Paper.
6. S. Underwood, "Blockchain beyond Bitcoin," *Comm. ACM*, 2016, vol. 59, no. 11, pp. 15–17.
7. G. Wood, "Ethereum Yellow Paper," website, 2014; http://gavwood.com/paper.pdf.
8. C. Dwork and M. Naor, "Pricing via Processing or Combatting Junk Mail," *Proc. Ann. Int'l Cryptology Conf.*, 1992, pp. 139–147. Springer, Berlin, Heidelberg.
9. P. De Filippi, "The Interplay between Decentralization and Privacy: The Case of Blockchain Technologies," *J. Peer Production*, 2016; peerproduction.net/issues/issue-9-alternative-internets/peer-reviewed-papers/the-interplay-between-decentralization-and-privacy-the-case-of-blockchain-technologies.
10. V. Buterin, "Ethereum Development Tutorial," 2015; github.com/ethereum/wiki/wiki/Ethereum-Development-Tutorial
11. G.W. Peters and E. Panayi, "Understanding Modern Banking Ledgers through Blockchain Technologies: Future of Transaction Processing and Smart Contracts on the Internet of Money," *Banking beyond Banks and Money*, P. Tasca, T. Aste, L. Pelizzon, and N. Perony eds., 2016, Springer, Cham, pp. 239–278.
12. R.H. Coase, "The Nature of the Firm," *Economica*, vol. 4, no. 16, 1937, pp. 386–405.
13. J.M. Quintana Diaz, "The Merger of Cryptography and Finance--Do Cryptographic Economic Systems Lead to the Future of Money and Payments?," 2014; available at SSRN: ssrn.com/abstract=2536876.

ROMAN BECK is at IT University of Copenhagen. Contact him at romb@itu.dk.

Editor: **Cesare Pautasso**
University of Lugano
c.pautasso@ieee.org

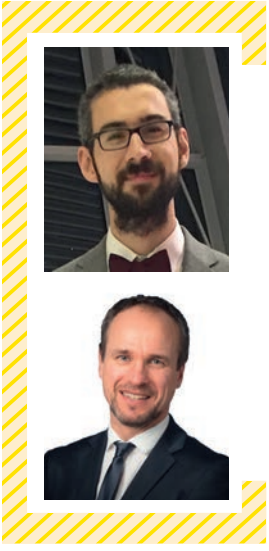
Editor: **Olaf Zimmermann**
University of Applied Sciences
of Eastern Switzerland, Rapperswil
ozimmerm@hsr.ch

Toward Evidence-Based Software Engineering

Lessons Learned in Healthcare Application Development

Artur Nowak and Holger J. Schünemann

Of all the domains of human activity, healthcare arguably has the most robust decision-making tools. This is no wonder because errors are particularly costly from both the human and the monetary perspective. In this installment of Insights, Artur Nowak and Holger Schünemann look back at the decisions made when designing, implementing, and evolving a collaboration tool to support evidence-based decisions in healthcare, and they reflect on how software engineers could benefit from similar methods. —*Cesare Pautasso and Olaf Zimmermann*



DECISIONS SHOULD BE based on the best available evidence, particularly when people's lives are affected. The GRADEpro Guideline Development Tool (GRADEpro GDT; gradepro.org) is a browser-based application that supports the whole cycle of creation of healthcare recommendations. It supports *evidence-based medicine* (EBM), in which decision making is grounded in the findings of research studies, mostly randomized controlled trials or observational studies.[1]

Such data is collected through systematic literature reviews. Well-performed systematic reviews follow a set of transparent methods to describe the state of the art of research on a specific topic. They are achieved by casting the net wide and then painstakingly filtering and analyzing the findings. The result represents the foundation of appropriate healthcare decision making. EBM derivations find their applications in different domains, including management, education, and software engineering.[2]

GRADEpro GDT users include medical professionals from around the world. For some of them, it is indispensable. For the others, especially busy clinicians, it is a mechanism to share their knowledge in a time-efficient way. Consumers of the tool's output are policy

2469-7087/18/$33.00 © 2018 IEEE Published by the IEEE Computer Society April 2018

makers, healthcare professionals, and patients. (For a typical output, see the European Commission's "Recommendations on Breast Cancer Screening."[3])

GRADEpro GDT was conceived as a reboot of a Windows-only application, extending it with numerous modules. As such, it inherited the title of the official tool of the GRADE Working Group (GRADE WG; gradeworkinggroup.org), an informal association of scientists that sets standards for appraisal of medical evidence. (GRADE stands for Grading of Recommendations Assessment, Development and Evaluation.) The methodology the GRADE WG created is widely acknowledged—for example, by the World Health Organization and Cochrane, the biggest nonprofit organization conducting systematic reviews.

Software architecture has been defined partly as "the set of significant decisions about … the organization of a software system."[4] Because decision support is a key capability of GRADEpro GDT, it makes sense to look at the decisions we made when designing GRADEpro from the perspective of the process the tool itself supports.

Decision Making in the Context of Software Architecture

Numerous decision-making approaches exist to balance design tradeoffs and evaluate commercial off-the-shelf (COTS) solutions. For instance, Davide Falessi and his colleagues described 15 such approaches and a framework for comparing and selecting them.[5] Here, we do not attempt to adapt the GRADE methodology to software engineering, although even a cursory look at the aforementioned framework suggests that such an adaptation would differ significantly from the existing approaches. Instead, we present the lessons learned from the development of GRADEpro, taking into account the perspective of our tool's users.

Perhaps the most striking difference between healthcare and software engineering is the historical scarcity of experimental data in the latter domain. A study of 5,453 papers published from 1993 to 2002 in nine journals and three conference proceedings (focused on the subject of empirical evidence) identified only 103 papers describing controlled experiments.[6]

However, the 21st century has seen increased awareness of controlled experiments' importance among both researchers and practitioners. A recent study on one experiment type alone reported that 82 of the 930 analyzed papers included empirical data.[7] This growth also has been reflected by

- the introduction of *empirical software engineering* courses into several universities' curricula (for an example, see goo.gl/NYPJ92),
- establishment of a dedicated journal (*Empirical Software Engineering*; www.springer.com/computer/swe/journal/10664), and
- publication of handbooks on the subject (for example, *A Handbook of Software and Systems Engineering: Empirical Observations, Laws and Theories*[8]).

We have also seen the rise of systematic literature reviews and mapping studies in the field. The significance of good reporting, which enables generalization of findings, has come to the attention of authors of nonexperimental empirical papers, such as case studies.[9] The emergence of public code-hosting services (for example, SourceForge and GitHub) has led to the concentration of open source efforts around these sites. This in turn has enabled many comparative studies based on mining this data. Similarly, big data initiatives in enterprises will likely extend to the aggregation of metrics on software production processes, leading to the collection of data that can be turned into insights. All these factors strengthen our belief that high-quality empirical evidence will become much more available in the coming years, leading to more widespread adoption of *evidence-based software engineering*.

Many software engineering experiments have studied how programming techniques impact worker performance, resembling studies in industrial and organizational psychology. Additionally, especially in the context of COTS evaluation, many objective metrics can be defined, such as a solution's memory footprint or the number of LinkedIn profiles that match a given technology (as a proxy for ease of hiring).

Likewise, health evidence comes from different sources and takes many shapes and forms. For example, randomized controlled trials measure an outcome (for instance, bleeding as an adverse effect of a blood thinner given to prevent strokes) in two randomly selected groups. In contrast, case control studies divide the population by the outcome's occurrence and seek differences in the prior events. These studies leave the users of the research with different degrees of certainty of the evidence. The reasons for the degrees of evidence are based on trained individuals' judgments that should be transparent, reproducible, and storable.

A look at the GRADEpro modules (team formation, conflict-of-interest management, scope definition, evidence synthesis, a structured decision process involving a panel of experts, and dissemination of the results) suggests that the same principles hold for decision making in software engineering. For example, you must always formulate questions before you apply analysis, to avoid the fallacy of *data dredging*: mining a dataset for potential correlations without testing the hypotheses on a holdout set. The recommendations should be made by people without conflicts of interest, including intellectual conflicts of interest; cognitive bias should be minimal. It is crucial to estimate not only the effect or association size but also your certainty or confidence in it and how it supports a particular recommendation. Ultimately, all investments involve both returns and risk levels.

The same goes for your certainty regarding the relative value of various system traits. Maybe a user survey indicated that a visual revamp of the user interface is important, but how representative are those users of the overall population? Perhaps self-selection of the participants occurred—especially if you titled the survey "Help Us Make the Software Look Better." Maybe only a minority of the stakeholders highly value a more ergonomic interface for data entry—not surprisingly, the people who will use it daily. In other words, you must consider not only the responses' mean value but also their variability.

Health intervention recommendations suggest whether to use one option or another in some population (for example, diabetics). Similarly, in software engineering, you are considering whether a team should use one or another solution or technique to address a given requirement. The attributes you use to perform the comparisons (bleeding, in our former example) are the time to implement the software solution, the amount of training required, and so on.

Different perceptions of the outcomes' meanings among stakeholders can lead to serious problems in both healthcare and software engineering. The understanding of terms such as "performance" or "complexity" might vary greatly among team members.[5] To address this issue, GRADEpro introduces a *marker states* database—a catalog of definitions of such terms. For any given project, the stakeholders (patients, physicians, policy makers, and so on) collaboratively refine the definitions to create a common understanding of what, for instance, "severe pain" means in the given context. We can view this activity as a software-supported, formalized process for creating the Business Glossary artifact in the Rational Unified Process or the ubiquitous language that domain-driven design, for example, calls for.[10]

A Guideline Isn't a Cookbook

A common misconception is that EBM uses evidence to formulate "cookbooks" for clinicians to use verbatim. On the contrary, it is all about "integrating individual clinical expertise and the best external evidence."[1] Combining the two knowledge sources is by far the most difficult part of the process.

The growing availability of experimental data in software engineering might lead to neglecting the value of professional judgment. However, assessing a study finding's applicability to an individual patient (or software system) requires years of training and experience. To make these judgments reusable in different contexts, GRADEpro breaks decision making into discrete steps (such as the assessment of problem priority, the desirable and undesirable anticipated effects, and the cost-effectiveness) for which choices and justifications are recorded.

In our own case, GRADEpro GDT was to be browser-based, to not only make it cross-platform but also allow its use in contexts in which installation is not possible (for example, virtualized, thin workstations in some IT environments). At the same time, it had to work offline, to operate, for instance, on airplanes or in areas with weak Internet connectivity (it's used in 150 countries). It also had to support group work and data sharing between users, preferably in nearly real-time, Google Doc-like fashion. Moreover, a full history of each record had to be kept for transparency of the process. Because some of the stakeholder organizations require full control over their data, we also planned on-premises installations, with the possibility of securely exchanging data with the central server.

All these requirements translate to effects we try to predict for every proposed solution: What will be the resulting difficulty of on-premises deployments? What about this functionality's delivery time? Clearly, different stakeholders faced with such tradeoffs will have different *values and preferences* (to use GRADEpro language). For health guidelines, meticulous recording of these choices enables adaptation to different clinical contexts (such as populations with the distinct prevalence of a condition or health systems with limited resources). Leveraging similar, structured formats of documenting design decisions might

lead to greater reusability of architectures (beyond design patterns), internally in the organizations and across entities.

The resulting balance of effects depends on both the aforementioned values and the magnitudes of individual effects (for example, increased complexity). Our stakeholders emphasized having a browser-based but offline-first application. We determined that the complexity of dealing with online or offline states in the classic request–response cycle would be unbearable in the long term. We aimed for a solution that maximizes simplicity in this regard at the expense of other areas. For example, we knew that some options with this characteristic would increase the operational burden in the on-premises deployment scenario owing to the use of uncommon server software and need for precise tuning. However, our evaluation showed that employing IT automation tools (such as Chef) can keep the burden manageable.

So, the most important decision probably was to move the database to the browser and write the (thick) client app with the assumption that all the data is always available locally. Of course, some features cannot work fully offline—for example, sharing a project with other users or sending a message requires an Internet connection—but they can be scheduled while offline. This in turn meant that any server-side processing must be message-driven because we cannot guarantee that the client will be able to reach an API endpoint. We coined the term "worker" for microservices that listen for changes in the data and perform operations synchronously on the central database.

