



THE Next Wave

The National Security Agency's review of emerging technologies

Vol. 21 | No. 2 | 2016



[Photo credit: Bobboz/iStock/Thinkstock]



SPECIAL EDITION





THE Next Wave

The National Security Agency's review of emerging technologies

GUEST Editor's column

Rebecca J. Richards,
NSA's Director of Civil Liberties and Privacy

The concept of privacy is a deeply personal one. The ever-increasing incorporation of technology into our daily lives presents unique challenges in determining what we consider private and how we protect our privacy. However, these challenges should also be seen as opportunities to develop innovative solutions that mitigate potential privacy harms and unleash the promise and benefits of emerging technologies, such as those available through the Internet of Things.

A critical relationship exists between current cybersecurity research and the growing field of privacy engineering and research. Apart from security itself, the privacy implications of technology present distinct and different challenges. As a society, we must rely on security as a foundation from which the full concept of privacy can be both understood and protected. Privacy research focuses on more than mere technology and must be multidisciplinary, since the concept of privacy touches on many human elements and considerations.

Privacy is essentially a social construct that involves technical implementation. Therefore, technology-focused privacy research implies a need to understand the sociopolitical, philosophical, and legal contexts that drive the usage and adoption of emerging technology. With respect to the Internet of Things, which promises a ubiquity expected to fundamentally alter society's interaction with technology, we must account for the sociopolitical, philosophical, and legal aspects of privacy in tandem with the technical. This multidisciplinary research will greatly aid progress in understanding how to measure the real impact to individual privacy as technology advances at an increasingly rapid pace.

In this context, NSA sees itself as a facilitator, bringing together diverse people and ideas to foment multidisciplinary research, and perhaps even to develop a true science of privacy. The articles in this issue of *The Next Wave* illustrate exactly the kind of facilitation and innovation that NSA seeks to support. Be it unleashing the benefits of Big Data and the Internet of

Contents



Things or supporting initial efforts to bridge privacy research across the technical and social sciences, NSA aims to contribute positively towards safe, privacy-sensitive design and usage of technology as the Internet of Things emerges around us.

Rebecca J. Richards,
NSA's Director of Civil Liberties and Privacy

2 The Internet of Things: It's a wonderfully integrated life

8 Privacy in the Internet of Things
ROY DONG, LILLIAN J. RATLIFF

17 Security and the Internet of Things: When your refrigerator steals your identity
STAFF WRITER

22 Simon and Speck: Agile block ciphers for the Internet of Things
RAY BEAULIEU, DOUGLAS SHORS, JASON SMITH, STEFAN TREATMAN-CLARK, BRYAN WEEKS, AND LOUIS WINGERS

40 GLOBE AT A GLANCE: Smart Cities

43 FROM LAB TO MARKET: NSA's NiFi available via open source, improves flow of Big Data

44 POINTERS

The Next Wave is published to disseminate technical advancements and research activities in telecommunications and information technologies. Mentions of company names or commercial products do not imply endorsement by the US Government. The views and opinions expressed herein are those of the authors and do not necessarily reflect those of the NSA/CSS.

This publication is available online at <http://www.nsa.gov/thenextwave>. For more information, please contact us at TNW@tycho.ncsc.mil.

[Cover photo credit: Bobboz/iStock/Thinkstock]



THE INTERNET OF THINGS: IT'S A WONDERFULLY INTEGRATED LIFE

☞ A **SMART THERMOSTAT** can easily regulate your home heating, ventilation, and air conditioning (HVAC) unit based on learned temperature preferences you have established over a relatively short period of time.

☞ **SMART SMOKE DETECTORS** can provide direct alerts to smart devices for fire, smoke, and gas detection. Smoke detection units can work with the thermostat to turn off a gas furnace and monitor carbon monoxide levels in the home.

Smoke detectors fitted with motion sensors can act as night lights or emergency lighting in dark or smoky conditions [1]. In a completely integrated environment, home systems can signal one another and push alerts to designated smart devices. If activated, the smoke detection system can send a message to have the thermostat shut off fans and heating to prevent the spread of smoke throughout house, or it can send a signal to the lighting system for smart bulbs to turn a different color if high levels of carbon monoxide are detected [2, 3].

☞ **SMART LIGHTS** can help to manage energy use, reduce monthly electric bills, and act as part of your home emergency system by signaling paths for a safe exit. Smart lights can be controlled from anywhere in the home or remotely through a smart device.

☞ Your **SMART REFRIGERATOR** can help you conserve energy and maintain the refrigerator's internal temperature by presenting a list of contents for you to look over via the interactive screen on its door. The interactive screen also gives you the option to e-mail grocery lists to smart devices, sync the family schedule to the refrigerator's calendar, watch your favorite show or listen to music, and order and pay for your groceries through its connected pay system.

☞ Your **SMART HOME SECURITY SYSTEM** will enable you to view and receive information about your home while you are away. A smart home security system includes devices like smart doors and window locks, video cameras, and interactive doorbells that can be accessed and controlled from a smartphone or tablet.

[Photo credit: snapgalleria, Askold Romanov/iStock/Thinkstock]

A future where smart devices will seamlessly communicate with one another is quickly becoming reality. This article illustrates how consumers might interact in the near future with the IoT, all through the day. Manufacturers are developing hubs to handle multiple protocols, and organizations such as the ZigBee Alliance and Thread Group are working to make interoperable devices.

From the moment you wake up, the IoT will be running in the background of nearly every aspect of your life. Even when you are asleep, your devices will be on, working to enhance your world with data generated by sensors embedded in the wearables and appliances you use every day. So, what will the IoT do for you?

The IoT at Home

Installing systems that work together throughout your home network can help keep your family safe, enable better water and energy management, and save time by automating daily tasks. You can manage home appliances and entertainment with the swipe of a finger, the tap of a screen, or by voice command. Start the coffee machine from the comfort of your bed in the morning, run the dishwasher while you sit in traffic. Have the teakettle boil water while you drive home in the evening. Tell your smart TV you want to stop watching the movie and start reviewing your social media feed. Ask your artificial intelligence (AI) home assistant to give you morning news updates.

☎ Your **SMART SPRINKLER SYSTEM** can work with the sensors in your lawn to track moisture levels and the overall health of your soil. Combined with information from a weather application, the smart sprinkler system can determine whether to water the lawn and for how long, conserving water and saving money on your next bill.

☎ **SMART ENTERTAINMENT SYSTEMS** can be operated through a smart device or by voice command. Your smart TV can access the Internet, present your social media feeds, play music from your smart phone's playlist, and let you shop from your couch.

☎ The **SMART OVEN** can be started remotely and adjusted through an app on your smartphone that also allows you to preselect a baking temperature and cook time. You can keep an eye on dinner while away from the oven.

☎ Your **SMART DOORBELL** can send an alert to your phone when a package arrives at your doorstep. A smart doorbell can also send a picture or provide a live feed of visitors to your smartphone, allowing you the option to remotely unlock the door if necessary.

☎ Once the **SMART DOOR LOCK** is activated, a signal sent through your home network can turn a video security system on.

☎ **SMART HOME ASSISTANTS** can prepare a daily summary of your home's network, alerting you to any devices that require new batteries or require software updates. They can help with various functions around the house—like executing scheduled tasks for appliances (e.g., running your dishwasher), providing you with morning news highlights and weather forecasts during breakfast, or sending a signal to start the car so it warms up while you finish preparing for work. Once you leave and the smart door lock is activated, the smart home assistant can ensure the other home systems are completing their functions, like turning off the lights and adjusting the temperature.

[Photo credit: elenabs/iStock/Thinkstock]



The IoT on the Road



[photo credit: elenabs/Stock/Thinkstock]

☞ Through **VEHICLE-TO-VEHICLE (V2V) COMMUNICATION**, your car can identify other vehicles on the road to help avoid traffic accidents. V2V services will help identify vehicles located in blind spots and communicate important information multiple times per second, including the speed, direction, and distance of nearby vehicles.

☞ The **ONBOARD VEHICLE ASSISTANT** may connect with a parking services application and identify the best parking for you to access based on information from sensors in surrounding parking garages.

☞ Vehicles can contact **EMERGENCY SERVICES** as necessary. With smart systems on board, emergency services get immediate notification of accidents on the road along with location information that can help dispatch the closest emergency services teams and save critical time.

☞ **VEHICLE-TO-INFRASTRUCTURE (V2I) COMMUNICATION** will make use of onboard equipment to communicate with roadside equipment along a route to send and receive real-time information about road conditions, weather and traffic changes, and road closures and construction information.

☞ Your vehicle will also connect to **WEARABLE DEVICES** to learn about your physiological state—whether you are driving exhausted, stressed, or calm. Such information allows the vehicle to present services like cruise control to help you through your drive. The vehicle could also alert you to impending medical distress and slow the vehicle to a safe stop before contacting emergency services.

Once you hit the road, your onboard vehicle assistant will help plot your route. Even though the assistant knows where you work, changes may be made daily depending on weather, traffic, and road management updates.

☞ Your company may issue you a **WEARABLE DEVICE** for personal use, but the information can also help your organization learn about the health and activity of its workforce. Wearable devices can also be used as a form of access for employees. Some forms of authentication may rely on biometric information, like iris scans [4] for authentication purposes, but the data will be read and processed by IoT devices.

☞ IoT offers a new window into **CUSTOMER RELATIONSHIP MANAGEMENT (CRM)**. Data generated by connected devices offers unique insight at every step of the CRM cycle—from marketing and sales to customer support and feedback. Organizations will have access to critical information about the way customers use devices compared to intended device use, enabling enterprises to create better products for consumers [5].

☞ **INTELLIGENT HVAC SYSTEMS** will use sensors to determine hours of occupancy when running air conditioning and heating. Like in home systems, intelligent HVAC (heating, ventilation, and air conditioning) systems in large buildings can also work with fire and smoke detection devices to shut off fans or gas furnaces during emergencies.

☞ **SMART LIGHTING SYSTEMS** will operate similarly to home lighting systems. Motion sensors will enable lights in different parts of an organization to turn on and off based on employee occupancy—reducing the amount of wasted energy used in buildings with manual switches for lighting.

As technology advances, enterprises are harnessing the power of smart systems and sensors to gauge employee wellbeing and to maintain the overall health and security of their organizations.

[Photo credit: elenabs/iStock/Thinkstock]

The IoT at Work



[Photo credit: enaps, Ajsold Romanov/iStock/Thinkstock]



The IoT on You

The use of wearables extends beyond popular devices, like smart watches, to items like clothing (e.g., e-textiles, shoes, diapers), medical equipment, and jewelry that can act as an authentication token or a control for appliances.

- ☞ Other **SMART ACCESSORIES** can enable you to access information on demand. Smart glasses can take video, respond to voice commands, conduct searches on the web for information, and provide real-time information about your exercise—like your location, the distance traveled, and rate of speed. Smaller accessories, like smart rings, can serve as authentication tokens and simple controls for other devices around your home.
- ☞ **SMART WATCHES** and **FITNESS BANDS** can monitor your health, heart rate, and activity levels. These devices can provide you with daily updates on your health and performance and can send the information to your smart devices.

- ☞ **E-TEXTILES** come in all shapes and sizes. E-textile sensors can detect chemicals in the environment, thermal changes, perspiration levels, heart rates, and skin moisture among other variables.
- ☞ Thin adhesive **SMART PATCHES** can record your heart rate and transmit that information to your smartphone. The information gathered by your smart watch, fitness band, or smart adhesive strips can provide a more accurate health profile to your doctor.
- ☞ Wearables even include **INFANT CLOTHES**, such as smart onesies and diapers, that can be monitored through your smartphone. These enable new parents to track information like sleeping patterns, movements, and breathing, and can help detect problems like urinary tract infections [7].

IoT References

[1] Steele C. "19 Ways the Internet of Things changes everything." *PC Magazine*. 31 March 2015. Available at: <http://www.pcmag.com/slideshow/story/323778/19-ways-the-internet-of-things-changes-everything>.

[2] "Learn how Nest products work together" NEST. 01 September 2015. Available at: <https://nest.com/support/article/Learn-how-Nest-products-work-together>.

[3] Tuohy J. "What is home automation and how do I get started?" *Network World*. 2016 Jan 26. Available at: <http://www.networkworld.com/article/2874914/internet-of-things/what-is-home-automation-and-how-do-i-get-started.html>.

[4] Hoffman C. "Wearables 101: What they are and why you'll be seeing a lot of them." *How-To Geek*. 2015 Jan 15. Available at: <http://www.howtogeek.com/207108/wearables-101-what-they-are-and-why-youll-be-seeing-a-lot-of-them/>.

[5] Miller M. *The Internet of Things: How Smart TVs, Smart Cars, Smart Homes, and Smart Cities Are Changing the World*. Indianapolis (IN): Que. 26 March 2015.

