Rethinking a Secure Internet of Things

http://iot.stanford.edu

The Internet will soon connect us to the physical world through personal health monitors, proximity networks, smart homes, smart cars, and automation networks. These new networks provide tremendous opportunity, but also bring tremendous risks. Today, hackers steal credit card numbers; tomorrow, they will be able to control your home, steal your personal medical data, and track you as you walk down the street. Existing approaches to secure computing systems are insufficient for these new cyber-physical applications, as they have very different trust models and network architectures, bridging pervasive local area networks, personal mobile devices, server storage, and web-based applications.

The Secure Internet of Things Project (SITP) is a 5-year collaboration between Stanford, UC Berkeley, and the University of Michigan to research fundamentally new, better ways to secure the Internet of Things and make them easy to use. The project seeks to answer three principal questions:

Analytics: how will we integrate these enormous streams of physical world instrumentation with all of our existing data?

Security: how can pervasive sensing and analytics systems preserve and protect user and system security?

Hardware and software systems: what hardware and software systems will make developing new intelligent and secure Internet of Things applications as easy as a modern web application?

These three efforts are tightly connected. Internet of Things (IoT) applications will need novel cryptographic protocols that are able to work on tiny, low-power devices yet also scale up to enormous stores of data in the cloud. Furthermore, IoT devices (such a smart lock) might remain in use for 10-20 years, so we must protect today's devices against attacks from 20 years in the future.

The principal investigators of the project met with collaborators and affiliates of the project for a 2-day retreat in June of 2016 to reflect on progress over the past year and lay out a research agenda for the second year of the project, summarized here in this document.

Project Goals

Our goal is to, over the five years of the project:

1. Research and define new cryptographic computational models and security mechanisms for Internet of Things devices to be secure for decades or more.
2. Research and implement secure, open source hardware/software frameworks to prototype and build Internet of Things applications that correctly use these new mechanisms.

First Year Research and Results

The first year of the project focused on three major research themes:

- 20-year security,
- application-level frameworks, and
- gateways.

20-year Security

The work on 20-year security has three major projects. The first is designing what a future embedded system-on-a-chip will need in terms of cryptographic primitives. We published an architectural design, consisting of a set of accelerators and hardware primitives to make ultra-low power micro controllers better able to evolve and adapt to changing security needs over a 20-year lifespan. The key insight in architecture, called CESEL, is that the economics

of modern system-on-a-chip (SoC) designs provides ample space for hardware accelerators and cryptographic engines. A next generation mote can therefore include many such co-processors and features at almost no production cost. The paper describes an initial design for what hardware security support such a device should have, focusing on five hardware primitives: an atomic, unique counter, a random number generator based on physical entropy, additional instructions to accelerate symmetric ciphers, an elliptic curve accelerator, and support for modular polynomial multiplication used in post-quantum cryptographic signing algorithms. We are currently collaborating with ARM to implement CESEL within a Cortex M0 core.

The second project is random number generation. Random numbers are the foundation of all cryptography and computer security. A first linchpin of a secure Internet of Things is fast and inexpensive ($ and board area) random number generation that anyone can easily incorporate into a device. We published a novel circuit for random number generation, called the Lampert circuit. Because the Lampert circuit is constructed out of simple hardware components, its operation is transparent and auditable. Using avalanche noise, a nondeterministic physical phenomenon, the circuit is inherently probabilistic and resists adversarial control. Further, because it compares the outputs from two matched noise sources, it rejects environmental disturbances like RF energy and power supply ripple. The resulting hardware produces more than 0.98 bits of entropy per sample, is inexpensive, has a small footprint, and can be disabled to conserve power when not in use.

The third and final project is the design and implementation of a new, secure embedded operating system. Embedded systems today are still typically written in low-level C. We learned in the end of the 20th century that this leads to buffer overflow attacks and many other security vulnerabilities. Desktop and server operating systems have since moved to use a variety of techniques and hardware mechanisms to protect against such attacks, but embedded processors do not have these luxuries. Our hypothesis is that using a type-safe systems language can provide a provably secure OS kernel that can allow multiple untrusted applications (e.g., apps loaded onto a smart watch) to run concurrently. We published our initial experiences implementing a safe and secure kernel in the Rust programming language, a new type-safe systems language. The operating system is completely open source, and we are planning a first public release, accompanied with a research hardware platform, this November.