Although our experience shows that this choice kept the synchronization mechanism's complexity at bay, someone might argue that the resulting architecture is unfamiliar (simple, but not easy[11]). For example, all the effects caused by the workers must be idempotent to guarantee both failure tolerance and correct concurrency control. Compared to the request–response style, this resulted in more time needed to design new functionalities and more effort on code reviews and training.

The tool must support old versions of the business logic—the creation of medical guidelines often spans years, and a consistent version of the process is necessary. The requirement to support legacy documents and be able to render and edit them using the previous version of the business logic was easier to satisfy using semistructured data. Our investigation of document-oriented databases showed that Apache CouchDB (couchdb.apache.org) has a robust, easy-to-implement replication mechanism that is up to the task of moving the data between the server and client.

The rationale behind this choice highlights another aspect we had to take into account when considering external evidence: How different from our problem was the problem the authors of that evidence had solved? No experimental studies existed on using document-oriented databases, but EBM calls for using the best available evidence—in our case, blog posts describing unsuccessful projects.[12,13] We faced the *publication bias*—people are more likely to investigate and publish "noteworthy" (in this case, negative) conclusions.

Document stores are hardly a golden hammer, but there is a specific domain of problems (for example, reads are far more frequent than writes) in which this approach really shines. At the time, NoSQL (not only SQL) solutions were gaining popularity mainly on the basis of ease of scaling. Every rose has its thorns; here, data redundancy was probably the biggest one.

Luckily, the initial engineering team (two developers) had previously helped develop an art collection management system based on a homegrown document-oriented database. From this work, we were familiar with the downsides of this approach and were on the lookout for the potential traps. This brings us to another source of indirectness: How did the people in the previous studies differ from our team? Perhaps the techniques were evaluated on developers with little previous experience?[14]

To borrow another healthcare term, we need to emphasize *ease of diagnosis* more in our systems. In particular, even today, the browser is still a hostile environment to work in. Even if standards compliance has greatly improved (although it is still far from perfect), the number of extensions (and sometimes malware) that interfere with our code results in cryptic errors. So, including high-quality client-side error reporting from the start (using Sentry; getsentry.com) has been extremely valuable. However, this advice applies to all areas of development. For example, it is relatively inexpensive to evaluate stack trace readability when you are selecting a web framework.

It is also important to estimate your choices' cost-effectiveness (and your certainty in this regard), instead of just the upfront cost. For example, you need to consider the cost of open source.[15] In the case of the core libraries, the team must either accumulate knowledge at the near-contributor level or be able to hire contributors. The rate of change in some projects might be overwhelming, forcing you to stick to an old, unsupported version. So, it's beneficial to monitor the libraries' footprint and at least skim through the source code to evaluate how difficult that code would be to fix if bugs occurred (this is, in the end, open source's great advantage). Otherwise, the changes in the transitive dependencies can have a snowball effect. We have observed increased awareness of these problems among the JavaScript community, whose members have recently started to praise libraries with few dependencies.

When balancing all these factors, we must not forget about development's social aspects. To paraphrase the Agile Manifesto's first principle, we need to value developers over frameworks. Forming a strong, versatile team with diverse experience and a skillset that matches the requirements is crucial for any project's success. Promoting technology-agnostic architectures such as microservices increases developers' freedom of choice, leading to increased acceptability of the solution.

References

1. D. Sackett et al., "Evidence Based Medicine: What It Is and What It Isn't," *BMJ*, vol. 312, no. 7023, 1996, p. 71.
2. T. Dyba, B.A. Kitchenham, and M. Jorgensen, "Evidence-Based Software Engineering for Practitioners," *IEEE Software*, vol. 22, no. 1, 2005, pp. 58–65; ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1377125&isnumber=30054.
3. "Recommendations on Breast Cancer Screening," European Commission, 2017; ecibc.jrc.ec.europa.eu/recommendations.
4. P. Kruchten, *The Rational Unified Process: An Introduction*, 3rd ed., Addison-Wesley Professional, 2003.
5. D. Falessi et al., "Decision-Making Techniques for Software Architecture Design: A Comparative Survey," *ACM Computing Surveys*, vol. 43, no. 4, 2011, article 33; dx.doi.org/10.1145/1978802.1978812.
6. D.I.K. Sjoeberg et al., "A Survey of Controlled Experiments in Software Engineering," *IEEE Trans. Software Eng.*, vol. 31, no. 9, 2005, pp. 733–753; doi:10.1109/TSE.2005.97.
7. S. Vegas, C. Apa, and N. Juristo, "Crossover Designs in Software Engineering Experiments: Benefits and Perils," *IEEE Trans. Software Eng.*, vol. 42, no. 2, 2016, pp. 120–135.
8. A. Endres and H.D. Rombach, *A Handbook of Software and Systems Engineering: Empirical Observations, Laws and Theories*, Addison-Wesley, 2003.
9. M. Shaw, "Writing Good Software Engineering Research Papers," *Proc. 25th Int'l Conf. Software Eng.* (ICSE 03), 2003, pp. 726–736; doi:10.1109/ICSE.2003.1201262.
10. E. Evans, *Domain-Driven Design: Tackling Complexity in the Heart of Software*, Addison-Wesley Professional, 2003.
11. R. Hickey, "Simple Made Easy," 2011; www.infoq.com/presentations/Simple-Made-Easy.
12. "Canonical Dropping CouchDB from Ubuntu One," *The H Open*, 22 Nov. 2011; www.h-online.com/open/news/item/Canonical-dropping-CouchDB-from-Ubuntu-One-1382809.html.
13. S. Mei, "Why You Should Never Use MongoDB," blog; www.sarahmei.com/blog/2013/11/11/why-you-should-never-use-mongodb.
14. I. Salman, A.T. Misirli, and N. Juristo, "Are Students Representatives of Professionals in Software Engineering Experiments?," *Proc. 37th IEEE Int'l Conf. Software Eng.* (ICSE 15), 2015, pp. 666–676; doi:10.1109/ICSE.2015.82.
15. C. Ebert, "How Open Source Tools Can Benefit Industry," *IEEE Software*, vol. 26, no. 2, 2009, pp. 50–51; ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4786952.

ARTUR NOWAK is the chief technology officer of Evidence Prime. Contact him at artur.nowak@evidenceprime.com.

HOLGER J. SCHÜNEMANN is the chair of McMaster University's Department of Health Research Methods, Evidence, and Impact, widely considered the birthplace of evidence-based medicine and problem-based learning. Contact him at schuneh@mcmaster.ca.

This article originally appeared in IEEE Software, *vol. 34, no. 5, 2017.*

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

Editor: **Daniel B. Neill,** H.J. Heinz III College, Carnegie Mellon University, neill@cs.cmu.edu

Graph Structure Learning from Unlabeled Data for Early Outbreak Detection

Sriram Somanchi, *University of Notre Dame*
Daniel B. Neill, *Carnegie Mellon University*

Event detection in massive datasets has applications to multiple domains, such as information diffusion or detecting disease outbreaks. In many of these domains, the data has an underlying graph or network structure: for example, an outbreak might spread via person-to-person contact. In the typical, graph-based event detection problem, we are given a graph structure $G = (V, E)$ and a time series of observed counts for each graph node v_i, and must detect connected subgraphs in which the recently observed counts are significantly higher than expected. Assuming that the graph structure is known, we can use various graph-based event detection methods to detect anomalous subgraphs.[1–3] A standard approach is to maximize a log-likelihood ratio statistic $F(S) = \log\left(\Pr\left(Data|H_1(S)\right)/\Pr\left(Data|H_0\right)\right)$ over connected subgraphs S. For example, we can compute the expectation-based Poisson scan statistic,[4] which assumes Poisson-distributed case counts and a uniform multiplicative increase over the affected subgraph, as $F(S) = \left(C\log\left(C/B\right) + B - C\right)1\{C > B\}$, in which the observed and expected counts are aggregated over subgraph S and denoted as C and B, respectively. Maximizing $F(S)$ over connected subgraphs is computationally challenging, but the GraphScan algorithm[3] can optimize $F(S)$ efficiently and exactly, scaling to graphs an order of magnitude larger than the previously proposed FlexScan approach[2] while outperforming heuristic approaches such as upper-level sets.[1]

In many cases, however, the network structure is unknown. For example, the spread of disease may be influenced by latent commuting patterns. Assuming an incorrect graph structure can result in less timely and less accurate event detection, because the affected area may be disconnected and therefore may not be identified as an anomalous subgraph. In such cases, learning the correct graph structure has the potential to dramatically improve detection performance. Thus, our goal is to learn a graph structure that minimizes detection time and maximizes accuracy when used as an input for event detection.

Several recent methods learn an underlying graph structure using labeled training data.[5–7] However, in many cases, labeled data is unavailable: for example, public health officials might be aware that an outbreak has occurred but might not know precisely which areas were affected and when. Hence, we focus on learning graph structure from unlabeled data, in which the affected subset of nodes for each training example is not given, and we observe only the observed and expected counts at each node.

Graph Learning Framework

Our framework for graph learning takes as input a set of training examples $\{D_1, \ldots, D_J\}$ assumed to be independently drawn from some distribution $\boldsymbol{D}$. Each example D_j represents a different snapshot of the data when an event is assumed to be occurring in some subset of nodes that are connected in the true (unknown) underlying graph structure G_T. For each example D_j, we are given the observed count x_i and expected count μ_i for each graph node v_i, $i = 1 \ldots N$. We assume that each training example D_j has an unobserved set of affected nodes S_j^T that is a connected subgraph of G_T. Unaffected nodes $v_i \notin S_j^T$ are assumed to have counts x_i drawn from some distribution with mean μ_i, whereas affected nodes $v_i \in S_j^T$ are assumed to have higher counts. Given these training examples, we have three main goals:

 Published by the IEEE Computer Society 2469-7087/18/$33.00 © 2018 IEEE

- Accurately estimate the true underlying graph structure G_T.
- Given a separate set of test examples $\{D_1, \ldots, D_J\}$ drawn from $\boldsymbol{D}$, identify the affected subgraphs S_j^T. Accuracy of detection is measured by the average overlap coefficient between the true and identified subgraphs.
- Distinguish test examples drawn from $\boldsymbol{D}$ from examples with no affected subgraph ($S_j^T = \phi$). Detection power is measured by the true positive rate for a fixed false-positive rate.

A key insight of our graph learning framework is to evaluate the quality of each graph structure G_m, with m edges, by comparing the most anomalous subsets detected with and without the graph constraints. For a given training example D_j, we can use the fast subset scan[8] to identify the highest-scoring unconstrained subset $S_j^* = \arg\max_{S \subseteq V} F(S)$, with score $F_j = F(S_j^*)$. This can be done efficiently (in linear rather than exponential time) because expectation-based scan statistics satisfy the linear-time subset scanning property.[8] We can use GraphScan to compute the highest-scoring connected subgraph $S_{mj}^* = \arg\max_{S \subseteq V: S \text{ connected in } G_m} F(S)$, with score $F_{mj} = F(S_{mj}^*)$. We then compute the mean normalized score $\overline{F}_{norm}(G_m) = (1/J) \Sigma_{j=1,\ldots,J(Fmj/Fj)}$ averaged over all J training examples as a measure of graph quality.

Intuitively, if a given graph G_m is similar to G_T, then the maximum connected subgraph score F_{mj} will be close to the maximum unconstrained subset score F_j for many training examples, and $\overline{F}_{norm}(G_m)$ will be close to 1. On the other hand, if graph G_m is missing essential connections, we expect the values of F_{mj} to be much lower than the corresponding F_j, and $\overline{F}_{norm}(G_m)$ will be much lower than 1. Additionally, we would expect a graph G_m with high scores F_{mj} on the training examples to have high power to detect future events drawn from the same underlying distribution. However, any graph with a large number of edges will also score close to the maximum unconstrained score. For example, if graph G_m is the complete graph on N nodes, then $\overline{F}_{norm}(G_m) = 1$. Such underconstrained graphs result in reduced detection power. Thus, we wish to optimize the tradeoff between a higher mean normalized score and a lower number of edges m. Our solution is to compare the mean normalized score of each graph structure G_m to the distribution of mean normalized scores for random graphs with the same number of edges m and choose the graph with the most significant score given this distribution.[9]

To avoid removing potentially important edges, we use correlation to break ties.