[6] "EyeLock showcase iris authentication solutions for the Internet of Things." PR Newswire Association. 2016 Jan 5. Available at: <http://www.prnewswire.com/news-releases/eyelock-showcases-iris-authentication-solutions-for-the-internet-of-things-300199369.html>.

[7] "Gartner says CRM will be at the heart of digital initiatives for years to come." Gartner. 2014 Feb 12. Available at: <http://www.gartner.com/newsroom/id/2665215>.

Privacy in the Internet of Things

Roy Dong and Lillian J. Ratliff

The views and opinions expressed herein are those of the authors and do not necessarily reflect those of the NSA/CSS.



Today, personal data is being collected at unprecedented levels. This occurred initially in databases, where information was being aggregated from multiple sources including government records, web-browsing patterns stored in cookies, consumer loyalty programs, or credit card transactions. Now, however, the physical world is being outfitted with more and more sensors with ever-greater connectivity. The result is the immense network of interactive objects commonly referred to as the *Internet of Things* (IoT), which is collecting large amounts of real-time data about physical systems.

These new technologies require a new analysis of privacy. When data lived entirely in databases, the issues of concern were identity theft, public disclosure, and appropriation. Now, the IoT enables constant mass surveillance—revealing not just one or two facts about a person at a single point in time, but monitoring someone persistently to acquire multiple pieces of data. Studies have shown that ongoing monitoring can influence behavior even when the original behaviors were not illegal [1]; this is known as the *chilling effect*.

When developing this new analysis of privacy, we must first consider what conception of privacy we are trying to protect [2]. From an engineering and policy perspective, the predominant conceptions are *control over information* and *secrecy*.

- ▶ Control over information requires: (a) transparency to the data owner about what data is being collected and stored about him or her, (b) consent for this data to be transmitted to any parties, and (c) an ability to correct mistakes in the data.
- ▶ Secrecy focuses on: what new inferences can be made about a person due to the presence of data.

Throughout this article we will consider both conceptions of privacy and how to measure privacy with mathematical models.

First, privacy is inherently a *social* phenomenon and cannot exist outside of society, a community in which people prefer to withhold information from one another in certain circumstances. Remember, however, that different cultures and eras have defined privacy differently for various reasons, such as the technologies they use [2]. Therefore, to model privacy accurately, we must carefully consider the context in which it arises. What information is considered private today? Which adversaries would breach our privacy? What information do these adversaries have access to?

In this article, we explore two real-world examples—transportation networks and electric grids—and discuss ways to create privacy metrics that protect consumers' privacy as well as ways to quantify the utility of data. This allows us to determine whether the benefit of collecting data on, for instance, smart-grid operations, justifies consumer exposure to new privacy risks. Finally, we discuss how to design contracts that take into account data utility and privacy risks.

Mathematics of privacy

Because so much depends on context, it is not easy to quantify privacy via a general formulation. In connection with databases, for example, the most popular definition has been *differential privacy* [3]. Broadly, differential privacy considers the change in an individual's privacy as a result of participation in a database. More specifically, this form of privacy bounds how much the distribution of a database's output changes by adding or removing one database entry. Usually, differential privacy relies on adding "noise" or extraneous data to the output. A differentially private solution requires situations where some amount of noise is tolerable and where output is not very sensitive to changes in individual entries.

The IoT's complex nature, however, may make it difficult to measure privacy in such terms. In some

instances, adding noise may not be possible, or what constitutes a single data entry may be ambiguous, or certain kinds of data (e.g., billing records) may not be aggregated across multiple consumers and therefore may be very sensitive to changes in one consumer. Despite these difficulties, we will demonstrate how one can still quantify privacy in varying contexts.

Transportation systems

One context in which we have analyzed privacy is transportation systems [4]. In the “routing game,” for example, drivers have a set amount of flow allocated to each origin-destination, and choose routes based on previously observed traffic patterns. In this context, origins and destinations are considered private. From information about travel times and locations, an adversary can infer individuals’ itineraries and even details about their personal lives. This situation is not merely hypothetical; such capabilities are already available in the marketplace. Concerns about how vendors are using or misusing consumer data have already appeared in the media; for example, when an executive at the travel service Uber controversially asserted that he was able to use customers’ travel information to identify when they were having an affair [5].

We model how much the observable traffic patterns are affected by a small number of drivers changing their origins and destinations. We note that these observable traffic patterns are noisy in the sense that two drivers taking the same route will not experience the same exact travel time. This uncertainty is what gives rise to privacy—even with a fixed amount of traffic flow, some unpredictability still exists in the traffic patterns observed by an adversary.

More formally, we can let I denote the number of different origin and destination pairs. For each population k , let θ_k be a vector such that the i th entry represents the amount of drivers in population k that wish to travel along origin-destination pair i . We will say two population allocations θ and θ' are *adjacent* if there exists k^* such that $\|\theta_{k^*} - \theta'_{k^*}\|_\infty \leq c$ and $\theta_k = \theta'_k$ for any $k \neq k^*$. Intuitively, this states that the demand for origin-destination pairs is unchanged for all but one population, and this one population doesn’t change its demand for any origin-destination pair by more than c .

Given a population allocation, the drivers will decide how to allocate their flows among different paths based on some learning dynamics. The congestion on each link will depend on these flows, as well as some noise. We let $\ell(\theta)$ denote the random variable representing the observed traffic congestions across T days when the population allocation is θ .

We can extend the definition of differential privacy and apply it to this model. We say that our system is (ϵ, δ) -differentially private if, for any adjacent θ and θ' and any measurable set B , we have $P(\ell(\theta) \in B) \leq \exp(\epsilon)P(\ell(\theta') \in B) + \delta$. The parameters ϵ and δ quantify how private the system is; lower values for both parameters are more private. For intuition, note that if $\epsilon = \delta = 0$, then the origin-destination demands have no effect on the observable traffic flows, and this bound holds trivially when $\delta = 1$.

In our research, we can analyze what aspects of the model affect these two privacy parameters (see figure 1). Our theorem states that, under a general class of learning dynamics (i.e., how sensitive a driver is to his/her daily observed traffic congestion and how it affects his/her future route choices) with generalized step sizes η_t and a reasonable noise model with variance σ^2 , we have that the privacy parameter is $\epsilon = \mathcal{O}\left(I \frac{c}{\sigma^2} \sum_{t=1}^T \eta_t\right)$. As expected, privacy decreases as the noise on the observable traffic flow decreases, and as more measurements are received across time. In practice, this means that an adversary can infer more and more about the origin and destination of drivers from traffic congestion (i.e., a higher value of ϵ and δ) as the adversary measures for longer periods of time and as these congestion measurements contain more information about the number of cars on different paths.

Less obviously, we can see that privacy decreases as the number of paths increases. Also, if the learning dynamics are more sensitive to current traffic delays than to past traffic delays, then the system is less private. Finally, we can see the complex fashion in which that privacy degrades across time—this can serve as a prescription of how often populations must change their origins and destinations to preserve privacy.

Smart grid

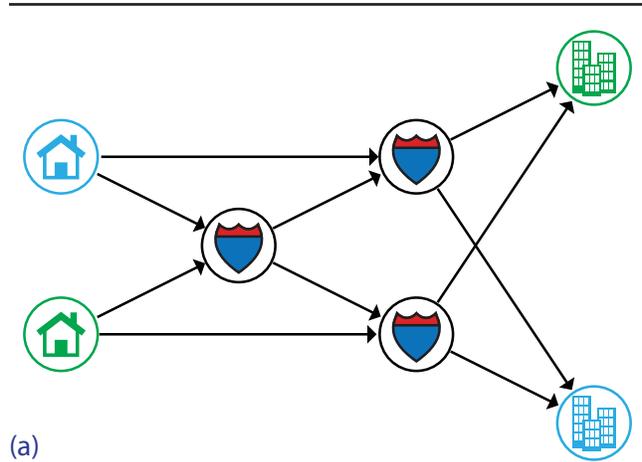
The electric power industry is another context in which the IoT is generating immense quantities of data, thanks to advanced metering that quantifies energy consumption with increasingly greater frequency. Although these measurements are limited to the household level, algorithms for energy disaggregation and nonintrusive load monitoring can recover device-level patterns of energy consumption based on individual power signatures [6]. Such capabilities, however, also enable pervasive monitoring of people’s movements inside their homes [7].

In the smart-grid context, it is not feasible to measure only data aggregated across several households because power companies bill for consumption at the individual household level. Similarly, additive noise would require electricity bills to be random variables, which is often not acceptable. Thus, we must find a way to analyze privacy inherent in the household itself rather than by injecting noise into the data.

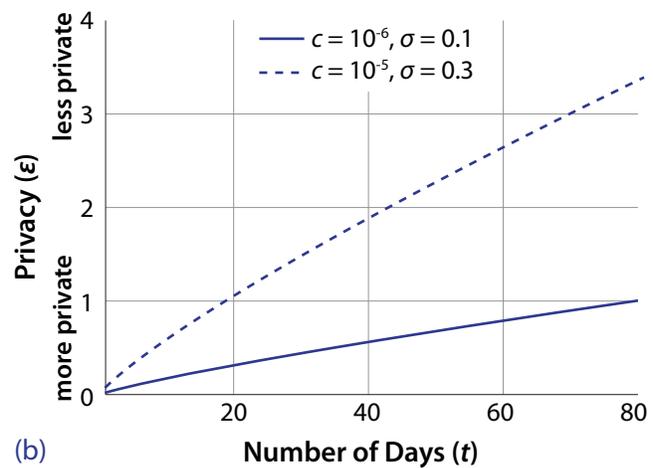
Privacy concept

We introduce, therefore, the concept of *inferential privacy* [7]. This is defined as a bound on the probability that an adversary will be able to infer a private parameter, regardless of the inference algorithm used. More specifically, we assume an adversary has access to information about devices in the households and their signatures. We also consider a fundamental limit of the energy disaggregation problem given this information set.

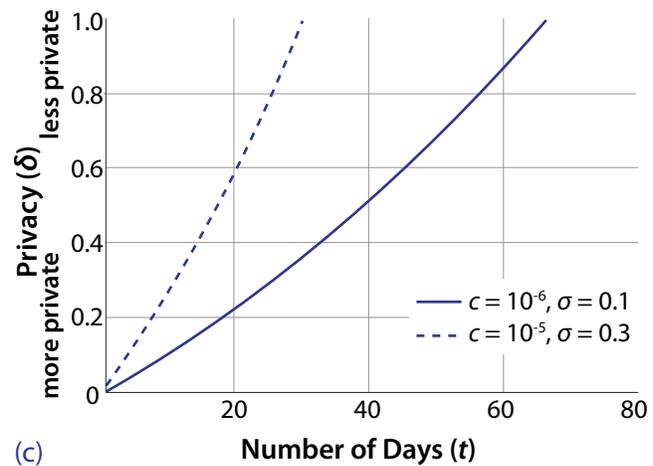
Formally, we suppose each use has a private parameter θ which takes values in a finite set Θ . For example, θ could denote whether or not a consumer is an energy saver or an energy waster, or whether a consumer cooks his or her own dinner. We suppose some distribution across the private parameter, so $\theta \sim \pi$. This private parameter influences device usage patterns, which we denote as u . For example, an energy waster is likely to leave the fridge door open; whereas, if a consumer consistently orders takeout, the stove top will not be used. So, given the private parameter θ , we say that the consumer’s device usage follows the distribution p_θ . Formally, we say $u|\theta \sim p_\theta$. Finally, we consider the device models. Device usage patterns will determine the power consumption of the devices, denoted y .



(a)



(b)



(c)

FIGURE 1. (a) For this small example network, the level of privacy decreases as more traffic observations are available, for differential privacy parameter ϵ (b) and for differential privacy parameter δ . (c) In fact, we can see the point at which there is no privacy at all for drivers ($\delta = 1$). (In (b) and (c), the dotted line represents privacy when more drivers are switching origins and destinations; in other words, the dotted line represents more variable traffic than the solid line represents.)

Given how the devices are being used, power consumption does not depend on the private parameter. Mathematically, we say $y|u, \theta \sim q(\cdot|u)$. This yields a hierarchical Bayes model of energy consumption.

An adversary observes y and has knowledge of π , p_θ , and $q(\cdot|u)$, and is attempting to infer the private parameter θ . As such, we define inferential privacy as follows. Our system is α inferentially private if, for any estimator $\hat{\theta}$, we have $P(\hat{\theta}(y) \neq \theta) \geq \alpha$. This estimator can depend on π , p_θ , and $q(\cdot|u)$. Thus, inferential privacy gives a guarantee of privacy in the parameter θ regardless of the inference methods our adversary uses.