Application Development Frameworks

A system is only as secure as its hardware and software. Most IoT applications follow a common "MGC" architecture, consisting of four parts: eMbedded devices, a Gateway device such as a mobile phone, and servers (in the computing cloud). These devices span 3-6 orders of magnitude in resources and each use their own languages, operating systems, and application frameworks. Developing these applications is hard: verifying and establishing security properties across this mish-mash of existing systems is even harder.

We have published several approaches to make developing IoT applications easier. The first, called Fabryq, builds on the observation that web scripting is a common starting point for programmers today. Being able to prototype an IoT-based application or user interaction entirely as a web program makes it much easier to explore and try initial approaches. The second, called Ravel, builds on the observation that model-view-controller (MVC) is the dominant web application programming model today, proposing that IoT applications be described in an *distributed* MVC architecture that is partitioned across embedded, gateway, and cloud devices.

To guide our research in application development frameworks, we simultaneously researched several IoT applications. The Toastboard, for example, is a "smart" breadboard for designing and testing simple electrical circuits, a first step in many Maker projects. Drill Sergeant is an example of how everyday objects, such as power tools, can be augmented with sensors and displays to show someone how to construct a physical object. For example, using Drill Sergeant, a drill tells the operator where to drill and when to stop, as well as where to cut with a chop saw, displaying this directly on the materials with a mini projector. Finally, the Haunted House explores how smart environments can combine with telepresence in order to allow distant people to interact unobtrusively.

The IoT Gateway

Our final research thrust focused on the middle tier of IoT applications: the gateway. These devices are simultaneously powerful yet also resource-constrained. Unlike an embedded device, for example, a mobile phone has gigabytes of storage and a powerful multicore processor. But unlike a cloud server, it runs on a battery so must manage its energy consumption.

Furthermore, gateways connect embedded devices using low-power link layers to the broader Internet. For example, Bluetooth Low Energy (BLE) is increasingly used for personal-area networks and smart objects. Gateways coordinate BLE networks, coordinating communication schedules and channel hopping. Current operating system support for Bluetooth Low Energy forces peripherals into vertical application silos. As a result, simple, intuitive applications such as opening a door with a smart watch or simultaneously logging and viewing heart rate data are impossible. We published Beetle, a new hardware interface that virtualizes peripherals at the application layer, allowing safe access by multiple programs without requiring the operating system to understand hardware functionality, fine-grained access control to peripheral device resources, and transparent access to peripherals connected over the network.

Beetle assumes that a gateway can discover Bluetooth devices around it. This raises an interesting question: is it possible to have BLE devices that can only be discovered by gateways that are authorized to do so? This is a chicken-and-egg problem: the device can't reveal itself first, because then unauthorized gateways might find out it exists (e.g., a thief could know what kind of smart lock you have). Similarly, the gateway can't reveal itself first, because then every BLE device could become a passive snooper for who passed nearby. We have published a novel, private discovery protocol based on name prefixed and identity-based encryption (IBE). Using this protocol, a gateway can discover a device, and a device can discover a gateway, if and only if both are authorized to do so.

Additional Work

While our work focused on the three research thrusts described above, we published over 20 additional papers on topics relevant to the Internet of Things and its security. The SITP website has a full listing of all of our published work.

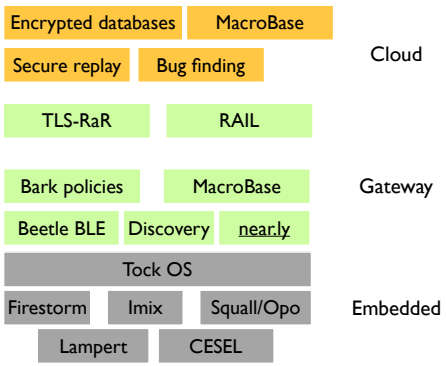
Second Year Plans

In our 2016 retreat, we identified three new research themes for the coming year. We will continue to research 20 year security, IoT gateways, and application development frameworks, and additionally begin exploring the following three topics:

1. a secure software stack,
2. smart maker tools,
3. and applications.