Learning Graph Structure Algorithm

Considering the mean normalized score $\overline{F}_{norm}(G_m) = (1/J) \Sigma_{j=1,\ldots,J} (F_{mj}/F_j)$ as a measure of graph quality, we can search for the graph G_m with the highest mean normalized score. However, it is computationally infeasible to search exhaustively over all $2^{|V|(|V|-1)/2}$ graphs. Even computing the mean normalized score of a single graph G_m could require a substantial amount of computation time, because it requires calling a graph-based event detection method such as GraphScan to find the highest-scoring connected subgraph for each training example D_j. We refer to this call as BestSubgraph(G_m, D_j) for a given graph structure G_m and training example D_j. Here, we instantiate BestSubgraph using the GraphScan algorithm[3] (for a comparison of other alternatives, see our previous work[9]).

Thus, we propose Learning Graph Structure (LGS), a greedy framework for efficiently learning graph structure. LGS starts with the complete graph on N nodes and sequentially removes edges until no edges remain (see Figure 1). For each graph G_m, we produce graph G_{m-1} by considering all m possible edge removals and choosing the one that maximizes the mean normalized score, which we refer to as BestEdge(G_m, D). Once we have obtained the sequence of graphs $G_0, \ldots, G_M$, we can then use randomization testing to choose the most significant graph G_m, as described earlier. The idea is to remove unnecessary edges while preserving essential connections that keep the maximum connected subgraph score close to the maximum unconstrained subset score for many training examples.

However, a naive implementation of greedy search would require $O(N^4)$ calls to BestSubgraph, because $O(N^2)$ graph structures G_{m-1} would be evaluated for each graph G_m to choose the next edge for removal. Even a sequence of random edge removals would require $O(N^2)$ calls to BestSubgraph to evaluate each graph $G_0, \ldots, G_M$. As described in our previous work,[9] our efficient graph learning framework improves on both of these bounds, performing exact or approximate greedy search with $O(N^3)$ or $O(N \log N)$ calls to BestSubgraph, respectively. The key insight is that, for a given graph G_m and example D_j, only $O(N)$ of the $O(N^2)$ candidate edge removals disconnects the highest-scoring subgraph S_j^*. For the remaining edges, we know $S_{m-1,j}^* = S_{mj}^*$ and do not need to call BestSubgraph.

To implement BestEdge(G_m, D), we note that greedily choosing the edge that

1. Compute correlation ρ_{ik} between each pair of nodes v_i and v_k, $i \neq k$, to be used in step 5.

2. Compute the highest-scoring unconstrained subset S_j^* and its score F_j for each example D_j using the fast subset scan.[8]

3. For $m = \frac{N(N-1)}{2}$, let G_m be the complete graph on N nodes. Set $S_{mj}^* = S_j^*$ and $F_{mj} = F_j$ for all training examples D_j, and set $\overline{F}_{norm}(G_m) = 1$.

4. **while** number of remaining edges $m > 0$ **do**

5. Choose edge e_{ik} = BestEdge (G_m, D), and set $G_{m-1} = G_m$ with e_{ik} removed.

6. **for** each training example D_j **do**

7. **If** removing edge e_{ik} disconnects subgraph S_{mj}^*, then set $S_{m-1,j}^* = \text{BestSubgraph}(G_{m-1}, D_j)$ and $F_{m-1,j} = F(S_{m-1,j}^*)$. Otherwise, set $S_{m-1,j}^* = S_{mj}^*$ and $F_{m-1,j} = F_{mj}$.

8. **end for**

9. Compute $\overline{F}_{norm}(G_m) = \frac{1}{J}\sum_{j=1,\ldots,J} \frac{F_{m-1,j}}{F_j}$.

10. **end while**

11. Repeat steps 3 through 10 for R randomly generated sequence of edge removals to find the most significant graph G_m.

Figure 1. The Learning Graph Structure (LGS) framework.

maximizes the mean normalized score for each graph G_m could still be prohibitively expensive. Thus, we consider a faster (but approximate) "pseudo-greedy" approach that uses the fact that $F_{m-1,j} = F_{mj}$ if removing edge e_{ik} does not disconnect subgraph S_{mj}^*, and $F_{m-1,j} < F_{mj}$ otherwise. Thus, we count the number of subgraphs S_{mj}^*, for $j = 1, \ldots, J$, which would be disconnected by removing each possible edge e_{ik} from graph G_m, and we choose the e_{ik} that disconnects the fewest subgraphs. The resulting graph G_{m-1} is expected to have a mean normalized score $\overline{F}_{norm}(G_{m-1})$, which is close to $\overline{F}_{norm}(G_m)$, since $F_{m-1,j} = F_{mj}$ for many subgraphs, but this approach does not guarantee that the graph G_{m-1} with highest mean normalized score will be found. However, because we choose the edge e_{ik} for which the fewest subgraphs S_{mj}^* are disconnected, and only need to call BestSubgraph for those examples D_j where removing e_{ik} disconnects S_{mj}^*, we are choosing the edge e_{ik} that requires the fewest calls to BestSubgraph for each graph G_m. This results in only $O(N \log N)$ calls to BestSubgraph instead of $O(N^3)$ for the exact greedy method.[9] To avoid removing potentially important edges, we use correlation to break ties: if two edge removals e_{ik} disconnect the same number of subgraphs, the edge with the lower correlation is removed. The intuition is that if two nodes are connected by an edge in the latent graph, then we expect both nodes to be simultaneously affected or unaffected by an event.

Experimental Setup

Our experiments focus on detection of simulated disease outbreaks injected into real-world Emergency Department (ED) data from 10 hospitals in Allegheny County, Pennsylvania.[9] The dataset consists of the number of ED admissions with respiratory symptoms for each of the $N = 97$ ZIP codes for each day from 1 January 2004 to 31 December 2005. Our simulations assume that the disease outbreak starts at a center location (chosen uniformly at random) and spreads over some underlying graph structure, increasing in size and severity over time. Outbreaks were assumed to be 14 days in length, and we assume that an affected node remains affected through the outbreak duration. Our previous work provides a detailed description of the outbreak simulation.[9]

We considered simulated outbreaks that spread from a given ZIP code to spatially adjacent ZIP codes, as is commonly assumed in the literature. Thus, we formed the adjacency graph for the 97 Allegheny County ZIP codes, in which two nodes are connected by an edge if the corresponding ZIP codes share a boundary. We performed two sets of experiments: for the first set, we generated simulated injects using the adjacency graph, whereas for the second set, we added additional edges between randomly chosen nodes to simulate travel patterns. As noted earlier, a contagious disease outbreak might be likely to propagate from one location to another that is not spatially adjacent, based on individuals' daily travel. We hypothesize that inferring these additional edges will lead to improved detection performance. For each set of experiments, we produced $J = 200$ training injects and an additional 200 test injects drawn from the same distribution.

We compared the performance of our learned graphs with that of the learned graphs from the MultiTree algorithm,[7] which was shown to outperform previously proposed graph structure learning algorithms such as NetInf[5] and ConNIe.[6] We used the publicly available implementation of the algorithm, and we assumed that MultiTree is given

the true labels of the affected subset of nodes for each training example. For each competing method, once a graph structure was learned, we used the GraphScan algorithm (assuming the given graph structure) to identify the highest-scoring connected subgraph S and its likelihood ratio score $F(S)$ for each day of each simulated inject, and for each day of the original ED data with no cases injected.

We evaluated detection performance using two metrics: average time to detection (assuming a false-positive rate of 1 per month, typically considered acceptable by public health), and spatial accuracy (overlap between true and detected clusters). To compute detection time, we first compute the score threshold F_{thresh} for detection at 1 false positive per month. This corresponds to the 96.7th percentile of the daily scores from the original ED data. Then, for each simulated inject, we compute the first outbreak day d with $F(S) > F_{\text{thresh}}$ and average the time to detection over all 200 test injects. To evaluate spatial accuracy, we compute the average overlap coefficient between the detected subset of nodes S^* and the true affected subset S^T at the midpoint (day 7) of the outbreak, where overlap is defined as $\left(|S^* \cap S^T| / |S^* \cup S^T|\right)$.

Detection performance is often improved by including a proximity constraint,[3] in which we perform separate searches over the local neighborhood of each of the N graph nodes, which comprises that node and its $k - 1$ nearest neighbors, and report the highest-scoring connected subgraph over all neighborhoods. We evaluate how performance varies as a function of neighborhood size, considering all $k = 5, 10, \ldots, 45$.

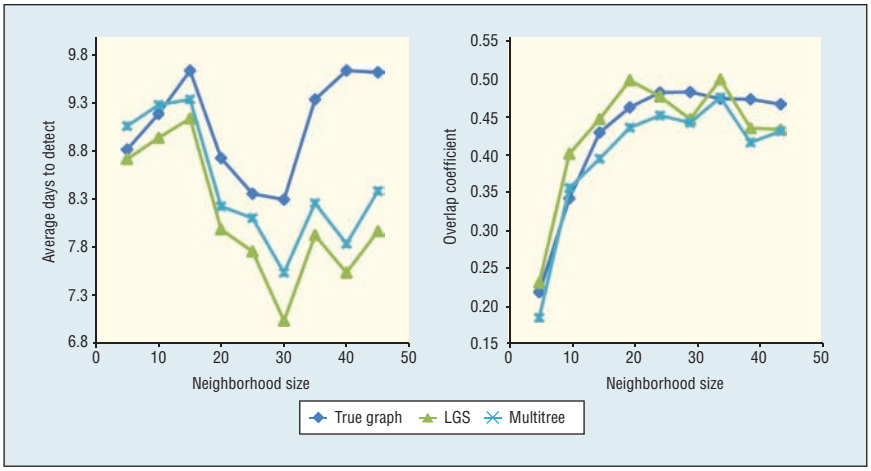


Figure 2. Comparison of detection performance of the true and learned graphs for injects based on ZIP code adjacency. The graphs learned by LGS achieved more timely detection than the true graph while maintaining a comparable spatial overlap coefficient.

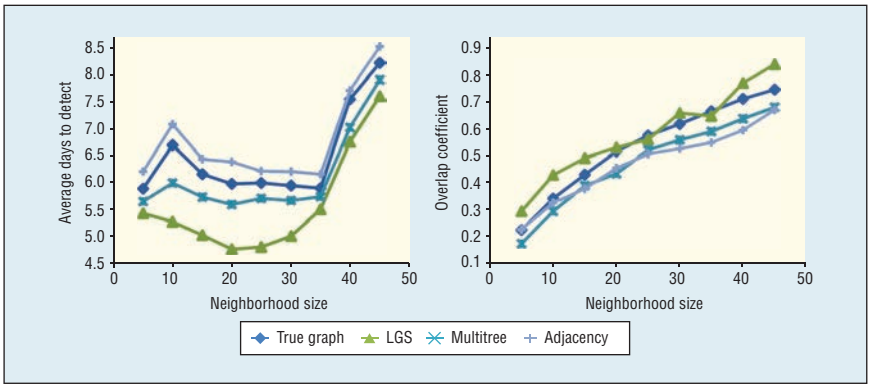


Figure 3. Comparison of detection performance of the true, learned, and adjacency graphs for injects based on adjacency with simulated travel patterns. The graphs learned by LGS achieved more timely detection than the true graph or assuming an incorrect graph (the adjacency graph in this case) while maintaining a comparable spatial overlap coefficient.