In this hierarchical Bayes setting, we can find an optimal estimator. That is, the estimator $\hat{\theta}$ that maximizes $P(\hat{\theta} = \theta)$ is given by:

$$\hat{\theta}_{MAP}(y) = \underset{i}{\operatorname{argmax}} \left(\pi(i) \cdot \int p_i(u)q(y|u)du \right)$$

Thus, the system is α inferentially private, with $\alpha = P(\hat{\theta}_{MAP}(y) \neq \theta)$.

Although this value itself is often intractable to calculate in most applications, we can find approximations by leveraging testing bounds. Using Le Cam's method [8], we can derive a theorem that guarantees inferential privacy with α determined by the pairwise total variation distance between distributions. Alternatively, using Fano's method [9], we can guarantee inferential privacy with α determined by the number of different possible values of θ and the Kullback-Leibler divergence between distributions [10]. In both cases, we have found guarantees of inferential privacy that can easily be calculated, even when the optimal estimator cannot.

When your refrigerator tattles on you

As a simple example, let us focus on the energy consumption of a refrigerator. Consider the case where energy consumers are characterized as "wasters," "average consumers," or "savers." For this example, suppose that consumers consider this parameter to be private. The parameter determines how fastidious consumers are about leaving their fridge door open, which in turn determines how long the compressor has to cycle to cool the air inside the fridge. Putting this model into our hierarchical Bayes framework, we can use our testing bounds to calculate how the

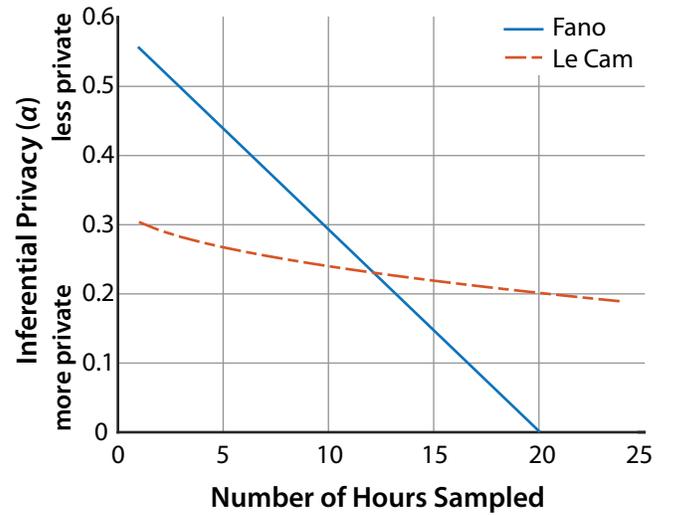


FIGURE 2. We can simulate energy consumption patterns for different consumers and quantify the privacy as a function of how many hours we sample. As the number of hours sampled increases, inferential privacy increases.

inferential privacy value changes as one records data across time (see figure 2).

Utility of data

An intelligent transportation system, enabled by numerous sensors in the IoT, could greatly improve roadway conditions through better tolling, ramp metering, and traffic light policies. The smart grid, using infrastructure for advanced metering, could improve load forecasting as well as exploit demand flexibility to enable operators to introduce efficient and renewable energy sources.

It is important to analyze the utility-privacy trade-off in collecting more data. This requires ways to quantify both the privacy risks inherent in the collection and the utility. For the former, the previous section allows us to analyze the privacy due to different measurement schemes. In this section, we discuss the issue of utility.

In the case of the smart grid, the National Institute of Standards and Technology has issued a data-minimization principle stating that electricity companies should only collect data needed for smart-grid operations [11]. We propose a framework to analyze how much data is required for smart-grid operations to run effectively [10].

HVAC systems: Trading energy efficiency for privacy

As an example, consider direct-load-control programs that use thermostatically controlled loads. Heating, ventilation, and cooling (HVAC) systems have a thermal inertia, which allows building operators to precool buildings and essentially use the stored thermal energy as a battery. This means the HVAC energy demand is deferrable, and smart-grid operators can exploit this situation to correct imbalances in load prediction or integrate renewable energy sources.

To guarantee the comfort of people inside the building, however, the direct-load controller must estimate the building's thermal dynamics [12]. As the controller receives fewer measurements, estimates get worse and reduce the controller's performance [11]. We can directly simulate that to determine how often the controller needs to get updates on the building's thermal state in order to control HVAC energy demand. Similarly, we can analyze how our privacy metrics decrease as the sampling rate increases. Unifying these two analyses, we can explicitly calculate the utility-privacy trade-off (see figure 3).

Privacy contracts

As already established, privacy is inherently a social phenomenon. But privacy preferences vary from consumer to consumer, and therefore we model privacy as a good that individuals value differently. Formally, we propose an economic solution that allows the power company to balance the trade-off between utility and privacy by combining privacy metrics with privacy-based service contracts.

In this framework, we assume the power company does not know the consumer's privacy preferences and that these preferences are distinct from private information that is subject to a privacy breach. We shall refer to the consumer's privacy preferences as his or her *type*. When the power company does not know how a consumer values the goods for which he or she is paying, we say the power company faces a problem of *adverse selection*. This can lead to a nonoptimal contract, under which power is allocated inefficiently.

We model the consumer's type as a parameter ξ that takes values in a finite set $\Xi = \{\xi_1, \xi_2, \dots, \xi_n\}$ where $\xi_i < \xi_{i+1}$ for each $i \in \{1, 2, \dots, n-1\}$. The contract

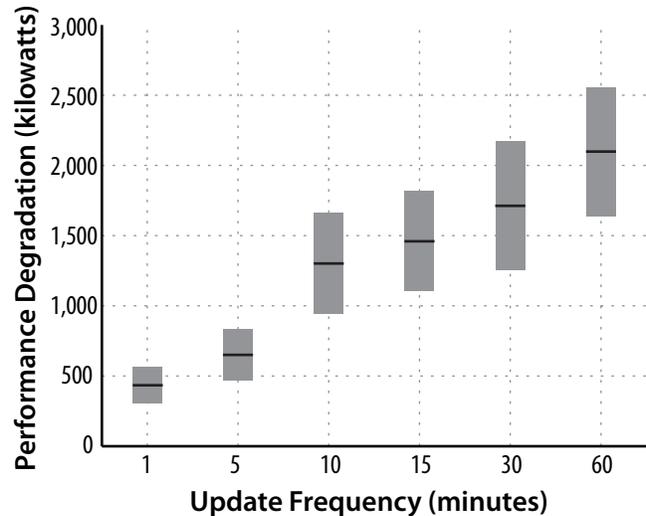


FIGURE 3. We can estimate the effect of different sampling policies on the performance of direct-load-control programs. If we suppose that the cost of electricity is roughly constant in the spot market (i.e., the purchase of electricity for immediate delivery), then this plot depicts the cost in dollars of different sampling policies: As the sampling period increases, the cost of inefficiencies increases.

consists of a privacy setting x on a smart meter, offered at a price t . Each type has a privacy setting, denoted x_i for type ξ_i . The privacy setting could be, for instance, a sampling rate or amount of noise injection. Similarly, we define t_i as the price for type ξ_i .

Taking a Bayesian approach, we assume the power company has a prior over the type space Ξ . Formally, we assume the power company faces a consumer of type ξ_i with probability p_i and $\sum_{i=1}^n p_i = 1$. Thus, the power company must design the menu of contracts $((x_1, t_1), (x_2, t_2), \dots, (x_n, t_n))$ in order to maximize their expected profit subject to the consumer voluntarily participating and truthfully reporting his or her type (i.e., selecting the contract designed for their type). We let $g(x_i)$ denote the unit cost of implementing privacy setting x_i ; it could for instance, be proportional to the error in the direct-load-control scheme discussed previously. Then, we will denote the expected profit as $\Pi((x_i, t_i)_{i=1}^n) = \sum_{i=1}^n p_i (t_i - g(x_i))$.

The power company wishes to maximize $\Pi((x_i, t_i)_{i=1}^n)$ by choosing $(x_i, t_i)_{i=1}^n$. However, the company must satisfy the individual rationality constraint, which ensures that consumers will voluntarily participate, and the incentive compatibility constraint, which ensures that customers will choose

the correct contract. Formally, let $U(x, \xi)$ denote the utility function of the consumers, which depends on the privacy setting and their type. The individual rationality constraint becomes $U(x_i, \xi_i) - t_i \geq 0$ for each i , and the incentive compatibility constraint becomes $U(x_i, \xi_i) - t_i \geq U(x_j, \xi_j) - t_j$ for all i, j .

The existence of several types leads to a large number of constraints, yet it is easy to reduce the set of constraints [13]. We can prove that we only need to consider the individual rationality constraint for the lowest type ξ_1 , which holds with equality: $U(x_1, \xi_1) - t_1 = 0$, and the local downward incentive compatibility constraints, which also hold with equality: $U(x_i, \xi_i) - t_i = u(x_{i-1}, \xi_i) - t_{i-1}$ for each $i \in \{2, \dots, n\}$. This results in a much more manageable problem.

Furthermore, we can make reasonable assumptions on the form of the utility function $U(x, \xi)$ and get a number of qualitative insights. For example, by solving the optimization problem and comparing to the socially optimal contract (i.e., the one that maximizes the sum of the power company's profit and the consumer's utility), we find that the consumers with the highest valuation of privacy ξ_n get the socially optimal privacy setting, yet pay much less than is socially optimal (i.e., these consumers free-ride on the rest of society).

In fact, because of the existence of the lowest type ξ_1 , all other types experience a positive result in the form of information rent—that is, they pay less than is socially optimal for their privacy setting. We can also show that the lowest type gets a socially inefficient allocation (i.e., the privacy setting received is lower than is socially optimal).

More realistically, we should consider that consumers may be wary of risk. A more complete model would have a consumer's utility given by $U(x_i, \xi_i) - t_i - (1 - \eta(x_i))\ell(\xi_i)$, where $\eta(x_i)$ is the probability of a privacy breach, depending on the privacy setting chosen, and $\ell(\xi_i)$ is the amount of loss experienced by consumers of type ξ_i when their privacy is breached. Here, bounds on $\eta(x_i)$ can be given by both the differential privacy and inferential privacy metrics previously provided.

We can study the impact of privacy loss risk on the optimal contract as a function of the losses experienced by each of the types and the prior across types.

First, we note that the optimal contracts, when we ignore risk, will violate incentive compatibility and individual rationality for the risk-averse customers. The lowest type will opt-out by not selecting a contract at all and will have an incentive to mask its type and select a contract designed for one of the higher types. In addition, the risk-averse contract suffers the same inefficiencies that we described above (i.e., the highest type gets free rides, and the lowest type gets zero surplus). More precise details on the theoretical results are in [14].

Furthermore, in the case where we consider risk, the privacy setting for the highest type ξ_n increases, independent of the prior beliefs on types. The privacy setting for other types decreases with respect to the prior beliefs on types, and whether the privacy setting increases or decreases with the introduction of risk depends on the losses experienced by each type and the prior beliefs. We can further characterize the optimal contracts given a risk-averse consumer as compared to a risk-neutral consumer by examining the losses experienced by each of the types and prior beliefs across types [13][14]. For certain distributions of types, particularly ones with a larger probability of high types, social welfare decreases with the introduction of privacy loss risks. The same is true for the power company's profit. Thus, the presence of privacy loss risks provides the power company an incentive to invest in insurance or security.

In this framework, security and privacy are tightly intertwined. More specifically, any security measure taken by the power company not only reduces its profit but also modifies the privacy metric $\eta(x_i)$. We have made initial efforts to understand insurance investment by both the consumer and the power company [14]; however, much remains to be done in terms of understanding the balance between security and insurance. Further, inefficiencies with respect to social welfare motivate an investigation into regulation mechanisms (e.g., subsidies or privacy taxes).