Secure Software Stack

In the past year, we have researched some of the foundations of a secure software stack for Internet of Things Applications. There are currently numerous projects tackling this problem at different levels of the stack, from Prof. Popa's work on encrypted databases, to Prof. Dutta's work on gateways and naming, to Prof. Bailis' work on data analytics. Our first goal for the next year is to begin to bring these projects together into a software stack, stretching from a secure embedded operating system (Tock) to auditable secure transport at the gateway (TLS-RaR) to interactive analytics (Macrobase) on secure databases (Arx). The figure on the right shows our first organization of these hardware and software components and how they might come together into a complete



stack.

Applications

Our second topic focus for the year is to build and deploy several applications using our software stack. There are several application projects already underway in the group, including smart small-scale manufacturing with Professors Engler and Horowitz, a water use sensing network at Stanford with Professors Levis, Horowitz, and Bailis, and smart home networks with Professors Dutta and Culler. Building full applications using components of the stack will guide its requirements and expose further research problems.

Smart Maker Tools

Our third topic focus for the year is smart maker tools. While the secure software stack focuses purely on the software side of security and applications, IoT systems are uniquely challenging due to their interplay between physical and digital components. An application involves not only software, but also building embedded devices, designing human interactions, and generally building physical things. We will explore new tools that tackle and help with problems that cross the digital and physical domains, building on our current work with the Toastboard and Drill Sergeant.

Staff and Faculty

Philip Levis
Faculty Director

Steve Eglash
Executive Director

Faculty

Peter Bailis is an assistant professor of computer science at Stanford University. Peter's research in the Future Data Systems group focuses on the design and implementation of next-generation data-intensive systems. His work spans novel distributed protocol design, large-scale data management, and architectures for high-volume complex decision support. He is the recipient of a NSF Graduate Research Fellowship, a Berkeley Fellowship for Graduate Study, best-of-conference citations for research appearing in SIGMOD and VLDB, and a CRA Outstanding Undergraduate Researcher Award. He received a Ph.D. in Computer Science from UC Berkeley in 2015 and an A.B. from Harvard College in 2011.

Dan Boneh is a Professor of Computer Science at Stanford University where he heads the applied cryptography group. Dr. Boneh's research focuses on applications of cryptography to computer security. His work includes cryptosystems with novel properties, security for mobile devices and the Internet of Things, Web security, and cryptanalysis. He is the author of over a hundred publications in the field and is a recipient of the Godel prize, the Packard Award, the Alfred P. Sloan Award, the RSA award in mathematics and five best paper awards. In 2011 Dr. Boneh received the Ishii award for industry education innovation.

David Culler is the Freisen Professor Electrical Engineering and Computer Sciences and Faculty Director of i4energy at the University of California, Berkeley. He is a member of the National Academy of Engineering, an ACM Fellow, an IEEE Fellow and was selected for the 2013 Okawa Prize, ACM's Sigmod Outstanding Achievement Award, Scientific American's 'Top 50 Researchers', and Technology Review's '10 Technologies that Will Change the World'. He has done seminal work on networks of small, embedded wireless devices, planetary-scale internet services, parallel computer architecture, parallel programming languages, and high performance communication, and including TinyOS, PlanetLab, Networks of Workstations (NOW), and Active Messages. He is currently focused on utilizing information technology to address the energy problem and is co-PI on the NSF CyberPhysical Systems projects LoCal and ActionWebs and PI on Software Defined Buildings.

Prabal Dutta is an Associate Professor of Electrical Engineering and Computer Science at the University of Michigan. He envisions a future in which the Internet of Things results in a trillion new wireless, embedded Internet hosts online within a decade or so. His research interests include how these devices and their software systems should be designed so that they survive and thrive in this not-too-distant future. More broadly, his research interests straddle the hardware/software interface and include embedded systems, networking, and architecture.

Dawson Engler is an Associate Professor in EE and CS at Stanford. His research focuses on techniques that automatically find serious errors in real code, ranging from system-specific static analysis, to model checking, to symbolic execution. His research group has won numerous ``Best Paper'' awards and its static checking work formed the basis of Coverity, recently purchased by Synposys. His group has submitted thousands of bug reports to the Linux kernel and core libraries such as libc; in some cases these bugs had been known but unfound for over a decade. He won the 2006 ACM SIGOPS Mark Weiser Award and 2009 Grace Hopper Award.