Experimental Results

We first evaluated the detection time and spatial accuracy of GraphScan, using the graphs learned by LGS and MultiTree, for simulated injects that spread based on the adjacency graph, as shown in Figure 2. The figure also shows GraphScan's performance given the true ZIP code adjacency graph. The graphs learned by LGS had a better spatial overlap coefficient and more timely detection as compared to graphs learned by MultiTree. Surprisingly, all of the learned graphs achieved more timely detection than the true graph: for the optimal neighborhood size of k = 30, LGS detected an average of 1.4 days faster than the true graph. This could be because the learned graphs, in addition to recovering most of the edges of the adjacency graph, also included additional edges to nearby but not spatially adjacent nodes (for example, neighbors of neighbors). These extra edges provided added flexibility and improved detection time when some nodes were more strongly affected than others, enabling the strongly affected nodes to be detected earlier

in the outbreak, before the entire affected subgraph was identified.

Next, we compared detection time and spatial accuracy using the graphs learned by LGS and MultiTree for simulated injects that spread based on the ZIP code adjacency graph, with additional random edges added to simulate travel patterns (see Figure 3). This figure also shows the detection performance given the true (adjacency plus travel) graph and the adjacency graph without travel patterns. Again, LGS has a better spatial overlap coefficient as compared to the original adjacency graph and MultiTree. Our learned graphs can detect outbreaks 0.8, 1.2, and 1.7 days earlier than MultiTree, the true graph, and the adjacency graph without travel patterns, respectively. This demonstrates that our methods can successfully learn the additional edges due to travel patterns, substantially improving detection performance.

As our associated technical report shows,[9] our results demonstrate that the graph structures learned by LGS are similar to the true underlying graph structure, capturing nearly all of the true edges but also adding some additional edges. The resulting graph achieves a similar spatial overlap coefficient between true and detected clusters. Interestingly, the learned graph often has better detection power than the true underlying graph, enabling more timely detection of outbreaks. We believe this is because the learning procedure is designed to capture not only the underlying graph structure, but the characteristics of the events that spread over that graph.

Our ongoing work focuses on extending the graph structure learning framework in several directions, including learning graph structures with directed rather than undirected edges, learning graphs with weighted edges, and learning dynamic graphs where the edge structure can change over time. Our current approach does not rely on temporal information, using only the observed and expected counts at each node to compute correlations and to identify the highest scoring connected subgraph for each combination of graph structure and training example. To learn directed edges within our general structure learning framework, we plan to incorporate this temporal information by considering cross-correlations between each pair of nodes and by incorporating a new variant of GraphScan[10] that can detect dynamic patterns on graphs while enforcing constraints on temporal consistency. ■

Acknowledgments

This work was partially supported by NSF grant IIS-0953330.

References

1. G.P. Patil and C. Taillie, "Upper Level Set Scan Statistic for Detecting Arbitrarily Shaped Hotspots," *Environmental and Ecological Statistics*, vol. 11, no. 2, 2004, pp. 183–197.
2. T. Tango and K. Takahashi, "A Flexibly Shaped Spatial Scan Statistic for Detecting Clusters," *Int'l J. Health Geographics*, vol. 4, no. 11, 2005; doi:10.1186/1476-072X-4-11.
3. S. Speakman, E. McFowland III, and D.B. Neill, "Scalable Detection of Anomalous Patterns with Connectivity Constraints," *J. Computational and Graphical Statistics*, vol. 24, no. 4, 2015, pp. 1014–1033.
4. D.B. Neill et al., "Detection of Emerging Space-Time Clusters," *Proc. 11th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining*, 2005, pp. 218–227.
5. M. Gomez-Rodriguez, J. Leskovec, and A. Krause, "Inferring Networks of Diffusion and Influence," *Proc. 16th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining*, 2010, pp. 1019–1028.
6. S. Myers and J. Leskovec, "On the Convexity of Latent Social Network Inference," *Advances in Neural Information Processing Systems*, vol. 23, 2010, pp. 1741–1749.
7. M. Gomez-Rodriguez and B. Schölkopf, "Submodular Inference of Diffusion Networks from Multiple Trees," *Proc. 29th Int'l Conf. Machine Learning*, 2012, pp. 489–496.
8. D.B. Neill, "Fast Subset Scan for Spatial Pattern Detection," *J. Royal Statistical Soc. Series B: Statistical Methodology*, vol. 74, no. 2, 2012, pp. 337–360.
9. S. Somanchi and D.B. Neill, "Graph Structure Learning from Unlabeled Data for Event Detection," tech. report, Carnegie Mellon Univ., 2017; http://arxiv.org/abs/1701.01470.
10. S. Speakman, Y. Zhang, and D.B. Neill, "Dynamic Pattern Detection with Temporal Consistency and Connectivity Constraints," *Proc. 13th IEEE Int'l Conf. Data Mining*, 2013, pp. 697–706.

Sriram Somanchi is an assistant professor of business analytics in the Mendoza College of Business at the University of Notre Dame. Contact him at somanchi.1@nd.edu.

Daniel B. Neill is an associate professor of information systems and the director of the Event and Pattern Detection Laboratory at Carnegie Mellon University's Heinz College. Contact him at neill@cs.cmu.edu.

This article originally appeared in IEEE Intelligent Systems, *vol. 32, no. 2, 2017.*

IEEE computer society

SHARE THE GIFT OF KNOWLEDGE:

Give Your Favorite Student a Membership to the IEEE Computer Society!

With an **IEEE Computer Society Membership**, your student will be able to build their network, learn new skills, and access the best minds in computer science before they're even out of school. Your gift includes thousands of key resources that will quickly transition them from classroom to conference room, such as:

- **A subscription to *Computer* magazine** (12 issues per year)
- **A subscription to *ComputingEdge*** (12 issues per year)
- **Local chapter membership**
- **Full access to the Computer Society Digital Library**
- **Eligible for 3 student scholarships where we give away US$40,000 yearly**
- **Skillsoft:** Learn new skills anytime with access to 3,000 online courses, 11,000 training videos, and 6,500 technical books.
- **Books24x7:** On-demand access to 15,000 technical and business resources.
- **Unlimited access to computer.org and myCS**
- **Conference discounts**
- **Members-only webinars**
- **Deep member discounts** on programs, products, and services

Give Your Gift at: www.computer.org/2018gift

Editor: Charles Petrie • cjpetriejr@gmail.com

Social Media Won't Free Us

Daniel Gayo-Avello • *University of Oviedo, Spain*

After losing the presidential elections in Iran in 2009, candidate Mir-Hossein Mousavi and his supporters claimed electoral fraud and confronted the regime forces in bloody clashes. In January 2011, president Zine El Abidine Ben Ali (in office since 1987) fled Tunisia after massive demonstrations spread through the country. After that, a number of Arab countries experienced their own demonstrations, seeking political change. The Arab Spring was born.

All of these events captured the attention and imagination of Westerners, not because of the prospect of democracy arising in those countries, but because of the presumed role played by Western-made technologies — namely, social media platforms.

According to journalists, Iran experienced a "Twitter revolution," while Egypt's was a "Facebook revolution." We were told that social media was crucial for dissenters to organize themselves, plan their actions, and publicize their agendas both in their own countries and abroad. The hype reached such a point that Twitter was asked to delay a scheduled outage to allow the protest in Iran go on undisturbed, and some officials even suggested that its founders deserved the Nobel Peace Prize.

Those glossy portrayals unfortunately were wrong: the Arab Spring was as much a social media revolution as the Mexican Revolution was a Leica (a brand of camera) revolution.[1] Certainly, social media played a role but only for a minority of protesters and, to be honest, it turned out quite badly for some of them — at least in Iran.

Power and Paranoia

The truth is that during the protests in Iran, only about 20,000 people (from a population of 70 million) were using Twitter, and they were far from being representative of Iranians as a whole, nor even of the opposition forces. Similar arguments can be applied to Egypt, where social media activists that Western media labeled *spokespersons* have been accused of not only being detached from most protesters but even from the uprising's real action and violence.[2]

On top of that, for an authoritarian regime being confronted by some of its people, it doesn't matter whether social media plays a role. It's enough for relevant state actors to claim or assert that use of such technology leads to security risks, so that any user might be viewed as a potential dissenter — or even a spy.

Indeed, the viewpoint from Western media and governments about Twitter usage in Iran backfired on Iranian Twitter users. To start with, access to the service was blocked by local ISPs; moreover, all of the information about those dissenters was readily available for authorities to collect — pictures, videos, texts, and connections to other activists on the same social networking site. What's more, regime sympathizers were not only using social media to spread proregime information to demobilize dissenters, but also crowdsourcing the identification of protesters appearing in the collected pictures.

Actually, claiming that social media is able to overthrow authoritarian regimes only makes those regimes more authoritarian and paranoid. For instance, in Egypt, which experienced the uprisings a year and a half after the Iranian Green Movement and shortly after the Tunisian revolution, Twitter and Facebook were blocked almost immediately after the first demonstrations, and the whole Internet was eventually shutdown for almost a week.

 Published by the IEEE Computer Society 2469-7087/18/$33.00 © 2018 IEEE

Still, the Egyptian shutdown wasn't the only one nor the longest; before that there was another case in China. After ethnic riots in July 2009 in the Xinjiang Uyghur Autonomous Region, the Internet was shut down for 10 months, and, needless to say, webmasters and bloggers were imprisoned.

However, shutting down all Internet access has negative outcomes for the regime performing it and for the country's economy. That's why such an approach is seldom used. It's much more common to degrade the service — for instance, limiting uploading videos or streaming platforms for broadcasting — or to block particular platforms such as social networking sites.

Under Pressure

At this point, it must be noted that all of these authoritarian behaviors toward the Internet and social media require the collaboration of private companies, many of them headquartered in democratic countries. This collaboration always takes place on the grounds of legality — that is, the laws from the authoritarian regime — and, hence, companies claim they're forced to comply. That, however, is false: foreign companies could choose not to enter a given market and those who do it are choosing profit over principles.

To their credit, the most popular social media platforms such as Facebook or Twitter are blocked in China and Iran, so we can assume they haven't yielded to those governments' demands. Still, they aren't free of governmental pressures and blocks, even in nonauthoritarian countries.

Indeed, the most worrisome aspect of social media platforms isn't the behavior of their owners when confronted with authoritarian governments, but when facing demands from presumed democratic governments.

This is crucial, because whether we like it or not, most liberal democracies are slowly drifting to a flawed status in a post-democracy scenario.[3] Under such a scenario, social media in particular and virtual realms in general will be much more strongly controlled — mainly based on the claim of fighting terrorism. At the same time, these platforms will play an increasingly important role during election campaigns.

Flawed and Post-Democratic Scenarios

Hence, before proceeding any further I must briefly discuss the features of flawed and post-democratic scenarios, even at the risk of simplifying them a bit. In both cases, elections are free and fair and thus governments are changed by the people; still, political culture and participation are poor. According to the post-democracy theory, such a situation exists mainly because people have little voice in actual politics, which are slanted to the advantage of elites and corporations. Most participation focuses on elections, which have become a controlled spectacle aimed at persuading people to vote for one or another candidate on the basis of issues that lobbyists or special interest groups (most of them powered by corporations) have carefully selected. Another feature of this scenario is the rise of nonmainstream parties and movements, which at some points use unconventional, even contentious, approaches to political participation and rely heavily on social media.

Under such circumstances, social media owners are confronted with two main dilemmas: First, most of the time their platforms provide a realm for political participation, going from the conventional to the extreme, and are thus used to spread political material that sometimes will be contentious — including some that could be considered extremist or even terrorist. Second, during electoral campaigns, social media is used as a "weapon of mass persuasion"[4] where politicians attempt to "seem authentic" but can become victims of as well as instigate smear campaigns.

In both cases social media, and thus its owners, will be deemed responsible for allowing the spread of extremist or terrorist material; spreading biased views, even fake news; polarizing people and isolating them in echo chambers; and powering social bots that unfairly tilt electoral outcomes. Because these platform owners are deemed responsible, they will be pressured to take some kind of action by both the population and the authorities.

When faced with the spread of extremist or terrorist material, social media companies might be forced to identify some users, filter content, or block accounts. However, depending on the definition of terrorism, such measures could be considered as attacks on free speech but implemented under the guise of national security.[5,6] If you think that such menaces just occur in Russia or Turkey — each of them a flawed democracy on the brink of authoritarianism — you should think twice, because they're being implemented or are under consideration in presumed liberal countries.