In summary, we have introduced privacy-based service contracts to capture the fact that different consumers value privacy differently, and to balance the utility-privacy trade-off. We have also developed a number of qualitative insights about how social welfare and efficiency are affected by privacy preferences in the population of consumers. At the core of this framework, we combined privacy metrics (detection

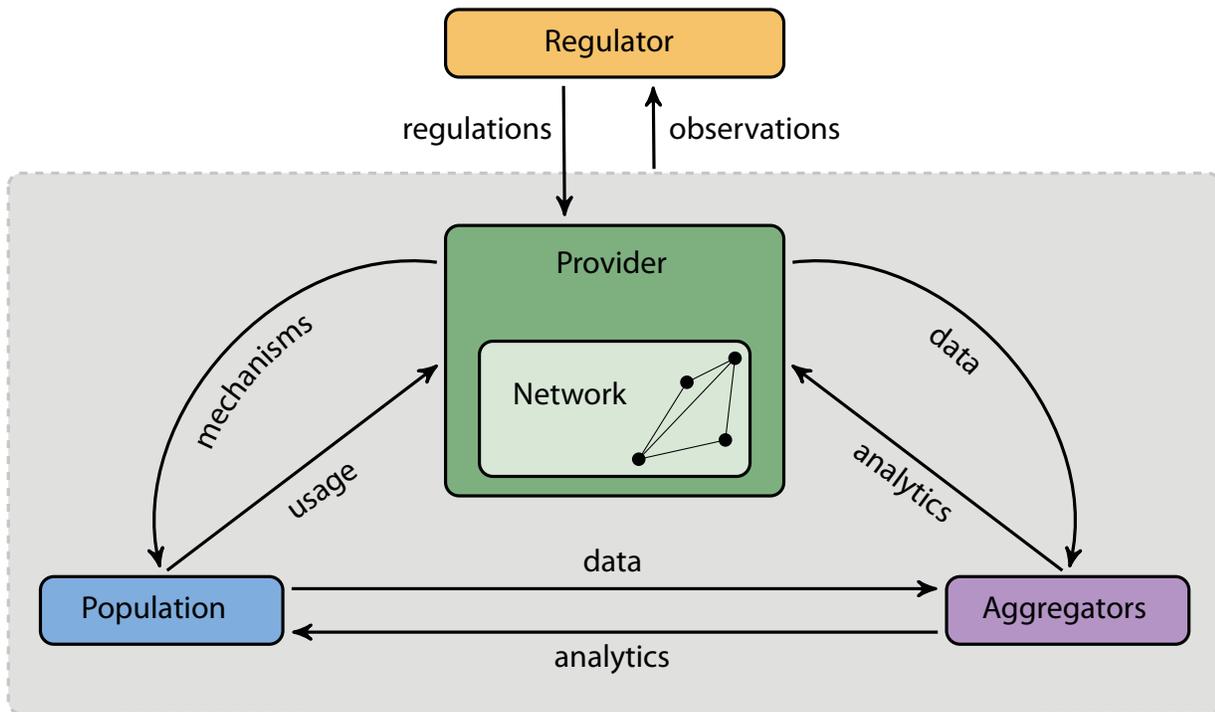


FIGURE 4. In future work, we hope to explore an ontology of the different actors in the IoT and understand the regulatory structure and incentives that interplay to create the data market.

theory) with economic tools (game theory) in order to consider not only preferences across privacy but also the following information exchanges: (a) between consumer and power company (e.g., where hidden preferences can cause adverse selection) and (b) between consumer/power company and adversary (privacy metrics).

Next, we discuss the importance of considering such information exchanges more generally in the IoT, as it is being used to facilitate and improve operations within critical infrastructure systems.

Data market

Thus far, we have considered the structure of markets to be fixed. However, privacy is a social phenomenon supported by interactions among a variety of agents with differing values, behaviors, priorities, and available information. To understand how privacy evolves in the IoT context, we need to understand the incentives of these agents, the regulatory structure that

restricts their actions, and the information available to agents.

In addition, we have analyzed the utility-privacy trade-off in the IoT. As noted, however, privacy is a complex phenomenon, highly dependent on contextual factors and operating across multiple dimensions. We are currently working on connecting the privacy metrics, utility of data analysis, and privacy contracts together in a framework in which we understand the interactions between different categories of actors (see figure 4). By building this ontology, we can move beyond a utility-privacy trade-off consideration of privacy and begin analyzing the sociotechnical system that is evolving from the enabling technologies of the IoT. [🔗](#)

About the authors

Roy Dong is a PhD candidate in electrical engineering and computer sciences at the University of California, Berkeley (UC-Berkeley). He received a BS in computer

engineering and a BS in economics from Michigan State University and is the recipient of the National Science Foundation (NSF) Graduate Research Fellowship. His research focuses on privacy risks and other vulnerabilities that arise as physical systems are networked and distributed.

Lillian Ratliff is a postdoctoral researcher at UC-Berkeley, where she also received her PhD in electrical engineering and computer sciences. She received her BS in mathematics, BS in electrical engineering, and MS in electrical engineering from the University of Nevada, Las Vegas. She is a recipient of the NSF Graduate Research Fellowship. She will be a postdoctoral researcher at UC Berkeley from July 2015 to September 2016 and an assistant professor in electrical engineering at the University of Washington, Seattle starting in the Fall of 2016. Her research focuses on modeling and understanding new vulnerabilities in societal-scale cyber-physical systems, such as intelligent energy, transportation, and healthcare systems. Her work combines tools from game theory and statistical learning to design economic and physical controls that help balance the trade-off between system efficiency and vulnerability.

References

- [1] Oulasvirta A, Pihlajamaa A, Perkiö J, Ray D, Vähäkangas T, Hasu T, Vainio N, Myllymäki P. “Long-term effects of ubiquitous surveillance in the home.” In: *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing, UbiComp’ 12*, 2012. doi: doi=10.1.1.278.7664.
- [2] Solove DJ. “Conceptualizing privacy.” *California Law Review*. 2002;90(4):1087–1155. Available at: <http://scholarship.law.berkeley.edu/californialawreview/vol90/iss4/2>.
- [3] Dwork C. “Differential privacy.” Microsoft Research. 2006. Available at: <http://research.microsoft.com/pubs/64346/dwork.pdf>.
- [4] Dong R, Krichene W, Bayen AM, Sastry SS. “Differential privacy of populations in routing games.” 2015. Submitted to *2015 IEEE Conference on Decision and Control* (under review).
- [5] Tufekci Z, King B. “We can’t trust Uber.” *The New York Times*. 2014 Dec 7. Available at: <http://www.nytimes.com/2014/12/08/opinion/we-cant-trust-uber.html?smid=pl-share>.
- [6] Dong R, Ratliff LJ, Ohlsson H, Sastry SS. “Energy disaggregation via adaptive filtering.” In: *51st Annual Allerton Conference on Communication, Control, and Computing (Allerton)*; 2013. doi: 10.1109/Allerton.2013.6736521.
- [7] Lisovich MA, Mulligan DK, Wicker SB. “Inferring personal information from demand-response systems.” *IEEE Security and Privacy*. 2010;8(1):11–20. doi: 10.1109/MSP.2010.40.
- [8] Le Cam L. “Convergence of estimates under dimensionality restrictions,” *The Ann. of Statistics*. 1973;1(1): 38–53. Available at: <http://www.jstor.org/stable/2958155>.
- [9] Yu B. “Assouad, Fano, and Le Cam.” In: *Festschrift for Lucien Le Cam*. 1997, pp. 423–435. doi: 10.1007/978-1-4612-1880-7_29.
- [10] Dong R, Cárdenas AA, Ratliff LJ, Ohlsson H, Sastry SS. “Quantifying the utility-privacy tradeoff in direct load control programs.” 2015. Submitted to *IEEE Transactions on Smart Grid* (under review).
- [11] National Institute of Standards and Technology. *Guidelines for Smart Grid Cybersecurity: Volume 2- Privacy and the Smart Grid*. 2014. doi: 10.6028/NIST.IR.7628r1.
- [12] Mathieu L, Koch S, Callaway DS. “State estimation and control of electric loads to manage real-time energy imbalance.” *IEEE Transactions on Power Systems*. 2013;28(1):430–440. doi: 10.1109/TPWRS.2012.2204074.
- [13] Ratliff LJ, Dong R, Ohlsson H, Cárdenas AA, Sastry SS. “Privacy and customer segmentation in the smart grid.” In: *IEEE 53rd Annual Conference on Decision and Control (CDC)*, 2014. doi: 10.1109/CDC.2014.7039714.
- [14] Ratliff LJ, Barreto C, Dong R, Ohlsson H, Cárdenas AA, Sastry SS. “Effects of risk on privacy contracts for demand-side management.” 2014. Submitted to *IEEE Transactions on Smart Grid* (under review).

Security and the Internet of Things: When your refrigerator steals your identity

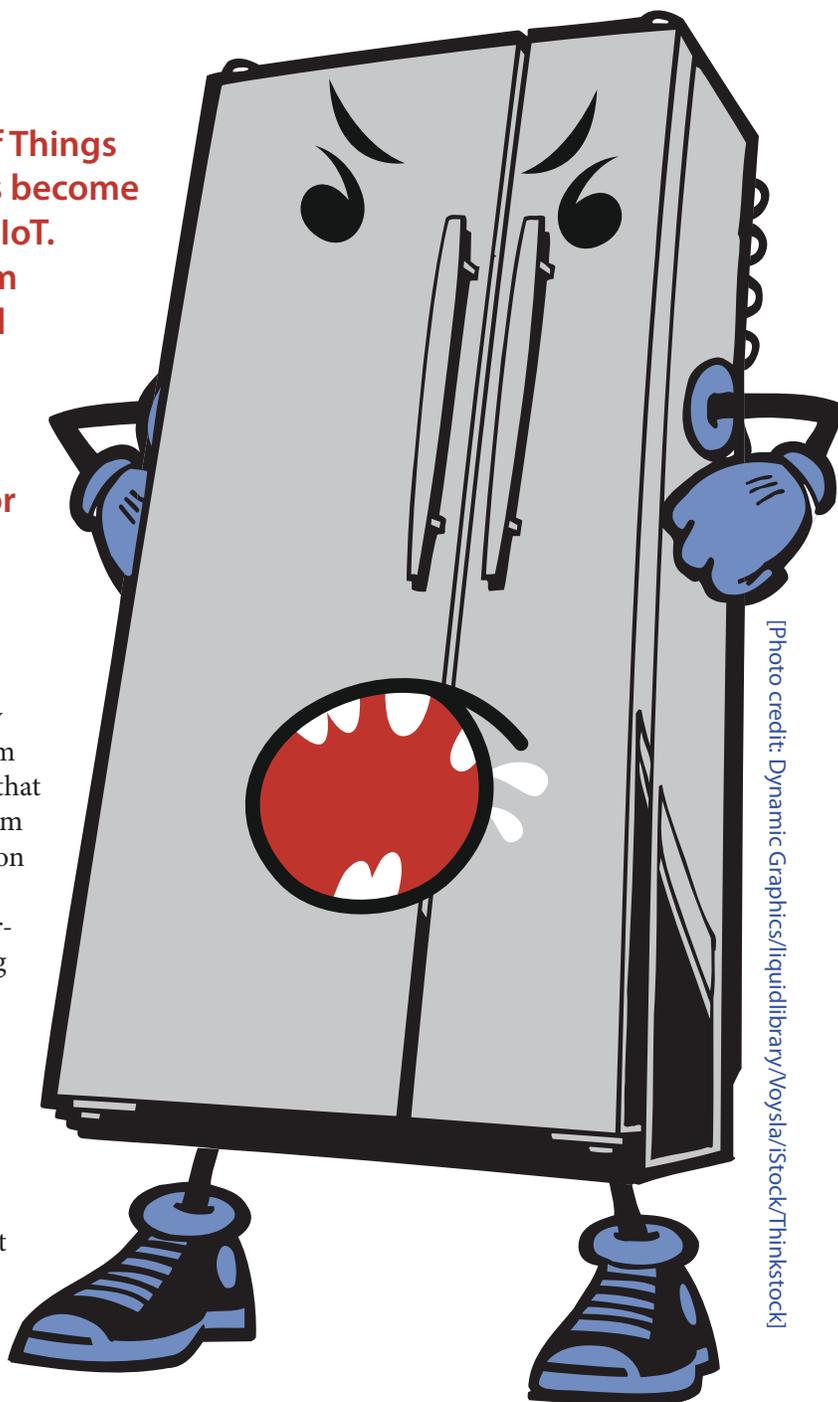
Staff Writer

The often maligned Internet of Things (IoT)-enabled refrigerator has become a symbol for insecurity in the IoT.

Although at first glance it may seem laughable that a normal household appliance could be used to launch a cyberattack, the fact is that nontraditional connected devices are becoming the new attack vector for hackers.

The state of IoT security

IoT adoption is forecast to increase dramatically over the next few years. The market research firm International Data Corporation (IDC) predicts that the installed base of IoT endpoints will grow from approximately 9 billion in 2013 to about 28 billion in 2020. This represents a growth rate of 17.5% through 2020, see figure 1 [1]. IoT devices generate vast amounts of data representing everything from medical information to electricity consumption. Protecting the integrity of IoT networks, however, seems to be an afterthought for some device vendors. The current IoT security landscape is reminiscent of the early days of the Internet with most players focused on producing minimally viable products at the expense of security. Many devices do not have security built in, while the ones that do may be hampered by poor update practices and the lack of a unified security standard.