Björn Hartmann is an Associate Professor in EECS at UC Berkeley. He co-founded the CITRIS Invention Lab where he teaches classes in IoT product design and also co-directs Berkeley's Swarm Lab. His research in Human-Computer Interaction focuses on design, prototyping and implementation tools for the era of post-personal computing. As computation moves away from single-user desktop applications, he investigates how new algorithms, applications and design principles can support the creation of novel user interfaces. He has received a Sloan Fellowship, NSF CAREER, and an Okawa research award.

Mark Horowitz is the Yahoo! Founders Professor at Stanford University and was chair of the Electrical Engineering Department from 2008 to 2012. He co-founded Rambus, Inc. in 1990 and is a fellow of the IEEE and the ACM and a member of the National Academy of Engineering and the American Academy of Arts and Science. Dr. Horowitz's research interests are quite broad and span using EE and CS analysis methods to problems in molecular biology to creating new design methodologies for analog and digital circuits and systems.

Philip Levis is an Associate Professor of Computer Science and Electrical Engineering at Stanford University, where he heads the Stanford Information Networks Group (SING). His research centers on computing systems that interact with or represent the physical world, including low-power computing, wireless networks, sensor networks, embedded systems, and graphics systems. He has been awarded the Okawa Fellowship, an NSF CAREER award, and a Microsoft New Faculty Fellowship. He's authored over 60 peer-reviewed publications, including three best paper awards, one test of time award, and one most influential paper award. His research is the basis for Internet standards on how embedded devices connect to the Internet (RFC6550 and RFC6206). He has an Sc.B. in Biology and Computer Science with Honors from Brown University, a M.S. in Computer Science from The University of Colorado at Boulder, and a Ph.D. in Computer Science from The University of California, Berkeley.

David Mazières is a professor of Computer Science at Stanford University, where he leads the Secure Computer Systems research group. Prof. Mazieres received a BS in Computer Science from Harvard in 1994 and Ph.D. in Electrical Engineering and ComputerScience from MIT in 2000. Prof. Mazieres's research interests include Operating Systems and Distributed Systems, with a particular focus on security. Some of the projects he and his students have worked on include SFS (a self-certifying network file system), SUNDR (a file system that introduced the notion of fork linearizability), Kademlia (a widely used peer-to-peer routing algorithm), Coral (a peer-to-peer content distribution network), HiStar (a secure operating system based on decentralized information flow control), tcpcrypt (a TCP option providing forward-secure encryption), and Hails (a web framework that can preserve privacy while incorporating untrusted third-party apps).

Raluca Ada Popa is an assistant professor of computer science at UC Berkeley in 2015. She is interested in security, systems, and applied cryptography. Raluca developed practical systems (such as CryptDB and Mylar) that protect data confidentiality by computing over encrypted data, as well as designed new encryption schemes that underlie these systems. Some of her work has had early impact, with Google applying CryptDB's design to their SQL-like BigQuery service and surgeons at Boston's Newton-Wellesley hospital using Mylar to secure their medical application. Raluca is the recipient of a Google PhD Fellowship, Johnson award for best CS Masters of Engineering thesis from MIT, and CRA Outstanding undergraduate award from the ACM. Raluca has received her PhD in computer science as well as her two BS degrees, in computer science and in mathematics, from MIT.

Christopher Ré (Chris) is an assistant professor in the Department of Computer Science at Stanford University. His work's goal is to enable users and developers to build applications using analytics that enable them to more deeply understand and exploit data. Analytic techniques and tools from his group has been incorporated into scientific efforts including the IceCube neutrino detector and PaleoDeepDive, and into Cloudera's Impala and products from Oracle, Pivotal, Google Brain, and Microsoft's Adam.

Keith Winstein is an assistant professor of computer science at Stanford University. His work applies statistical and predictive approaches to teach computers to design better network protocols and applications. Winstein and colleagues created the State Synchronization Protocol and the Mosh (mobile shell) tool for remote access over challenged networks, the Sprout algorithm for transporting video over cellular networks, which was awarded a 2014 Applied Networking Research Prize, and the Remy system, in which computers design network protocols from first principles. He received a B.S. and M.Eng. in electrical engineering and computer science, an E.E., and a Ph.D. from the Massachusetts Institute of Technology.