For instance, South Korea passed a real name verification law in 2007, which was enforced for five years before being declared unconstitutional; such a law was actually encouraging self-censorship. The UK has proposed banning individuals from broadcasting content, including social media, on the basis of the so-called Extremist Disruption Orders. In France, using social media is an aggravating circumstance when facing charges of terrorism, and the mere action of consulting online information labeled as terrorist is a terrorist act; moreover, authorities

might ask providers to block access to sites hosting such content.

In my own country — Spain — the criminal code was modified in such a way that any pressure on public authorities, including through social media, can be considered a form of terrorism. Indeed, the simple action of "making a statement on social media that could be 'perceived' as inciting others to commit violent attacks will be outlawed, even if the statement cannot be directly linked to an act of violence."[7] Actually, at the moment of this writing a 21-year-old Spanish woman has received a one-year suspended jail sentence for being accused of glorifying terrorism due to two jokes made on Twitter.[8]

Thus, we find in presumed democratic countries laws and measures that discourage users from expressing unconventional and contentious political ideas for fear of being accused of extremism or terrorism, but allow authorities to block sites or content. At the same time that contentious ideas are increasingly risky for users to post on social media, the role that social media plays during electoral campaigns is becoming more prominent.

Purportedly, social media can help candidates spread their message, organize their campaign, raise funds, boost grassroots support, persuade undecided voters, get feedback from the electorate, engage with citizens in fruitful discussion, and get a minor but still valuable boost in votes.

The truth is that social media is used by candidates mainly as a broadcasting platform when spreading their message, often with the goal of setting the mass media agenda; feedback is seldom if ever incorporated into their manifests; and discussions with common citizens are rare and carefully orchestrated. Still, this isn't a problem per se but just a symptom of post-democracy, where electoral campaigns are a spectacle that most citizens watch in a passive attitude.

However, not every citizen is passive when facing electoral campaigns, and some of these people use social media as a megaphone to vocalize their positions. The problem is that those who are more vocal and active tend to represent the most extreme within each party, and that's a big problem. It's problematic because moderate voices prefer to remain silent or are silenced by those dominating the conversation. That, in turn, means that those solely following the posts but not posting themselves tend to believe that the majority position is the most extreme and vocal.

Because of this, we have a highly polarized population that's ripe for misinformation, disinformation, and propaganda. During the past five years many researchers have raised their voices to warn us about this worrisome situation,[9,10] but social media owners have done little. Research about the spreading of disinformation was targeted as a smear campaign itself.[11] However, after the 2016 presidential election in the US and the purported role played by the spreading of so-called fake news, this research is grabbing much more attention.

Social media companies claim that they're going to fight against fake news. However, there's a problem with this: while fact-checking a piece of fake news is possible, albeit not simple, for humans, it's a daunting task for machines. Certainly a machine-learning approach is feasible, but then the eventual system would not be fact-checking news but simply exercising the opinion bias of those who trained the system. In that case, the question is who's going to be the judge about what's true and what's false? The private owners of social media? If this is the case, what's the difference between censorship outsourced in China to service providers and liberal democracies asking for fake news filtering?

Indeed, the crux of the matter is that any technology deployed to filter and remove content can, and likely would, be applied to content different from that originally intended. Therefore, the requests to filter hate speech, extremism, terrorism, and fake news arising in liberal democracies might very well be used against legitimate users and organizations expressing views differing from those of the supporters of the status quo.

Social media might be a powerful tool, but it can be used in many adversarial ways: chasing dissenters, spreading disinformation, and eventually silencing those with a point of view different from the dominating (not majority) opinion. Moreover, asking private owners of social media to detect and filter problematic content is not only difficult but might very well backfire against free speech. On top of that, the current climate in liberal democracies isn't the best for implementing such technologies, given both the passivity of most citizens and the eagerness of authorities to fight extremist positions. I can't offer a solution, but what I'm sure about is that social media won't free us.

References

1. U. Mejias, "The Twitter Revolution Must Die," *Int'l J. Learning and Media*, vol. 2, no. 4, 2010, pp. 3–5.
2. Z. Abul-Magd, "Occupying Tahrir Square: The Myths and the Realities of the Egyptian Revolution," *South Atlantic Q.*, vol. 111, no 3, 2012, pp. 565–572.
3. C. Crouch, *Coping with Post-Democracy*, Fabian Society, 2000.
4. T. Hwang, "Weapons of Mass Persuasion," *Motherboard*, 18 Sept. 2015; https://motherboard.vice.com/en_us/article/weapons-of-mass-persuasion.

5. P. Wintour, "The UN Accuses Saudi Arabia of Using Terror Laws to Suppress Free Speech," *The Guardian*, 4 May 2017; www.theguardian.com/world/2017/may/04/un-accuses-saudi-arabia-of-using-terror-laws-to-suppress-free-speech.
6. J. Beasley-Murray et al., "'Anti-Terror' Laws Already Eroding Free Speech, Debate," *The Tyee*, 5 Apr. 2017; https://thetyee.ca/Opinion/2017/04/05/Anti-Terror-Laws-Free-Speech.
7. Amnesty Int'l, "Spain: Two-Pronged Assault Targets Rights and Freedoms of Spanish Citizens, Migrants, and Refugees," 26 Mar. 2015; www.amnesty.org/en/latest/news/2015/03/spain-two-pronged-assault-targets-rights-and-freedoms-of-spanish-citizens-migrants-and-refugees.
8. S. Jones, "Jail for a Joke: Student's Case Puts Free Speech under Spotlight in Spain," *The Guardian*, 18 Apr. 2017; www.theguardian.com/world/2017/apr/18/student-cassandra-vera-tweet-case-puts-free-speech-under-spotlight-in-spain.
9. P.T. Metaxas and E. Mustafaraj, "Social Media and the Elections," *Science*, vol. 338, no. 6106, 2012, pp. 472–473.
10. J. Ratkiewicz et al., "Truthy: Mapping the Spread of Astroturf in Microblog Streams," *Proc. 20th Int'l Conf. Companion on World Wide Web*, 2011, pp. 249–252.
11. J. Mervis, "An Internet Research Project Draws Conservative Ire," *Science*, vol. 346, no. 6210, 2014, pp. 686–687.

Daniel Gayo-Avello is an associate professor in the Department of Computer Science at the University of Oviedo, Spain. His research interests include information retrieval, Web mining, in particular query log mining, and online social network analysis. Gayo-Avello has a PhD in computer science from the University of Oviedo. Contact him at dani@uniovi.es or via Twitter at @PFCdgayo.

This article originally appeared in IEEE Internet Computing, *vol. 21, no. 4, 2017.*

ADVERTISER INFORMATION

Advertising Personnel

Debbie Sims: Advertising Coordinator
Email: dsims@computer.org
Phone: +1 714 816 2138 | Fax: +1 714 821 4010

Advertising Sales Representatives (display)

Central, Northwest, Southeast, Far East:
Eric Kincaid
Email: e.kincaid@computer.org
Phone: +1 214 673 3742
Fax: +1 888 886 8599

Northeast, Midwest, Europe, Middle East:
David Schissler
Email: d.schissler@computer.org
Phone: +1 508 394 4026
Fax: +1 508 394 1707

Southwest, California:
Mike Hughes
Email: mikehughes@computer.org
Phone: +1 805 529 6790

Advertising Sales Representative (Classifieds & Jobs Board)

Heather Buonadies
Email: h.buonadies@computer.org
Phone: +1 201 887 1703

Advertising Sales Representative (Jobs Board)

Marie Thompson
Email: marie@4caradio.org
Phone: 714-813-5094

EDITORS: Rick Kuhn, NIST, kuhn@nist.gov
Tim Weil, Scram Systems, tweil.ieee@gmail.com

It Doesn't Have to Be Like This

Cybersecurity Vulnerability Trends

Rick Kuhn, *NIST*
Mohammad Raunak, *Loyola University Maryland*
Raghu Kacker, *NIST*

It often seems that every newly announced major data breach sets a record for depth and size of impact. Internet users—nearly everyone these days—naturally wonder: Why is this happening, and how much worse can it get? In the inaugural article for this column, published in January 2009, we reviewed trends in vulnerabilities for the previous eight years.[1] Our goal, then as well as now, is to improve the understanding of cybersecurity vulnerabilities so that we can prevent them. One Moore's law generation later, we followed that article with another review of trends, finding some encouraging results.[2] Here, we review some of those earlier findings, discuss what has happened since then, and highlight prospects for the near future.

Early Analyses

Our data source is the US National Vulnerability Database (NVD; nvd.nist.gov), which has collected nearly all publicly reported vulnerabilities since 1997 using the Common Vulnerabilities and Exposures (CVE) dictionary. The NVD was developed and is run by NIST, with support from the US Department of Homeland Security's National Cyber Security Division. As of 2017, the NVD included more than 85,000 vulnerabilities, and the collection is expanded daily. With two decades of data, the NVD is an invaluable resource for security analysts.

One of the primary observations from the January 2009 analysis was that the total number of vulnerabilities per year had begun to decline, from a peak of nearly 7,000 in 2006 to about 5,500 in 2008. It appeared that developers and security administrators had begun taking security seriously, including it as a key component in development, and staying up-to-date on mitigation techniques. Code flaws that were widely used in system exploits in the 1980s and 1990s, such as format string vulnerabilities and race conditions, were appearing in only a dozen or two cases each year, accounting for less than 1 percent each of the vulnerabilities in thousands of applications. Better development methods and tools had begun to make a difference.

But the 2009 analysis also revealed a trend that we see repeatedly in all aspects of security—new IT produces new challenges to secure it. During the previous decade, e-commerce and other web-based services proliferated, producing new challenges for protection and new opportunities for attackers. While buffer overflows and misconfigurations had long been the main sources of weakness in systems defenses, SQL injections and cross-site scripting were respectively the number one and number two vulnerability types in 2008 (Figure 1). (Note that the analysis is limited to the distribution of primary vulnerability categories; another 10 to 15 percent each year are classed as either "other" or "insufficient information.") As we will see later in this article, the trends

 Published by the IEEE Computer Society 2469-7087/18/$33.00 © 2018 IEEE

for these two vulnerability types illustrate an important lesson for managing cybersecurity.

A follow-on review added data from 2009 to 2010,[2] providing more in-depth analysis and showing that vulnerabilities continued to decline as they had since 2006. Among the interesting findings from this analysis was that the average difficulty of exploitation began to change in 2006. Prior to this time, nearly all vulnerabilities had been easy to exploit, but after this time, the access complexity of about half of the vulnerabilities was either medium or high. This finding suggests that defensive measures in code and system administration were being successfully employed.

Among the negative findings in that study, it was found that the proportion of high-, medium-, and low-severity vulnerabilities had changed little over the period from 2001 to 2010. That is, serious errors were roughly as common in 2010 as they had been a decade earlier (Figure 2). Additionally, buffer errors were still one of the major sources of system vulnerabilities, and we reported on a separate analysis that found that roughly 93 percent of these involved only a single condition (typically, failure to check array bounds; a few buffer errors required that two or more conditions be true to exploit). We pointed out that even the most basic of secure programming practices, such as ensuring checks of all input string length, could eliminate a large proportion of these problems.

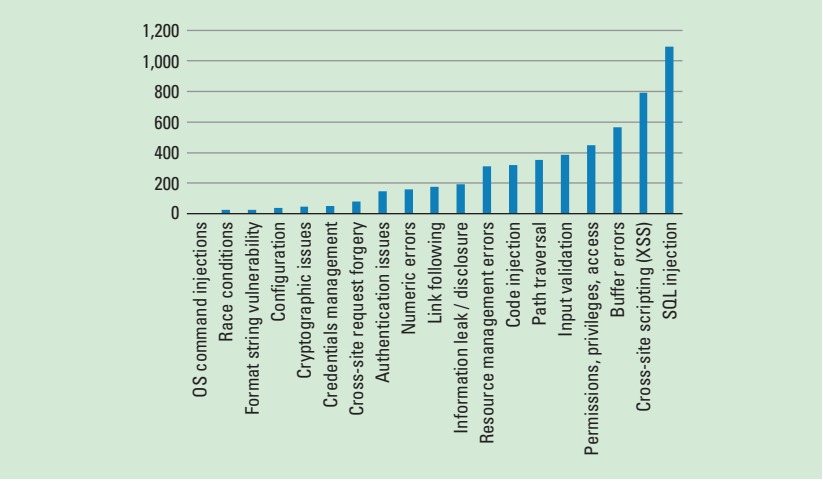


Figure 1. Major National Vulnerability Database classes from 2008. Web-related vulnerabilities were common during this time.