[Photo credit: Dynamic Graphics/liquidlibrary/Voystar/Stock/Thinkstock]

In June 2015, the Open Web Application Security Project (OWASP) published a top ten list of IoT vulnerabilities that underscored the need for a holistic approach for protecting IoT devices. The security pitfalls uncovered by OWASP highlight security failures spanning the IoT ecosystem from the sensor to the user. The number one vulnerability cited was insecure web interfaces. This vector is easily exploitable with the potential to severely impact IoT networks due to data loss, denial of access, and device takeover. Attacks used in this scenario include cross-site scripting (XSS) and Structured Query Language (SQL) injection. The complete list of vulnerabilities includes [2]:

1. Insecure web interface,
2. Insufficient authentication/authorization,
3. Insecure network services,
4. Lack of transport encryption,
5. Privacy concerns,
6. Insecure cloud interface,
7. Insecure mobile interface,
8. Insufficient security configurability,
9. Insecure software/firmware, and
10. Poor physical security.

A 2014 study by Hewlett-Packard (HP) outlined the importance of a security management plan for the IoT. HP released the results of security scans that concentrated on 10 IoT devices including televisions, home thermostats, hubs, home alarms, and door locks. On average, HP found approximately 25 vulnerabilities per device including lax password practices, insecure web interfaces, unencrypted data transfer, and inadequate software protection. Many of the vulnerabilities were found on solutions from companies with no information security background. Also, HP found that 90% of the surveyed devices collected at least one piece of personal data via the device, the cloud, or a corresponding mobile application [3, 4].

Although there are a few devices that perform firmware updates automatically, such as Google's Nest Thermostat, most IoT endpoints have no official avenue for updating firmware or software—leaving devices vulnerable or placing the responsibility of updating devices on consumers. Push notifications for IoT users may not be an option because smaller manufacturers may not have the capacity to push upgrades due to ignorance of the process, limited security

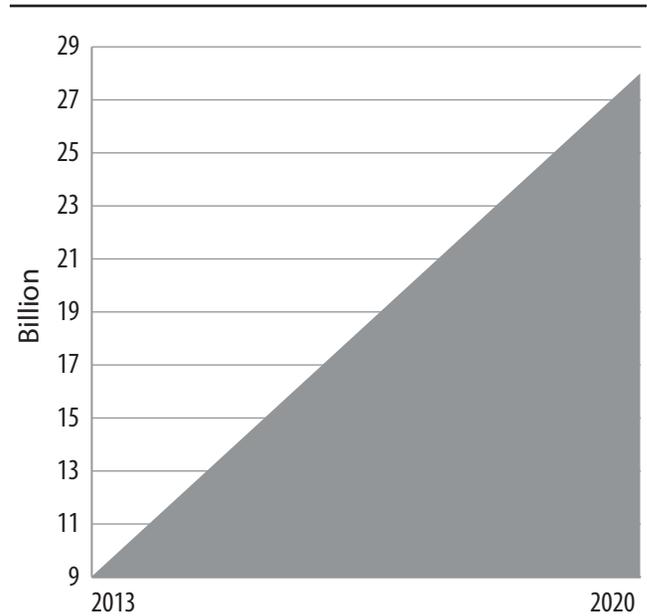


FIGURE 1. IDC predicts that the installed base of IoT endpoints will grow from approximately 9 billion in 2013 to about 28 billion in 2020, representing a 17.5% growth rate through 2020 [1].

knowledge, or a lack of infrastructure needed to track user purchases. To deal with the update issue, the Chief Security Officer for In-Q-Tel, the Intelligence Community's strategic investment firm, has suggested that IoT devices either be programmed to "die" at a predetermined time or call for updates regularly as a way to take the burden off of the user [3, 5, 6, 7].

An overarching theme to IoT security seems to be that most devices lack the computing horsepower to accommodate security software. Simplicity, and therefore low cost, is one of the drivers for the high rate of adoption for IoT devices. Simple, purpose-built endpoints are inexpensive, and adding on security management software could reduce adoption by increasing prices and decreasing functionality.

Due to the constraints of IoT devices, a one-size-fits-all approach to security management may not be feasible. Tailoring security options for computing platforms in the home was manageable because of the limited number of connected devices (i.e., laptop, tablet, and smartphone). However, with everything becoming connected, this scenario is quickly becoming unwieldy. Until market-wide security practices are adopted, users may need to weigh the risk of not securing their IoT endpoints against the impact that nonsecure devices will have on their network. For

example, a power supplier may want to secure the transmissions of devices deployed in the grid, but protecting a refrigerator's cyber integrity might not be a high priority.

A viable option for securing IoT endpoints was introduced in 2013 by NSA researchers. The researchers published an online paper describing two families of block ciphers—Simon and Speck. The ciphers are free and open-source algorithms developed specifically for the IoT, meeting the constraints of an IoT solution. Both ciphers can be used for hardware and software; however, Simon is optimized for hardware, while Speck works better with software. Since being published in 2013, the ciphers have not been broken, and both have been submitted to the International Organization for Standardization (ISO) for inclusion in the ISO 29192-2 standard [8]. The next article in this issue of TNW focuses on Simon and Speck; see page 22.

Rogue refrigerators—threats to home networks

As consumers continue to connect IoT devices to their home networks, the aperture for hackers to attack or launch attacks from local networks increases. As discussed above, most IoT devices, especially in the consumer market, are insecure by default, meaning that they come with little to no security options. The devices that do have security “baked in” often require the consumer to take initiative for patches and upgrades to software and firmware, because of vendors' generally lack of a reasonable upgrade path.

In January 2014, security firm Proofpoint claimed to have discovered a group of consumer devices enlisted into a botnet to send spam. Proofpoint described this network as consisting of approximately 100,000 everyday devices (including televisions and one refrigerator) transmitting more than 750,000 malicious e-mails. While many experts have recommended skepticism about this particular claim, they also maintain this type of botnet is theoretically plausible [9].

Further proof of the existence of an IoT botnet surfaced in September 2014 when *Network World* reported on a malware kit called Spike that could infect IoT devices and amass them into botnets. These Spike-created botnets were supposedly responsible for

distributed denial-of-service (DDoS) attacks in Asia and the US, with one attack peaking at 215 gigabits per second and 150 million packets per second. Antivirus companies cited Linux, Windows, and ARM-based Linux as susceptible to Spike, with the ARM variant able to infect IoT devices [10].

The need for a viable, streamlined method for IoT security updates was dramatically demonstrated by a highly publicized 2014 incident involving the malicious hacking of a baby monitor. A family in Ohio was terrorized by an unknown voice shouting at their daughter, and discovered that the voice emanated from a webcam in the child's room. Foscam, the company that sold the monitor, had warned users a year earlier to upgrade the camera's software and change the default password, but this advisory had not trickled down to the Ohio family. In September 2015, information security firm Rapid7 released a study assessing the security of several baby monitors. Of the seven monitors tested, all exhibited some level of vulnerability ranging from information leaks to privilege escalation. After identifying the vulnerabilities, Rapid7 contacted the vendors to report the issue [11, 12].

Telematics—threats to vehicles

In January 2016, market research firm Gartner stated that by 2020 approximately 250,000,000 connected vehicles would be in use worldwide. This represents approximately 10% of an expected 25 billion total connected things by 2020, making cars one of the larger representative samples of the IoT ecosystem. For many, including the National Highway Traffic Safety Administration, this underscores the need for automakers to secure networked vehicles as more and more new vehicles are Internet-enabled [13]. In July 2015, two security researchers remotely hacked a Chrysler Jeep Cherokee. A week later, researchers disclosed the ability to breach General Motors' OnStar system—unlocking car doors, starting the ignition, and accessing the owner's e-mail.

Not all attempts at hacking vehicles have been as “successful” as these two examples. Tesla has apparently survived security researchers' probing a little better than most. Researchers found the Tesla S exhibited some low-level vulnerabilities, but did not believe taking complete control of the vehicle was possible because a gateway separated the entertainment system

(the launch point for the Jeep attack) from the more vital systems. Another area where Tesla has outmaneuvered its competitors has been in patching the exposed flaws. While Chrysler has required owners to bring in their vehicles for updates, Tesla has pushed an automated update to customers over the air [14, 15].

The Jeep hack is important because it represented a worst-case-scenario of how a connected vehicle could be taken over. Hacking the Jeep's infotainment system gave the security researchers access to the Controller Area Network (CAN) bus, a system that connects the many intelligent systems in a vehicle. Once the researchers owned the infotainment system, they were able to inject code and gain root privileges, which allowed them to add code to the firmware to send commands to multiple critical systems. The researchers were able to kill the transmission while the vehicle was in operation as well as manipulate the brakes. Even without being able to gain access to the CAN bus, the researchers were able to use their access to the infotainment system to manipulate the radio and wipers and to track the car via Global Positioning System (GPS) data. The most disturbing revelation was that all of this was done remotely over Sprint's network [15].

These incidents follow an established path of insecurity that many IoT products and implementations travel down—time-to-market over security. However, the automobile industry is making a bid to make connected cars more secure, by creating the Intelligence Sharing and Analysis Center (ISAC). The ISAC will allow automakers worldwide access to information on vulnerabilities and cyber threats to vehicles and associated networks [13, 16].

Pacemakers and CT scanners—threats to medical devices

One area of the IoT that is causing major concern for security professionals is connected medical devices. Not only could these devices create a backdoor into a hospital's or a user's network, but hacker control of a connected device could be directly hazardous to your health. Hackers are not simply stealing credit card data from the "Bank of Wherever, USA," they are also frequently stealing medical information. Hackers supposedly value medical data much more than credit card data. Not only can the information be used for blackmail, but in the case of device takeover scenarios, for the deployment of life-threatening ransomware.

Wired magazine recently compiled a list of what it deemed as some of the most concerning connected medical products that hackers may target including drug infusion pumps, insulin pumps, and CT scanning equipment. Malicious actors who gain access to these types of devices could alter the amount of drugs, insulin, or radiation that a patient receives, with deadly consequences. In July 2015, the Food and Drug Administration (FDA) released an alert for Hospira's Symbiq Infusion System that warned of potential cybersecurity vulnerabilities associated with this infusion pump. However, the alert stated that neither the FDA nor Hospira was aware of any compromises associated with the pump. Finding exploitable medical devices is apparently not that difficult. In some instances, security researchers were able to locate vulnerable medical devices by using Shodan, a search engine for Internet-connected devices. Search terms like "radiology" and "podiatry" uncovered vulnerable devices. Flaws exhibited by discovered devices included configuration errors and unchanged default passwords [17, 18, 19].

Cybersecurity firm TrapX refers to the hacking of medical devices as MEDJACK and released a report citing attacks on equipment such as X-ray equipment, communications systems, and blood gas analyzers (BGAs). In one instance, TrapX found that attackers had used BGAs to gain access to hospital networks and exfiltrate data. The researchers also found the Zeus and Citadel malware variants on the network. Many of the devices were running out-of-date operating systems, which presented an avenue for exploitation. Also, although hospitals use firewalls and anti-virus software, IT professionals are not able to detect an attack because most medical devices are closed systems. This leaves the task of securing the devices up to the manufacturer [20].

Conclusion

The diverse IoT product ecosystem encompasses anything from toys to smart meters, and in theory any object can be "chipped" and made IoT-enabled. This vast device landscape provides vendors with a lucrative revenue stream and hackers with a broad attack surface for enterprise and consumer networks. In the world of IoT, computers are disguised as everyday things and accordingly are treated like everyday things. When we buy a smart refrigerator for

instance, our first thought is not “Hey, how do I adjust the security settings on this?” but instead “Hey, how do I adjust the temperature in the crisper?” This new generation of computing devices is not thought of as computers, and so securing the data that they collect and or process is often overlooked. 