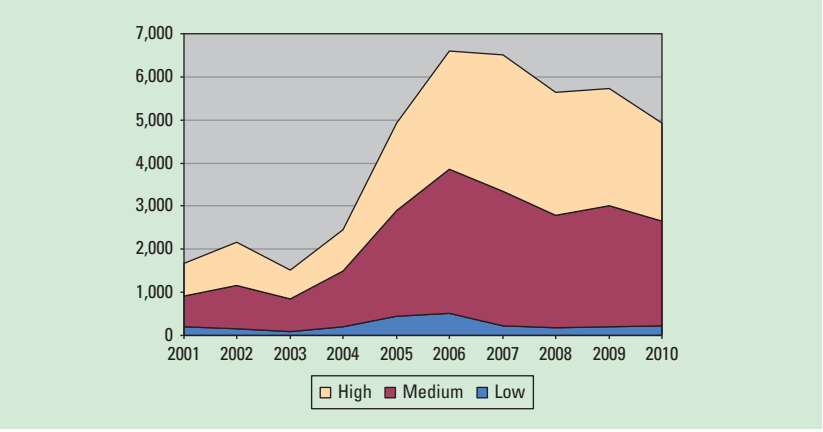


Figure 2. Vulnerabilities by severity. Vulnerabilities declined from 2006 to 2010, but about 96 percent were of medium to high severity.

Recent Trends

More recently, we revisited the review of NVD data through 2016[3] and found that medium- to high-severity vulnerabilities had declined slightly, from 96 percent in 2008 to about 90 percent for 2016 (Figure 3).

This review also included an additional type of analysis. Not all security-critical errors in software are specifically related to security. For example, buffer overflow errors usually result from failing to check that input is the appropriate size for internal storage—a check that should always be done—and could result in ordinary failures that are not necessarily security-relevant. How prevalent are ordinary coding errors like these among the vulnerabilities cataloged in the NVD?

To address this question, we can distinguish at least three types of errors: administrative and configuration errors, fundamental design-related errors, and ordinary coding or implementation-related errors:

- Configuration vulnerabilities result from bad configuration files or other administrative errors. One example is missing password checks. Information

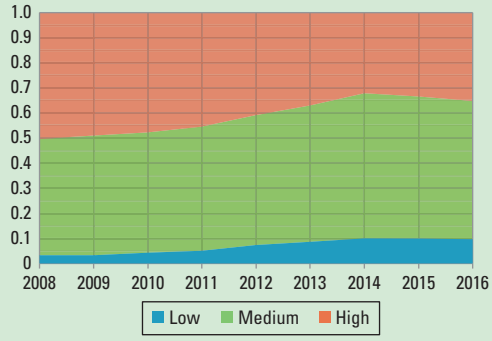


Figure 3. Vulnerability severity trends from 2008 to 2016. Medium to high-severity vulnerabilities declined slightly in this time span.

*Table 1. National Vulnerability Database categories.**

CWE-ID	Description	Type	Trend
CWE-16	Configuration	C	↓
CWE-20	Input validation	I	↑
CWE-22	Path traversal	I	↓
CWE-59	Link following	I	≈
CWE-78	OS command injections	I	↑
CWE-79	Cross-site scripting (XSS)	I	≈
CWE-89	SQL injection	I	↓
CWE-94	Code injection	I	↓
CWE-119	Buffer errors	I	↑
CWE-134	Format string vulnerability	I	≈
CWE-189	Numeric errors	I	↓
CWE-200	Information leak/disclosure	C	↑
CWE-255	Credentials management	D	↑
CWE-264	Permissions, privileges, access	D	↑
CWE-287	Authentication issues	D	≈
CWE-310	Cryptographic issues	D	↑
CWE-352	Cross-site request forgery	I	≈
CWE-362	Race conditions	I	↑
CWE-399	Resource management errors	I	↓

C = configuration, D = design, I = implementation

leaks also frequently result from failing to set up controls or apply updates.

- Design-related vulnerabilities originate in the planning and design of the system, and include selecting an outdated or weak cryptographic algorithm.
- Implementation vulnerabilities are errors in code, such as the buffer overflow example mentioned previously. Cross-site scripting is less obvious, but generally results from missing or inadequate input validation, and other forms of input validation failures are common.

Table 1 designates configuration, design, and implementation errors as C, D, and I, respectively. Note that the table also indicates whether the different vulnerability types are increasing (↑), decreasing (↓), or approximately unchanged (≈). Some of the changes have been significant. For example, information leak/disclosure (CWE-200) moved from the 9th to the 3rd most common vulnerability. There are some conjectures we could make about the increase of CWE-200:

- recent applications are getting more and more complex, which creates larger attack areas for information leak/disclosure types (CWE-200) of vulnerabilities; or
- attackers have found more success looking for vulnerabilities related to information leaks and have been focusing more on exploiting them over recent years.

Additional investigation would be useful to discover the reasons for this increase.

Vulnerabilities Resulting from Implementation Errors

As Figure 4 shows, implementation errors are by far the major source of vulnerabilities, accounting for roughly two-thirds of the total. Note that the number of vulnerabilities is related to the number of applications released, and new applications are released constantly, so it is important to consider the proportion rather than the number of vulnerability types. However, note that the total number of reports for the CWE types in Table 1 (excluding "other" and "insufficient information") was similar in these years, declining about 10 percent from 5,196 in 2008 to 4,722 in 2015. Note that three of the top five vulnerabilities in the table are implementation-based: buffer

errors, cross-site scripting, and input validation.

Remarkably, the proportion of implementation vulnerabilities for 2008 to 2016 is close to the 64 percent reported for 1998 to 2003 in another analysis.[4] This is somewhat surprising and discouraging, given that these vulnerabilities usually result from simple mistakes that should be easy to prevent. However, this finding also suggests the potential for relatively low-cost improvements. Static analysis tools can detect about 20 percent of CVE-defined errors,[5] and formal code inspection has been demonstrated to be highly effective in error reduction.[6] The key point of this analysis is that a very large proportion of security vulnerabilities arise from basic coding errors, which can be prevented and detected with a comprehensive program of static analysis and dynamic test methods.

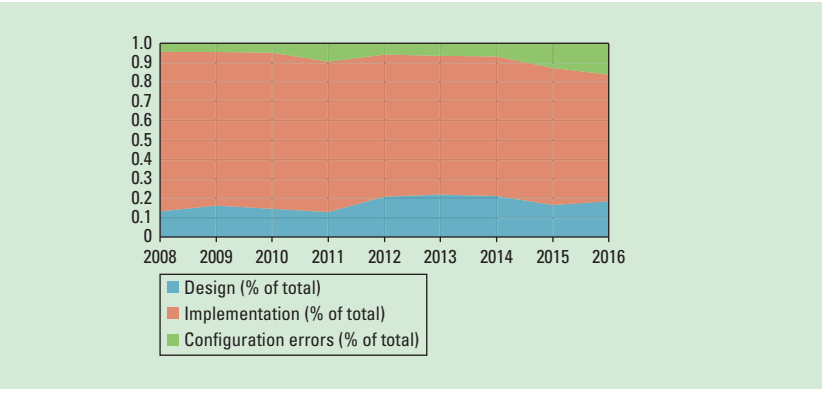


Figure 4. Vulnerability class trends from 2008 to 2016. Implementation errors are the major source of vulnerabilities, accounting for roughly two-thirds of the total.

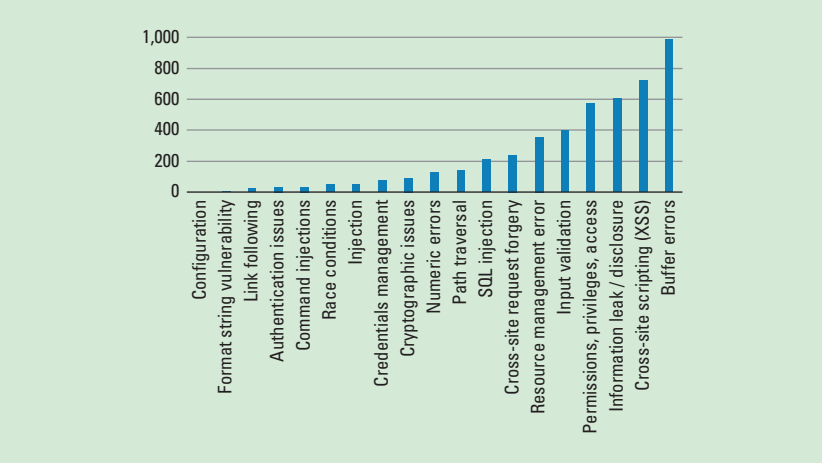


Figure 5. Distribution of vulnerabilities in 2015. This distribution changed significantly from 2008 to 2015.

As noted previously, SQL injection vulnerabilities were the number one most common type in 2008. By 2015, vulnerabilities of this type had been dramatically reduced (Figure 5). We believe better tools and improved development practices helped prevent this type of implementation error and can do so for the other types as well. For example, buffer overflow and similar buffer-related faults are now the largest category of vulnerabilities, despite the wide availability of tools to prevent them.[7] As suggested in the title of this article, we can reduce cybersecurity vulnerabilities using tools and methods that are readily available but must be applied.

Acknowledgments

Products may be identified in this document, but such identification does not imply recommendation by the US National Institute of Standards and Technology or the US government, nor that the products identified are necessarily the best available for the purpose.

References

1. R. Kuhn, H. Rossman, and S. Liu, "Introducing 'Insecure IT,'" *IT Professional*, vol. 11, no. 1, 2009, pp. 24–26.
2. R. Kuhn and C. Johnson, "Vulnerability Trends: Measuring Progress," *IT Professional*, vol. 12, no. 4, 2010, pp. 51–53.
3. D.R. Kuhn, M.S. Raunak, and R. Kacker, "An Analysis of Vulnerability Trends, 2008–2016," *Proc. IEEE Int'l Conf. Software Quality, Reliability, and Security Companion* (QRS-C), 2017, pp. 587–588.
4. J. Heffley and P. Meunier, "Can Source Code Auditing Software Identify Common Vulnerabilities and Be Used to Evaluate Software Security?" *Proc. 37th Ann. Hawaii Int'l Conf. System Sciences*, 2004; doi:10.1109/HICSS.2004.1265654.
5. I. Medeiros, N. Neves, and M. Correia, "Detecting and Removing Web Application Vulnerabilities with Static Analysis and Data Mining," *IEEE Trans. Reliability*, vol. 65, no. 1, 2016, pp. 54–69.
6. C. Jones, "Measuring Defect Potentials and Defect Removal Efficiency," *Crosstalk, J. Defense Software Eng.*, June 2008.

This article originally appeared in IT Professional, *vol. 19, no. 6, 2017.*

7. P.E. Black and I. Bojanova, "Defeating Buffer Overflow: A Trivial but Dangerous Bug," *IT Professional*, vol. 18, no. 6, 2016, pp. 58–61.