References

- [1] IDC. “IDC market in a minute: Internet of Things.” 2014 Jun 02. Available at: http://www.idc.com/downloads/idc_market_in_a_minute_iot_infographic.pdf.
- [2] Open Web Application Security Project (OWASP). “OWASP Internet of Things top ten project.” 2014 Jul 30. Available at: https://www.owasp.org/index.php/OWASP_Internet_of_Things_Top_Ten_Project.
- [3] Burt J. “Securing billions of IoT devices poses mind-boggling challenges.” *eWeek*. 2014 Jul 31. Available at: <http://www.eweek.com/networking/securing-billions-of-iot-devices-poses-mind-boggling-challenges.html>.
- [4] Hewlett Packard. “Internet of Things research study.” 2015 Nov. Available at: <http://www8.hp.com/h20195/V2/GetPDF.aspx/4AA5-4759ENW.pdf>.
- [5] Jackson Higgins K. “4 Hurdles to securing the Internet Of Things.” *Dark Reading*. 2014 Sep 04. Available at: <http://www.darkreading.com/informationweek-home/4-hurdles-to-securing-the-internet-of-things/d/d-id/1306978>.
- [6] Wigle L. “Securing the Internet of Things.” *Dark Reading*. 2014 Dec 11. Available at: <http://www.darkreading.com/partner-perspectives/intel/securing-the-internet-of-things/a/d-id/1318072>.
- [7] Hill K. “The half-baked security of our ‘Internet Of Things.’” *Forbes*. 2014 May 27. Available at: <http://www.forbes.com/sites/kashmirhill/2014/05/27/article-may-scare-you-away-from-internet-of-things/#120f901e23dd>.
- [8] Swedburg C. “NSA offers block ciphers to help secure RFID transmissions.” *RFID Journal*. 2015 Jul 17. Available at: <http://www.rfidjournal.com/articles/view?13288>.
- [9] Kharpal A. “Can your fridge be hacked in the ‘Internet of Things?’” *CNBC*. 2014 Feb 21. Available at: <http://www.cnbc.com/2014/02/21/can-your-fridge-be-hacked-in-the-internet-of-things.html>.
- [10] Greene T. “Bot-herders can launch DDoS attacks from dryers, refrigerators, other Internet of Things devices.” 2014 Sep 24. *Network World*. Available at: <http://www.networkworld.com/article/2687169/security0/bot-herders-can-launch-ddos-attacks-from-dryers-refrigerators-other-internet-of-things-devices.html>.
- [11] Stanislav M, Beardsley T. “HACKING IoT: A case study on baby monitor exposures and vulnerabilities.” 2015 Sep 29. *Rapid7*. Available at: <https://www.rapid7.com/docs/Hacking-IoT-A-Case-Study-on-Baby-Monitor-Exposures-and-Vulnerabilities.pdf>.
- [12] The Economist. “Home, hacked home.” 2014 Jul 12. Available at: <http://www.economist.com/news/special-report/21606420-perils-connected-devices-home-hacked-home>.
- [13] Gartner. “Gartner says by 2020, a quarter billion connected vehicles will enable new in-vehicle services and automated driving capabilities.” 2015 Jan 26. Available at: <http://www.gartner.com/newsroom/id/2970017>.
- [14] Schultz R. “Putting the brakes on car hacks: An IoT primer.” *Venture Beat*. 2015 Sep 21. Available at: <http://venturebeat.com/2015/09/21/putting-the-brakes-on-car-hacks-an-iot-primer/>.
- [15] Hoopes H. “DEF CON focuses on vehicle security and beyond.” *Gizmag*. 2015 Aug 10. Available at: <http://www.gizmag.com/vehicle-hacking-trends-def-con-23/38858/>.
- [16] Jackson Higgins K. “Automobile industry gears up for cyber-threat intel-sharing.” *Dark Reading*. 2015 Jul 14. Available at: <http://www.darkreading.com/vulnerabilities--threats/automobile-industry-gears-up-for-cyber-threat-intel-sharing/d/d-id/1321304>.
- [17] Zetter K. “Medical devices that are vulnerable to life-threatening hacks.” *Wired*. 2015 Nov 24. Available at: <http://www.wired.com/2015/11/medical-devices-that-are-vulnerable-to-life-threatening-hacks/#slide-1>.
- [18] U.S. Food and Drug Administration. “Cybersecurity vulnerabilities of Hospira Symbiq infusion system: FDA safety communication.” 2015 Jul 31. Available at: http://www.fda.gov/MedicalDevices/Safety/AlertsandNotices/ucm456815.htm?source=govdelivery&utm_medium=email&utm_source=govdelivery.
- [19] Nicolai J. “Thousands of medical devices are vulnerable to hacking, security researchers say.” *PCWorld*. 2015 Sep 29. Available at: <http://www.pcworld.com/article/2987813/thousands-of-medical-devices-are-vulnerable-to-hacking-security-researchers-say.html>.
- [20] Storm D. “MEDJACK: Hackers hijacking medical devices to create backdoors in hospital networks.” *Computer World*. 2015 Jun 8. Available at: <http://www.computerworld.com/article/2932371/cybercrime-hacking/medjack-hackers-hijacking-medical-devices-to-create-backdoors-in-hospital-networks.html>.



Simon and Speck: Agile block ciphers for the Internet of Things*

Ray Beaulieu,
Douglas Shors,
Jason Smith,
Stefan Treatman-Clark,
Bryan Weeks, and
Louis Wingers

[Photo credits: Fuse, wishblazer/iStock/Thinkstock]

Abstract

The US National Security Agency (NSA) developed the Simon and Speck families of lightweight block ciphers as an aid for securing applications in very constrained environments where Advanced Encryption Standard (AES) may not be suitable. This paper summarizes the algorithms, their design rationale, along with current cryptanalysis and implementation results.

Introduction

Biologists make a distinction between specialist species, which occupy narrow ecological niches, and generalists, which can survive in a broader variety of environmental conditions. Specialists include Kirtland's warbler, a bird that only nests in 5–20-year-old jack pine forests, and the koala, which feeds (almost) exclusively on eucalyptus leaves. Generalists such as the American crow and the coyote are able to adapt to a variety of different environments. In a stable world, it's a good strategy to specialize, but when conditions change rapidly, specialists don't always fare so well.

The new age of pervasive computing is nothing if not rapidly changing. And yet, in the world of lightweight cryptography, specialists abound. Of course there are important research challenges associated with optimizing performance on particular platforms, and the direction taken by many in the field has been to take on such challenges, generally quite successfully. This can involve optimizing with respect to the instruction set for a certain microcontroller, or designing algorithms for a particular application-specific integrated circuit (ASIC) application (e.g.,



with hard-wired key or for IC printing), or designing specifically for low-latency applications, and so on.

We would argue that what's needed in the Internet of Things (IoT) era is not more Kirtland's warblers and koalas, as wonderful as such animals may be, but crows and coyotes. An animal that eats only eucalyptus leaves, even if it outcompetes the koala, will never become widely distributed. Similarly, a block cipher highly optimized for performance on a particular microcontroller will likely be outcompeted on other platforms, and could be of very limited utility in 15 years when its target platform is obsolete.

Of course it's hard to get a handle on block cipher performance on devices that don't yet exist. But what we can do is strive for *simplicity*, by designing algorithms around very basic operations that are certain to be supported by any future device capable of computation. Simon and Speck aim to be the sort of *generalist* block ciphers that we think will be required for future applications in the IoT era.

It would be unsatisfactory if we had to defer any discussion of performance because we're waiting for the arrival of future devices. But we can measure performance on current platforms, and in this paper we demonstrate the sort of performance that is achieved by Simon and Speck on a broad range of existing software and hardware platforms. We emphasize, however, that the main point is not the performance of Simon and Speck with respect to other algorithms on any particular platform. Rather, it's that by limiting the operations we rely on to a small list that works well in hardware and software, we obtain algorithms that are likely to perform well just about anywhere.



*This paper was presented on 20 July 2015 at the Lightweight Cryptography Workshop sponsored by the National Institute of Standards and Technology (NIST).

AES and lightweight cryptography

Before focusing our discussion on Simon and Speck, we'd like to better establish the state of play. In particular, we note that quite a lot of effort has gone into reshaping the current go-to block cipher, AES, into a solution for lightweight applications. Indeed, great strides have been made in this direction in the past 15 years or so. ASIC implementations of AES-128 have been developed with an area of just 2400 gate equivalents (GE) [41] and fast software implementations are available for 8-bit [44] and 16-bit [21] microcontrollers.

However, there are limits as to how far these types of adaptations can be pushed. They tend to fall short of what is required for today's most constrained environments, and surely won't meet tomorrow's needs. For example, the consensus has long been that a budget of 2000 GE is all the chip area that might reasonably be allocated for security on the most constrained radio-frequency identification (RFID) tags [36], and this is well out of reach for AES implementations. On microcontrollers, AES implementations can be very fast but they also tend to be large and complex. Implementations that decrease size or complexity certainly exist, but small implementations tend to be complex (and slow), while simple implementations tend to be large (and slow).

One further point about AES: Not every application requires the same high level of security that AES is designed to provide. When resources are scarce, it doesn't always make sense to lavish them on an algorithm providing 128 (or 192 or 256) bits of security when 96 might suffice. In addition, the AES block size of 128 bits is not always optimal. An RFID authentication protocol may only ask that 64-bit quantities be encrypted, and demanding 128 bits of state when only 64 are necessary can amount to a significant waste of chip area.

These are the principal reasons for the development of new lightweight block ciphers, and many new algorithms have been proposed. Since the limitations of AES are more apparent in hardware than in software, most of the best efforts to date have focused on this aspect of the problem. This work has produced designs including PRESENT [17], KATAN [22], and

Piccolo [52], each of which has a very small hardware footprint. But none was meant to provide high performance on constrained software-based devices, e.g., 8- and 16-bit microcontrollers. The designers of LED [35] and TWINE [57] are more intent on supporting software implementations, but these algorithms retain a bias toward hardware performance.

We believe a lightweight block cipher should be "light" on a wide range of hardware- and software-based devices, including ASICs, field-programmable gate arrays (FPGAs), and 4-, 8-, 16-, and 32-bit microcontrollers. Moreover, as noted in [11], many of these devices will interact with a backend server, so a lightweight block cipher should also perform well on 64-bit processors.

It seems clear to us that there is a need for *flexible* secure block ciphers, i.e., ones which can perform well on *all* of these platforms. Our aim, with the design of Simon and Speck, is to make this sort of block cipher available for future use.

The Simon and Speck block ciphers

In 2011, prompted by potential US government requirements for lightweight ciphers [e.g., supervisory control and data acquisition (SCADA) and logistics applications] and the concerns with existing cryptographic solutions which we've noted above, we began work on the Simon and Speck block cipher families on behalf of the Research Directorate of the US National Security Agency (NSA).

Because our customers will rely on commercial devices, we determined that the only realistic way to make the algorithms available would be to put them in the public domain. Furthermore, because cost will be such an important driver in this area—a fraction of a penny per device may make the difference between whether a cryptographic solution is viable or not—we were motivated to make Simon and Speck as simple, flexible, and lightweight as we could. Our hope was that their availability would make it possible to raise the security bar for future IoT devices.

The development process culminated in the publication of the algorithm specifics in June 2013 [9]. Prior to this, Simon and Speck were analyzed by NSA

cryptanalysts and found to have security commensurate with their key lengths; i.e., no weaknesses were found. Perhaps more importantly, the algorithms have been pretty heavily scrutinized by the international cryptographic community for the last two years (see, e.g., [1, 2, 3, 4, 5, 6, 7, 15, 16, 20, 24, 25, 27, 29, 30, 37, 42, 47, 51, 53, 56, 59, 60, 62]).

Table 1 summarizes the cryptanalytic results as of this writing that attack the most rounds of Simon and Speck. (We note that the recent paper [7] purports to attack 24 rounds of Simon 32/64. The author informs us that this paper is currently under revision, and we have therefore not included those results in table 1. For more, see the comments regarding this work in [24].) The content of the table is simple: There are no attacks on any member of the Simon or Speck families, and each block cipher maintains a healthy security margin.

TABLE 1. Security of Simon and Speck.

Size	Alg	Rounds		Ref
		Total	Attacked	
32/64	Simon	32	23 (72%)	[24]
	Speck	22	14 (64%)	[29, 66]
48/72	Simon	36	24 (67%)	[24]
	Speck	22	15 (68%)	[66]
48/96	Simon	36	25 (69%)	[24]
	Speck	23	16 (70%)	[66]
64/96	Simon	42	30 (71%)	[24]
	Speck	26	19 (73%)	[66]
64/128	Simon	44	31 (70%)	[24]
	Speck	27	20 (74%)	[66]
96/96	Simon	52	37 (71%)	[24, 61]
	Speck	28	20 (71%)	[66]
96/144	Simon	54	38 (70%)	[24]
	Speck	29	21 (72%)	[66]
128/128	Simon	68	49 (72%)	[24, 61]
	Speck	32	22 (69%)	[66]
128/192	Simon	69	51 (74%)	[24]
	Speck	33	23 (70%)	[66]
128/256	Simon	72	53 (74%)	[24]
	Speck	34	24 (71%)	[66]

As we see in the table, Simon and Speck are not simply block ciphers, but are block cipher *families*, each family comprising 10 distinct block ciphers with differing block and key sizes to closely fit application requirements.

We will write Simon $2n/mn$ to mean the Simon block cipher with a $2n$ -bit block and m -word (mn -bit) key. We will sometimes suppress mention of the key and just write Simon 128, for example, to refer to a version of Simon with a 128-bit block. The analogous notation is used for Speck.

The block and key sizes we support are shown in table 2. The range here goes from tiny to large: a 32-bit block with a 64-bit key at the low end, to a 128-bit block with a 256-bit key at the high end.

We note that key lengths below 80 bits or so do not provide an especially high level of security, but they may still be useful for certain highly constrained applications where nothing better is possible.

TABLE 2. Simon and Speck parameters.