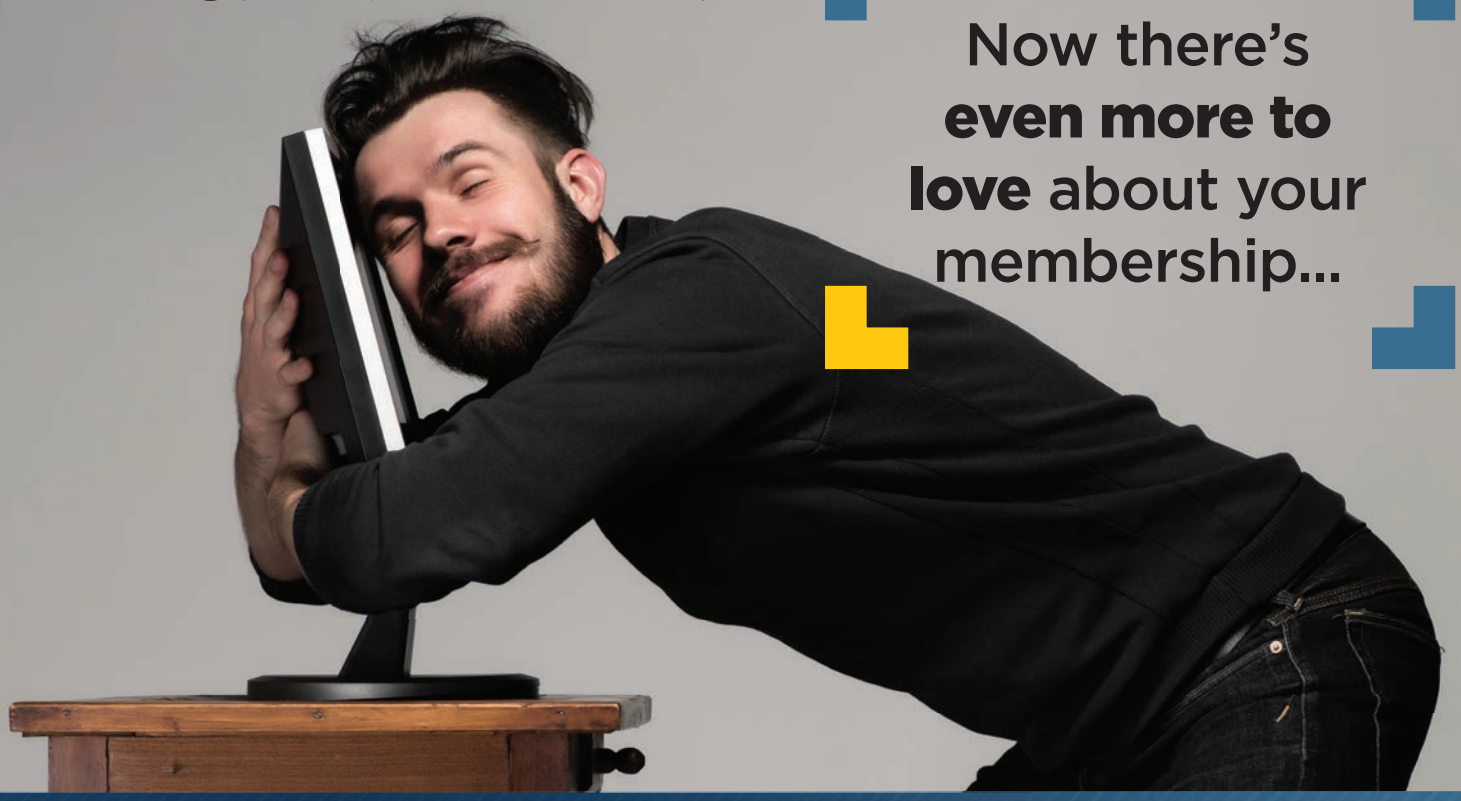
Rick Kuhn *is a computer scientist in the Computer Security Division of NIST. His current technical interests are in software assurance, access control and cybersecurity, and empirical studies of software failures. Contact him at kuhn@nist.gov.*

Mohammad S. Raunak *is an associate professor in the Computer Science Department at Loyola University Maryland. His current research interests are in testing "difficult-to-test" systems, software assurance, and validation quantification in simulation models. Contact him at raunak@gmail.com.*

Raghu Kacker *is a mathematical statistician in the Applied and Computational Mathematics division of NIST. His current technical interests are in software testing, and uncertainty quantification in computational models. Contact him at raghu.kacker@nist.gov.*

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

IEEE computer society



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

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

myCS

NO ADDITIONAL **FEE**

▶ON YOUR COMPUTER
▶ON YOUR SMARTPHONE
▶ON YOUR eREADER
▶ON YOUR TABLET

Introducing myCS, the digital magazine portal from IEEE Computer Society.

Finally...go beyond static, hard-to-read PDFs. Our go-to portal makes it easy to access and customize your favorite technical publications like *Computer*, *IEEE Software*, *IEEE Security & Privacy*, and more. Get started today for state-of-the-art industry news and a fully adaptive experience.

▶LEARN MORE AT: **mycs.computer.org**

Editor: Gary McGraw, gem@synopsys.com

Silver Bullet Talks with Wafaa Mamilli

Gary McGraw | Synopsys

Hear the full podcast and find show links, notes, and an online discussion at www.cigital.com/silverbullet.

Wafaa Mamilli is vice president, chief information security officer (CISO) at Eli Lilly and Company, a pharmaceutical company, where she leads a global enterprise-wide information and product security organization. Mamilli started her career consulting in Paris prior to joining Lilly France in 1995. Before being named CISO, she held several international leadership responsibilities across Lilly, including a stint as information officer of the Diabetes Division.

Mamilli embraces her international experience as she was born and raised in Morocco, and lived in France, the UK, and the Middle East, before relocating to Indianapolis in 2008.

Let's start with your diverse and very deep experience in multiple cultures. In your experience, do people approach technology itself and the management of technology differently in Europe versus the US?

Yeah they do. There are two dimensions. There is the risk aversion that plays in some places more than others. Entrepreneurial leapfrogging and leading the way I found more prevalent in the US. But more importantly, especially in the field I'm in now, the definition of privacy differs from one place to another.

Privacy is a right when you are in Europe. People fought for it over centuries, not just decades. In Europe, a company's information might not be seen as company information if it belongs to a person.

In information security, we have to take this into account and get to the right [information] for the right reasons for the business and, at the same time, understand where people are coming from, wherever they live or work.

You've been a CISO and vice president for 18 months, but you spent 22 years in a really distinguished career moving up from program management. How did you set and recalibrate your career path?

My father used to tell us (and I still tell my kids), "Learning is a never-ending process," and that helps with agility. I didn't know a lot early on about what I wanted to do with my career, but I knew a few things. I knew I wanted to be challenged, I knew I wanted to keep learning, and I knew I wanted breadth. I like the big picture, to understand the connections, so that's how I navigated my career.

I've been in many first-time jobs at Lilly. The first CISO at this level at Lilly, first at my previous job where I built our real-world evidence environment and hub for the company; in Europe, the same. So I spent time more focused on where I could learn, where I could foster my leadership capabilities. Because in the end, those are transferable skills.

I also spent a lot of time making sure I really understood the business. I understand how balance statements and profit and loss [P&L] work. I understand how the business leaders think. I get to the depth of the business we're in, and it's helped me throughout my career, because every job is preparing me for a job I didn't even know existed.

How easy has it been to learn "security"?

I really feel it's been the steepest learning curve of my life. I also had a big sense of urgency because I didn't have time to learn. They appoint you CISO, and you are the CISO. You are immediately accountable.

Also, security is the first field in my life where I really feel the ground is shifting under our feet every day. I've always worked on the discipline of strategy, translate strategy to execution, execution, and then iteration. Then you find speed and agility. But this is a world where you

 Published by the IEEE Computer Society 2469-7087/18/$33.00 © 2018 IEEE

have to be very agile, because you're learning something new every day. You know that road map you're telling your organization to be focused on executing—you might have to trade a few things, because something has changed. It's very, very fast paced.

And, after all of that, the risk is never zero.

When you set out to develop metrics in measuring security, what did you learn? I know that your approach to business management is very driven by metrics and measurement. What makes security different or the same?

Because of my previous jobs, I wanted to make sure that, on the metrics side, we're making a distinction between operational metrics and stuff that we (as information security organizations, internally, at different levels) use either to track operations or to make decisions versus what we're going to show to an executive team—where less is more.

I do think that there is a propensity left over from the old days of measuring stuff that doesn't really matter. It's just a measurement; it doesn't tell you the why.

Exactly. We have to be careful; sometimes you get a number, and if you're not clear on the why, you find yourself in a rabbit hole of a conservation. You've got to focus on who sees the numbers, and why they're seeing them, and what you're trying to convey, and then ensure consistency of your story. Whatever you are telling your audience, less is more; give them the insights, not just the information.

You already tipped your hand on this, but do you expect to stay in security, or is this another rung in the ladder up?

I don't know. I love security, more than I thought I would. I think about it as playing chess. I think of it as a business risk management job, and I'm loving that. I'm focusing on learning, building an organization that's inspired, making a difference not just at Lilly, but in the community as well, giving back. So we'll see.

You've studied business management at Harvard and at the London Business School. What was the most valuable thing you learned in those programs that you're applying today as CISO?

The biggest thing is how CEOs run businesses. As a matter of fact, I think all leaders in every company need to understand how the CEO runs the business. How do they look at the P&L? How do they look at the top line, bottom line? How do they calculate those ratios that Wall Street cares about, or not? When I'm speaking to a senior executive leader that I am selling something to, or I'm explaining a risk, I can take their lens, not because I'm an expert in what they do, but because I understand what they do. I've studied it, and I've applied it in different jobs. You have to master finance if you want to be a leader anywhere. And I think of myself as a business leader with information security accountability.

I know already the answer to this one, but do CISOs need more business school?

We all learn in different ways, but you need the skills. If you don't understand accounting, if you've never read your company's annual reports, you have to go do it. If you've never listened to an investor call (if your company is publically traded), go listen to that. Because then you'll see what's in the head of the shareholders and the pressures that your CEO is facing. Make sure you understand outside-in: what are the threats to your industry, and what's going on in your market?

Of course we have to understand InfoSec, but unless you're running a security company, which I am not, you have to understand the field. So I need to understand pharma, I need to understand healthcare, I need to understand the US market, the Asian markets, European markets, pricing pressures. And then after that, I need to make sure that I understand the information security field.

How did your work as information officer of the Diabetes Division differ from your work now as CISO?

It was very different, with a few similarities. In my information officer role, I had the privilege of

About Wafaa Mamilli

Wafaa Mamilli is vice president, chief information security officer at Eli Lilly and Company, where she leads a global enterprise-wide information and product security organization. Mamilli started her career consulting in Paris prior to joining Lilly France in 1995. Before being named CISO, she held several international leadership responsibilities across Lilly, including a stint as information officer of the Diabetes Division. Mamilli is multilingual and holds a master's degree in computer sciences from InSEA Engineering College in Rabat, Morocco, and a master's degree from IFSEC REN University in France, as well as a General Management Certificate from the London Business School. In 2015, Mamilli graduated from the Harvard Business School Advanced Management Program. She is married and lives with two kids in Indianapolis.

running our digital diabetes program. It was a lot of innovation and dream-thinking, and then of course delivery and operations. But there was a lot of ideation, and I worked very, very closely with the patients. That's the piece that is dramatically different.

We were launching at that time six drugs in 18 months, and we needed to make sure that we could speak about solutions with the patient. I needed to understand the drugs that were given to the patients. I studied the scientific side of how they operate. So it was very "core" to what the company is doing.

My job now has a different "core." When first I took this job, I wasn't expecting these kinds of responsibilities. I had to work on making the connection between the patient and this job, I had to feel it in my heart, in my belly.

I understand now: we're protecting the IP so that we can manufacture, we're protecting the manufacturing operations, we're protecting commercial pricing, so that we really can serve the patients.

And of course there is the nature of information security: you're not controlling anything in your environment. The risks come from the outside. Again that's ground shifting under your feet. I never had that, and I certainly did not have it in the information officer role.

Do you think that the CISO in general should report to the CIO or not? This is a big debate.

I think it can work either way. I do not report to the CIO today, but regardless of the reporting line, the biggest thing is partnership—and that was one of the first things I worked on when I took my job. You have to have a very strong, tight partnership with the IT organization, and we built that up.

Also crucial is the relationship with the CIO and the information officers, all of them, to make sure that you can have tough conversations and healthy debates. But the other side of my job as CISO, of any CISO, should be about an honest way of looking at risk management and elevating the risks as needed.

If there is trust between the CIO and the CISO, and the company has the right governance, when you have an IT issue, you're going to use your governance to make sure that the company knows about it. Let's say you're having an IT cyber-hygiene discipline issue. You need to be able to tell the company, "Hey we're having this issue and we're working on it." You can't *not* say it because you're working in IT.

It's not a question of reporting, it's a question of trust, governance, and capabilities of the two leaders.

The main philosophical conversation about this is whether you can audit the thing that you're in sufficiently, and your answer addressed that by saying, well it really depends on the politics and the governance.

Exactly. You have to be an honest and responsible leader. If you have an issue, you have to have the capabilities and the openness in your organization to say, "I'm here. My job is risk management. And yes, we have this risk, and we're struggling with patching."

In my CISO project work, we ran around the country interviewing 25 CISOs, (including you; thanks for being part of that). One thing that came up over and over was what I call the "missing middle management problem in security." Very briefly, we have some great executives and we have great worker bees, but a very thin rank of management in the middle. Have you noted this problem in the field?

Absolutely yes. I think it's historical. It's because security people grew through the technology ranks by way of infrastructure, not the business-facing side of technology in organizations. As a community, I don't think we have done a good job in people development, understanding that leadership is expected as well. It's not just about delivering the technology, it's about leadership, influence, communication, presentations, storytelling.

At Lilly, I'm taking my full middle management team or leadership team through education sessions and trainings on storytelling, presentations. We're also pulling people from other organizations so that we mix the diversity of ties and experiences. For example, bringing in a marketer to my team had an excellent impact in how we think about workforce customer experience. That's what marketers do—we need to bring that thinking to InfoSec as well.

A little slight change of gears. Women make up about 11 percent of the workforce in information security. What are you doing inside and outside of Lilly to develop and retain more women in the workforce in information security?

Excellent question. Well, first I do my duty by going and talking at high schools, colleges. I'm on the board of the Indiana University Cyber Security Program. We engage with the Purdue Series Program as well, helping them to bring diversity to the front.

I'm a member on the board of the Executive Women Forum [EWF], which does mentoring. I brought women from my organization there, and they are very engaged. I personally reviewed each of their development and career plans to make sure that we're having the right conversations.

To get more women in the field, we're going to have to market information security in a different way than when I was offered the job. I did not know then that information security was business risk management. We have to talk about it

that way. We have to explain to women and even girls early—I think from middle school—that, there are different opportunities. You don't have to have a hoodie. If you want to have a hoodie, fine, but you don't need one to come and work.

I was at a high school talking to a girl who was proficient in technology, and she was kind of studying the computer science class and said, "Well, I don't want to do anything like this because I want to talk to people." Because she's thinking that's all we're doing in InfoSec is looking at our computer and cables.

We also have to have more women who are visible in the field, and that's what we're trying to do with the EWF. I can tell you I connected with a lot of CISO women, and together we go to events, we mentor.

So, a funny question: I'm not sure how to put this properly, but do women have an easier or a harder time interfacing with senior leadership?
Generally speaking, it's stereotyping. It's interesting, something happens in the life of women—they tend to be better at communication and socializing when they're girls and teenagers. And something happens to them when we bring them to work where we make them more shy, less confident.

I think it's more of a question of confidence and opportunity, and we have to show women that, "Yeah, you can do a good job of integrating your personal life and professional life and have the career you are dreaming about."

Right. Well I think you're setting quite a great example so well done on that.
Thank you.

The last question has nothing to do with information security or business really. You've lived in some places with amazing cuisine in your life. What do you miss more: French cooking from Paris or North African food from Morocco?
You know, it's very, very simple, but I really miss the French baguette you can get from anywhere. And by the way, you eat half of it before making it home because you're going to be walking. Then from Morocco, it's the freshness of tagine. It's fine and fresh and all those carrots and tomatoes, and now I'm hungry and I'm very far from Morocco and from France.

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.

This article originally appeared in
IEEE Security & Privacy,
vol. 16, no. 1, 2018.

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

got flaws?

IEEE CYBER SECURITY

Find out more and get involved:
cybersecurity.ieee.org

IEEE CENTER FOR SECURE DESIGN

IEEE

IEEE computer society

Editors: Peter Gutmann, pgut001@cs.auckland.ac.nz | David Naccache, david.naccache@ens.fr | Charles C. Palmer, ccpalmer@us.ibm.com

Freedom of Encryption

Aisling Connolly | École Normale Supérieure

The year is 1943. You need a key. Deciding to keep it simple, you press A, a rotor turns, you take some paper and write K. Press B, write Q. Press C, write G. Again, press A, then B, then C. Write R, N, J. Next, you can begin communication, press W, write D and continue; press E, T, T, E, R, B, E, R, I, C, H, T, write OAJKXTQHETTI. You have your message. Move to your radio and transmit KQGRNJDOAJKXTQHETTI … and you've sent your first encrypted weather report. Does the thought ever arise in your mind as to whether or not it is dishonest to scramble your message? You do this for the sake of national security, for strategy in time of war, for your nation. You need ask no questions; this is your duty.

We jump to 1970. The height of the post-war, Golden Age of Capitalism. Electronic fund transfers (EFTs) are rampant, and the number of issued credit cards surpasses 1 million in the United States. The world's economy is booming. Life is sweet.

It's 1977. Recovering from the 1973–75 recession, you are more skeptical about EFTs. Data protection laws surrounding the collection of payment information are passed. You need more secure systems and welcome the development of DES. But to use it is no mean feat. What was once an instrument solely used for military advantage, encryption is now commercially required due to post-recession insecurities and the growth of electronic and computing industries, and is allowed only through the granting of special licenses.

Let's move to 1991. You possess your own Personal Computer. Imagine that! For the first time, you see the ability to encrypt moving into the hands of the citizen. This yields excitement, but also, it is immediately obvious that this will cause some consternation. On the one hand, the First Amendment of the US Constitution strongly protects freedom of speech and expression, which—in a roundabout way—means that cryptography within the US cannot be controlled. On the other hand, cryptography remains on the US Munitions List, meaning that its export is still heavily regulated.

The next decade sees some of the bloodiest years of the Crypto Wars. With global connection to the Internet, pressure mounts on the US government to loosen the laws surrounding the export of software. The battles are fought in court, and in 1996, encryption software is removed from the Munitions List. By the turn of the century, rules surrounding the export of commercial and open source software containing cryptography are greatly simplified, restrictions on keys are lifted, backdoors are prohibited, and you, the citizen, feel that progress is underway.

Fast-forward to June 2013. You sit happily tip-tapping on your smartphone, sharing Doge memes and giggling over screaming goats,

 Published by the IEEE Computer Society 2469-7087/18/$33.00 © 2018 IEEE

This article originally appeared in
IEEE Security & Privacy, *vol. 16, no. 1, 2018.*

before moving on to check the news. You learn that the US government has forced Verizon to hand over the phone records of millions of Americans. It's not such a nice story. Over the coming days you see more articles of a similar vein. You discover the NSA's direct access to data held by all your beloved Internet giants. You learn of secret programs and backdoors, and within months, you come to terms with the fact that you live in a quasi-surveillance state. This is a grave situation, and once again, as you did two decades ago, you find yourself debating the same disparity between individual privacy and state security.

Between then and now. The debate raged on and the disparity still exists. Several nations have the desire to forbid encryption, to keep it as a military tool, to stifle progression, to retain control, and to undermine democracy. But there has also been much progress. Within the coming year, the EU will put two new directives into place: the GDPR (General Data Protection Regulation) and the ePrivacy directive. They expand and solidify the points set out in the Charter of Fundamental Rights of the European Union and the Treaty on the Functioning of the European Union, which state that EU citizens have the right to privacy both online and offline. They insist that in order to maintain security of the individual while ensuring compliance with the regulation, appropriate measures (such as encryption) must be used. The regulations cover the collection, storage, processing, and deletion of personally identifiable information (PII) and personal communications. The regulation applies to the handling of EU citizens' PII irrespective of the location of the organization handling the data.

The United Nations Human Rights Council and the General Assembly have also specified the necessity for encryption to ensure the right to privacy in the digital age, building their argument by paying particular focus to the dangers faced by journalists. The UN[1]

> *[e]mphasizes that, in the digital age, encryption and anonymity tools have become vital for many journalists to exercise freely their work and their enjoyment of human rights, in particular their rights to freedom of expression and to privacy, including to secure their communications and to protect the confidentiality of their sources, and calls upon States not to interfere with the use of such technologies, with any restrictions thereon complying with States' obligations under international human rights law.*

These are simply two examples of many that build upon the arguments of yore. However strong were the efforts made by the privacy advocates in the 90s, they were very few voices. Now, with technology in every inch of our lives, with the increased media attention due to the Snowden revelations, high-profile court cases, and freer flowing information, citizens are much more aware of the consequences of not using encryption. This time around, there are many voices.

And so here we are. You have come a long way since your button-pressing days on the Enigma. You've seen four world recessions, men walking on the moon, the fall of the Berlin Wall, and some moves toward equality. You've danced to records, to cassette tapes, to mp3s, and now to Spotify. You must be tired. But you cannot sit yet! If I ask you to send me an encrypted email, right here and now, can you do it? If I ask you to remove any records of me, to grant my request to be forgotten, is it easy? If I want to travel, to meet people in the world, will you stop me at the border? Will you question me and demand my passwords? If I want to talk, to exercise a curiosity, to learn and teach and spread information, but without prying eyes, will you let me? Ultimately, all I'm asking is to exercise a right. Is it possible?

Until the answer to all of these questions is a definitive *Yes*, then I'm afraid we still have some work to do. ■

Reference

1. "The Safety of Journalists," UN Human Rights Council, A/HRC/RES/33/2, 6 Oct 2016; http://daccess-ods.un.org/access.nsf/Get?Open&DS=A/HRC/RES/33/2&Lang=E.

Aisling Connolly is a PhD candidate at the École Normale Supérieure. Contact her at aisling.connolly@ens.fr.

TECHNOLOGY

Oracle America, Inc.

has openings for

APPLICATIONS DEVELOPER

positions in **Chicago, IL**.

Job duties include: Analyze, design, develop, troubleshoot and debug software programs for commercial or end-user applications.

Apply by e-mailing resume to scott.bockelman@oracle.com, referencing 385.19595.
Oracle supports workforce diversity.

IEEE computer society

SkillChoice™ Complete

Now with expanded libraries and an upgraded platform!

Valued at **$3,300!**

3,000+ online courses

MENTORSHIP

28,000+ BOOKS

6,000+ VIDEOS

Practice Exams

15,000+ *Books24x7 titles*

OVER 20x as many resources as before

One membership. **Unlimited knowledge.**

Did you know IEEE Computer Society membership comes with access to a high-quality, interactive suite of professional development resources, available 24/7?

Powered by Skillsoft, the SkillChoice™ Complete library contains more than $3,000 worth of industry-leading online courses, books, videos, mentoring tools and exam prep. Best of all, you get it for the one low price of your Preferred Plus, Training & Development, or Student membership package. There's something for everyone, from beginners to advanced IT professionals to business leaders and managers.

The IT industry is constantly evolving. Don't be left behind. Join the IEEE Computer Society today, and gain access to the tools you need to stay on top of the latest trends and standards.

Learn more at **www.computer.org/join**.

ACCESS TO SKILLSOFT IS AVAILABLE WITH

CONNECT ON INTERFACE

Explore **INTERFACE**, a communication resource to help members engage, collaborate and stay current on Computer Society activities. Use **INTERFACE** to learn about member accomplishments and find out how your peers are changing the world with technology.

We spotlight our professional sections and student branch chapters, sharing their recent activities and giving leaders a window into how chapters around the globe grow, thrive and meet member expectations. Plus, **INTERFACE** will keep you informed on Computer Society–related activities so you never miss a meeting, career development opportunity or important industry update.

Connect today at interface.computer.org

IEEE COMPUTER SOCIETY
INTERFACE