Block Size	Key Sizes
32	64
48	72, 96
64	96, 128
96	96, 144
128	128, 192, 256

The desire for flexibility through simplicity motivated us to limit the operations used within Simon and Speck to the following shortlist:

- modular addition and subtraction, $+$ and $-$,
- bitwise XOR, \oplus ,
- bitwise AND, $\&$,
- left circular shift, S^j , by j bits, and
- right circular shift, S^{-j} , by j bits.

Speck gets its nonlinearity from the modular addition operation, which slightly favors software performance over hardware. Simon's nonlinear function is a bitwise AND operation, which tends to favor hardware over software. But modular addition can be computed efficiently in hardware, and similarly, bitwise AND is easy and natural in software.

The round functions for Simon $2n$ and Speck $2n$ each take as input an n -bit *round key* k , together with two n -bit *intermediate ciphertext* words. For Simon, the round function is the 2-stage Feistel map

$$R_k(x, y) = (y \oplus f(x) \oplus k, x),$$

where $f(x) = (Sx \& S^8x) \oplus S^2x$ and k is the round key. For Speck, the round function is the (Feistel-based) map

$$R_k(x, y) = ((S^{-\alpha}x + y) \oplus k, S^\beta y \oplus (S^{-\alpha}x + y) \oplus k),$$

with rotation amounts $\alpha = 7$ and $\beta = 2$ if $n = 16$ (block size = 32) and $\alpha = 8$ and $\beta = 3$ otherwise.

The round functions are composed some number of times which depends on the block and key size. See table 1.

Each algorithm also requires a *key schedule* to turn a key into a sequence of round keys. We briefly describe the key schedules, but refer the reader to [9] for complete details.

For Simon, if we let the key value be k_0, \dots, k_{m-1} ($m \in \{2, 3, 4\}$ is the number of key words), the sequence of round keys is k_0, k_1, k_2, \dots , where

$$\begin{aligned} k_{i+2} &= k_i \oplus (I \oplus S^{-1})S^{-3}k_{i+1} \oplus C_i, \\ k_{i+3} &= k_i \oplus (I \oplus S^{-1})S^{-3}k_{i+2} \oplus D_i, \\ k_{i+4} &= k_i \oplus (I \oplus S^{-1})(S^{-3}k_{i+3} \oplus k_{i+1}) \oplus E_i, \end{aligned}$$

depending on whether m is 2, 3, or 4, respectively. The values C_i , D_i , and E_i are round constants which serve to eliminate slide properties; we omit discussion of them here. I is the $n \times n$ identity matrix.

Like Simon, Speck has 2-, 3-, and 4-word key schedules. Speck's key schedules are based on its round function, as follows. We let m be the number of words of key, and we write the key as $(l_{m-2}, \dots, l_0, k_0)$. We then generate two sequences k_i and l_i by

$$\begin{aligned} l_{i+m-1} &= (k_i + S^{-\alpha}l_i) \oplus i \text{ and} \\ k_{i+1} &= S^\beta k_i \oplus l_{i+m-1}. \end{aligned}$$

The value k_i is the i th round key, for $i \geq 0$. Note the round counter i here which serves to eliminate slide properties.

Design notes

Efficiency and security are competing goals in cryptographic design, and understanding how to strike

the right balance is the primary challenge faced by a designer. If security is not important, efficiency is easy: do nothing! Conversely, if efficiency doesn't matter, then it makes sense to build a round function using the most secure cryptographic components available, and then iterate an absurdly large number of times. But in the real world both of these things matter, and we'd like to design algorithms that are maximally efficient, while still providing the advertised level of security, as determined by the key size.

There is an important intellectual challenge associated with understanding optimally secure cryptographic components such as 8-bit S-boxes. However, we would argue that the way to design efficient cryptography, particularly cryptography for constrained platforms, is to forgo them in favor of very simple components, iterating an appropriate number of times to obtain a secure algorithm. Such simple components are by their nature cryptographically weak, making them unappealing to some designers. But simplicity enables compact implementations, and deciding on appropriate numbers of rounds is possible with analysis.

The question is whether there is something inherently wrong with this approach. It seems clear to us that there isn't: After all, a complex round function can always be factored into a composition of simple functions (transpositions, even), and so *every* block cipher is a composition of simple functions. It's just that in general the decomposition into simple functions is not useful to an implementer, because the factors tend to be unrelated, and so there is no associated efficient implementation of the algorithm. Viewed this way, we could imagine that Simon and Speck are based on complex round functions—a "round" in this sense may in fact mean eight of the usual rounds—but we've worked to make those complex round functions factor into identical functions, at least up to the translations by round key.

We now discuss in a bit more detail the thinking that went into the design of Simon and Speck.

Nonlinear and linear components

Most designers of lightweight block ciphers employ S-boxes to provide nonlinearity; a notable feature of Simon and Speck is their lack of dependence on S-boxes. The appeal of S-boxes is that, when used as a part of a substitution-permutation network (SPN),

they allow for relatively easy security arguments, at least with respect to standard attacks. But for efficiency on constrained platforms, we believe that these sorts of designs are not optimal. We prefer to increase the one-time work necessary to do the cryptanalysis, in order to reduce the every-time work of encryption and decryption.

Lightweight block ciphers often use bit permutations as part of an SPN. The role of these bit permutations is to spread bits around in some optimal manner, and therefore allow SPN-style security arguments. If the target platform is an ASIC this is a perfectly reasonable thing to do, as such permutations are essentially free. But if we care about software implementations at all, then extreme care must be taken to ensure that the bit permutation can be done efficiently on a microprocessor. The bit permutations we use are all circular shifts, which are easy to effect on just about any platform. While we lose something in diffusion rates as compared with more general bit permutations, we are able to achieve significant improvements in software performance, even when increased round numbers are factored in.

One might argue that arbitrary bit permutations are fine in software, because efficient *bit-sliced* implementations are possible. However, it doesn't seem wise to rely on these, as they have drawbacks—including relatively expensive data transpose operations on the plaintext and ciphertext, and the inability to efficiently encrypt single plaintext blocks (and single encryptions will be necessary for many lightweight communication and authentication protocols). In addition, the code size and the random-access memory (RAM) requirements tend to be quite large, making such implementations unsuitable for some lightweight applications.

Parameters

Both Simon and Speck are equipped with a single set of rotation parameters for all variants (with the exception of the smallest version of Speck, which has its own set of parameters). Besides allowing a succinct description of the family, this *uniformity* helps reduce the risk of coding errors whereby a programmer might mistakenly use the Simon 64/128 parameters, say, for Simon 128/128.

Many microcontrollers only support shifts by a single bit; the result is that a rotation by two bits is twice

as expensive as a rotation by one bit. On the other hand, 8-bit rotations tend to be easy on 8-bit microcontrollers, as they correspond to simple relabelings of registers, and well supported through byte-swap or byte-shuffle operations on machines with larger word sizes. So for efficiency on a variety of software platforms, it's best to keep rotation amounts as close to multiples of eight as possible.

The Simon and Speck rotation amounts were carefully chosen with this consideration in mind. Both algorithms employ 8-bit rotations, and the other rotations used are as close to multiples of eight as we could make them, without sacrificing security.

In-place operations in software

Speck's superior performance in software is due in part to the fact that it's possible to implement it entirely with in-place operations, and so moves are unnecessary. This can be seen in the following pseudocode for a round of Speck:

$$x = \text{RCS}(x, \alpha)$$

$$x = x + y$$

$$x = x \oplus k$$

$$y = \text{LCS}(y, \beta)$$

$$y = y \oplus x$$

Simon requires some moves, because multiple operations are done on a single word of intermediate ciphertext, and copies need to be made. This fact (combined with the fact that Simon uses a weaker nonlinear function than Speck, and so more rounds are required) makes Speck outperform Simon in software.

Encrypt/decrypt symmetry

To enable compact joint implementations of the encryption and decryption algorithms, it's best to make encryption look like decryption. Simon decryption can be accomplished by swapping ciphertext words, reading round keys in reverse order, and then swapping the resulting plaintext words.

We note that Simon beats Speck in this regard (Speck decryption requires modular subtraction, and the rotations are reversed), *because* its Feistel stepping

performs all operations on one word, which is precisely why its software implementations required moves.

Key schedule considerations

Speck's reuse of the round function for key scheduling allows for reductions in code size and improves performance for software implementations requiring on-the-fly round key generation.

Because Simon was optimized for hardware, it does not take advantage of this software-oriented optimization. Instead, it uses a key schedule which was designed to be a little lighter than the round function.

Of course it is possible to have key schedules even simpler than the ones we have used for Simon and Speck; for example, one can produce round keys simply by cycling through key words. This leads to the possibility of "hard-wiring" the key in an ASIC implementation, thereby saving considerably on area by eliminating any flip-flops needed for holding the key. But such an approach, when used together with very simple round functions, can lead to related-key issues, and we therefore avoided it.

We believe the ability to use hard-wired key is of limited utility, and it runs counter to our flexibility goal by optimizing for a particular sort of use, perhaps to the detriment of other uses in the form of increased numbers of rounds or cryptanalytic weaknesses. Our key schedules do the minimal mixing that we thought would eliminate the threat of related-key attacks.

Both block ciphers include round constants, which serve to eliminate slide issues. Speck, where design choices were made to favor software over hardware, uses one-up counters. Simon achieves a small savings in hardware (at a small cost in software) by using a sequence of 1-bit constants generated by a 5-bit linear register.

As a final point, we omit plaintext and ciphertext key whitening operations, as such operations would increase circuit and code sizes. This means that the first and last rounds of the algorithms do nothing cryptographically, beyond introducing the first and last round keys.

We conclude this section by pointing to some work that we think helps to validate our approach to

the design of Simon and Speck. Designing an algorithm to perform well on a particular platform is a straightforward proposition; we believe the real test is performance on *unintended* platforms, in particular platforms which may not even exist today.

As we've noted, it's hard to get a handle on an issue like this, but we have one data point that's interesting: Because of its simplicity (more precisely, its low multiplicative depth), Simon has been picked up by more than one team [23, 38] for use in the decidedly non-lightweight world of homomorphic encryption.

Implementations on constrained platforms

In this section, we quickly summarize implementation results for Simon and Speck on constrained platforms, beginning with ASICs and FPGAs, and then moving on to microcontrollers.

ASICs

Until recently, designers of lightweight cryptography primarily took aim at ASIC performance. As a result, there are a number of excellent ASIC designs (see table 3), all of which can be implemented with substantially less area than the 2400 GE required by AES. Much of this improvement is possible because of the hardware complexity of AES components, in particular its S-box. But a significant gain comes from the recognition that a 128-bit block size is not always required for constrained applications, and there is a considerable area savings to be had by reducing to a 64-bit block.

As we've noted, care must be taken with an ASIC design, or else software performance can suffer. Software performance is indeed a weakness of a number of existing algorithms. Simon and Speck have improved on the state of the art for hardware implementation, while also offering leading software performance.

Simon has ASIC implementations with the smallest areas achieved to date, when compared with block ciphers with the same block and key size and with flexible key. This is because the logic required for a bit-serial implementation (meaning that only one bit of the round function is computed per clock cycle)

is minimal: Computing a bit of the round function requires just one AND and three XORs, and so there isn't much room for further improvement. There is of course additional logic required for control (which we've also worked to minimize), and a few XORs are needed in the key schedule, etc., but for the smallest implementations, almost all the area is used by the flip-flops required to store the state.

Because the logic required to compute a bit of the round function is so small, implementations of Simon scale nicely: Two bits or more can be updated in one clock cycle with minimal impact on area.

Speck is not far behind Simon with respect to small ASIC implementations. The primary differences are that Simon's AND gets replaced with a full adder, and some additional multiplexing is required because of how the state updates. Its area also scales well, but not quite as well as Simon's.

In the remainder of this section, we provide area and throughput data to illustrate the ASIC performance of Simon and Speck.

Our ASIC implementations were done in Very high-speed integrated circuit Hardware Description Language (VHDL) and synthesized using Synopsys Design Compiler 11.09-SP4 to target the ARM SAGE-X v2.0 standard cell library for IBM's 8RF 130 nanometers (nm) (CMR8SF-LPVT) process.

Worst-case operating conditions were assumed. We did not proceed to place and route: In an actual chip there will be interconnect delays that haven't been accounted for, and these delays will likely significantly affect clock speeds. But we note that most work in this field—in particular the work cited in this paper—uses this approach, similarly ignoring interconnect delays, so this shouldn't bias our comparisons.

The smallest flip-flop available to us had an area of 4.25 GE. For a block cipher with a 64-bit block and 128-bit key, this means at least $4.25 \cdot 192 = 816$ GE are required for flip-flops. Our bit-serial implementations of Simon 64/128 and Speck 64/128 have areas of 958 GE and 996 GE, respectively. This means that they require (at most) $958 - 816 = 142$ GE and $996 - 816 = 180$ GE, respectively, for all the logic required to compute the round function, key schedule, and do the control, which includes loading the plaintext and

TABLE 3. ASIC performance comparisons at a 100 kHz clock speed optimized for size.

Size	Algorithm	Area (GE)	Tput* (kbps)	Ref
48/96	Simon	739	5.0	[9]
	Speck	794	4.0	[9]
64/80	TWINE	1011	16.2	[57]
	PRESENT	1030	12.4	[65]
	Piccolo	1043	14.8	[52]
	Katan	1054	25.1	[22]
	KLEIN	1478	23.6	[33]
64/96	Simon	809	4.4	[9]
	Speck	860	3.6	[9]
	KLEIN	1528	19.1	[33]
64/128	Simon	958	4.2	[9]
	Speck	996	3.6	[9]
	Piccolo	1334	12.1	[52]
	PRESENT	1339	12.1	[65]
96/96	Simon	955	3.7	[9]
	Speck	1012	3.4	[9]
128/128	Simon	1234	2.9	[9]
	Speck	1280	3.0	[9]
	AES	2400	56.6	[41]

*Tput = Throughput

reading out ciphertext. And of the 142 GE not devoted to storing the cipher and key for Simon 64/128, $11 \cdot 4.25 = 46.75$ GE, or about a third, are flip-flops needed to count rounds in order to signal the end of encryption.

Table 3 compares size-optimized ASIC implementations of Simon, Speck, and some other prominent block ciphers, listing the area and throughput at a fixed 100 kHz clock rate. Note that we show our absolute smallest implementations of Simon and Speck, with correspondingly low throughputs. Throughputs can be doubled, quadrupled, etc., for small area increases. See [9] for data regarding additional implementations. For example, quadrupling the throughput for Simon 128/128 and Speck 128/128 increases the area by just 29 GE and 116 GE, respectively.

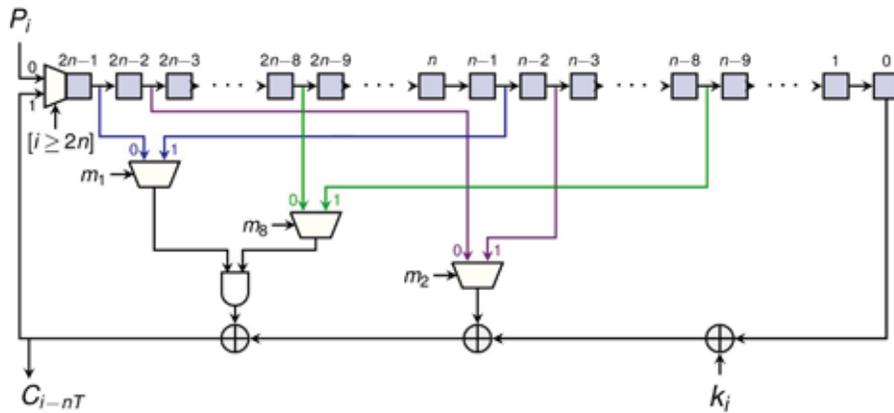


FIGURE 1. Simon round function serialization, one bit at a time. The clock steps from $i = 0$ to $(T + 2)n$. P_i denotes the i^{th} bit of plaintext, loaded at time i . k_i represents the round key bit required at time i . The control bits m_k into the MUXes are given by counter-dependent values $m_k = m_k(i) = \lfloor (i \bmod n) \geq k \rfloor$ (where $\lfloor a \rfloor = 1$ if a , else 0). Ciphertext bits are output during the final $2n$ cycles of encryption.

An important caveat is that these comparisons consider implementations done by different authors, with perhaps different levels of effort, and using different cell libraries, so it's hard to make really meaningful inferences regarding small differences in the table.

Large differences, on the other hand, are meaningful, and comparing Simon and Speck with AES shows the dramatic savings possible with a lightweight block cipher. At the same security level, Simon and Speck nearly halve AES's 2400 GE area to 1234 and 1280 GE, respectively. Keeping the same 128-bit key size and reducing the block size to 64 bits further drops the areas to 958 and 996 GE. Using smaller block or key sizes results in even greater area reductions.

Some applications won't require areas to be minimized; rather it may be important to maximize *efficiency* [throughput divided by area, in kilobits per second per GE (kpbs/GE)]. The implementations in table 3 have low efficiency, but efficiency can easily be raised by doing additional computation during each clock cycle, in effect to begin to amortize away the fixed cost of storing the state. The flexibility of Simon and Speck mean that many sorts of implementations are possible. See the section in this article on implementations on higher-end platforms for data regarding efficient implementations; in particular implementations which compute a full round per clock cycle, and implementations which fully unroll the algorithms.

We conclude this section by discussing *latency*, i.e., the time required to encrypt *one* plaintext block. Low-latency implementations of block ciphers have recently been much discussed; the leading voices have been the authors of [19]. The algorithm they propose, PRINCE, is a clever design which can encrypt in one clock cycle at the impressively small area of 8679 GE [19]. (We note that registers were not counted in this total, and a real system would probably need to register

the data, thus increasing the area by about 10% to around 9500 GE.) The recent paper [39] increases the area to 9522 GE (about 10500 GE counting registers), but achieves a record latency of 22.9 nanoseconds (ns).

It would appear that Simon and Speck are not low-latency designs, because they require many rounds. However, because of their simplicity, it's possible to compute multiple rounds per clock cycle, while maintaining reasonably good clock speeds. Indeed for Simon 64/128, we've found an implementation (at the same 130 nm feature size used in [39]) that almost exactly matches PRINCE's latency and area; it implements the combinational logic for five rounds, and encrypts in $\lfloor 44/5 \rfloor = 9$ cycles. In spite of its need to compute carry chains, Speck can get within a factor of 2.5 of PRINCE's latency, at a much smaller area. (Three rounds are computed per clock cycle, for a total of $(27/3) + 1 = 10$ cycles—our current Speck implementation requires a load cycle, which it should be possible to eliminate with a little more work.) Of course these are not single-cycle implementations, but we don't see a compelling case that such implementations are necessary, particularly at what seem to be artificially constrained clock speeds, and on the sort of devices considered in [39] where clocks are easy to generate. See table 4, where one Speck and two Simon implementations are shown; many other latency/area trade-offs are possible but are omitted here.

TABLE 4. Low-latency encrypt-only implementations of PRINCE, Simon, and Speck at 130 nm. The Simon and Speck implementations count 64+128 flip-flops; the PRINCE implementation doesn't.

Algorithm	Area (GE)	Latency (ns)	Clock (MHz)
PRINCE	9522	22.9	43.7
Simon 64/128	9516	22.88	437.1
	5072	31.90	344.9
Speck 64/128	6377	52.36	191.0

FPGAs

We've shown that it's possible to realize considerable reductions in ASIC area by using Simon or Speck instead of an algorithm such as AES. The advantages of Simon and Speck become even more pronounced on FPGA platforms.

In this section, we briefly discuss implementations of the algorithms on the Spartan-3, a low-end FPGA which is often used by cryptographers for comparisons. Table 5 presents some of these results for AES and PRESENT, alongside results for our algorithms.

On this platform, the smallest reported implementation of AES-128 requires 184 slices [26]. Remarkably, Simon 128/128 can be implemented in just 28 slices (15% of the size of AES), and Speck 128/128 can be done in 36 slices (20% of AES's size). Comparisons with PRESENT also show dramatic area reductions: PRESENT-128 requires 117 slices; the comparable Simon 64/128 and Speck 64/128 algorithms require 24 and 34 slices—21% and 30% of the area—respectively.

If higher throughputs are required, area reductions are still possible, as can be seen in table 5.

Other authors have reported Simon implementation results [8, 13, 34, 49] which are in line with our results, and extend them. In [34], it is shown that a joint implementation of all 10 versions of Simon can be done using 90 slices on the Spartan-3, which is about half the size of a single AES-128 implementation. The 87-slice implementation of Simon 128/128 described in [49] provides resistance to first-order differential power analysis, again at about half the area of an *unprotected* AES-128 implementation.

TABLE 5. FPGA performance comparisons on low-cost Xilinx Spartan FPGAs. All implementations are on the Spartan-3. Results marked with a † are our work. The Simon implementation labeled (DPA) is resistant to first-order DPA.

Size	Algorithm	Area (slices)	Tput (Mbit/s)	Ref
64/128	Simon	24	9.6	†
	Simon	138	512	†
	Speck	34	7.0	†
	Speck	153	416	†
	PRESENT	117	28.4	[64]
	PRESENT	202	508	[46]
128/128	Simon	28	5.7	†
	Simon	36	3.6	[8]
	Simon (DPA)	87	3.0	[49]
	Simon	197	567	†
	Simon	375	867	†
	Speck	36	5.0	†
	Speck	232	455	†
	Speck	401	920	†
	AES	184	36.5	[26]

Microcontrollers

We turn now to software implementations on 8-bit, 16-bit, and low-end 32-bit microcontrollers. Table 6 shows read-only memory (ROM) and RAM usage and encryption cost (in cycles/byte) for assembly implementations of Simon, Speck, and a few other algorithms [43, 44]. The first half of the table shows implementations optimized for *efficiency*¹ and the second half implementations optimized for speed.

The data for PRESENT exemplifies the potential difficulty of adapting hardware-oriented algorithms to software; this algorithm is unable to match the performance of AES, and is easily beaten by Simon and Speck in both throughput and code size.²

For high-speed applications on the 8-bit AVR microcontroller, AES-128 is the fastest 128-bit block cipher we know of, beating Speck 128/128 by about 17%. However, because of its low memory usage, Speck 128/128 has higher efficiency than AES-128. And as key sizes increase, Speck overtakes AES in

1. We define efficiency to be encryption throughput in bytes per cycle, divided by ROM+2 · RAM. See [10].

2. We note that there is a faster bit-sliced implementation of PRESENT [45], which encrypts at 370.875 cycles per byte, plus about 40 cycles per byte for data transposition operations. But it's much larger, requiring 3816 bytes of ROM and 256 bytes of RAM.

TABLE 6. Assembly implementations on the 8-bit AVR ATmega128 and 16-bit MSP430 microcontrollers.

Size	Algorithm	AVR			MSP430		
		ROM (bytes)	RAM (bytes)	Cost (cyc/byte)	ROM (bytes)	RAM (bytes)	Cost (cyc/byte)
Efficient Implementations							
64/80	PRESENT [31]	936	0	1340	-	-	-
64/128	Speck	218	0	154	204	0	98
	Simon	290	0	253	280	0	177
	TWINE [40]	1208	23	326	-	-	-
128/128	Speck	460	0	171	438	0	105
	AES-128 [10]	970	18	146	-	-	-
	Simon	760	0	379	754	0	389
Fast Implementations							
64/128	Speck	628	108	122	556	0	89
	Simon	436	176	221	324	0	153
128/128	AES-128 [21, 43]	1912	432	125	3147	176	132
	Speck	452	256	143	602	0	101
	Simon	510	544	337	1108	0	379

throughput because of how round numbers scale. Moreover, Speck 64/128, which has the same key size as AES-128, but a smaller block, is both smaller and slightly faster than AES-128.

On the 16-bit MSP430, Speck is the highest inefficiency and throughput. It is 23% faster than AES, uses no RAM and 81% less ROM. In [21] this performance advantage resulted in a 35% lower energy consumption compared to AES. Speck 64/128 consumes even fewer resources for the many applications where a smaller block size is acceptable.

Others' work supports our conclusions. In [28], C implementations of AES, Simon 64/96, Speck 64/96, and 10 other lightweight algorithms are compared on the 8-bit AVR, 16-bit MSP430, and 32-bit ARM Cortex-M3 microcontrollers. Algorithms were ranked in two usage scenarios using a *figure of merit* balancing performance, RAM, and code size across the three platforms. Speck and Simon place first and fourth in a large data scenario and first and second in a scenario involving encryption of a single block.

On the 32-bit ARM processor, the authors of this paper find Speck and Simon to be simultaneously the smallest and fastest block ciphers for both of the

scenarios they consider. We point out, however, that their C implementations of AES are faster than those of Speck on the 8-bit and 16-bit platforms by about a factor of two, presumably due to the GNU C compiler's poor handling of rotations. Implementing the rotations in assembly should lead to greatly improved performance for our rotation-dependent designs.

It is our opinion that for lightweight applications on microcontrollers, if high performance is important, then Simon and Speck should be coded in assembly: Because of the simplicity of the algorithms, these implementations are pretty straightforward, and they can improve performance by up to a factor of five over C implementations. Details on such implementations on the AVR microcontroller can be found in [10].

Implementations on higher-end platforms

Constrained devices will need to communicate with other, similar devices, but will also need to communicate with higher-end systems. These systems may perform functions such as aggregating sensor or inventory data. To facilitate these sorts of interactions

TABLE 7. Efficient, high-throughput 130 nm ASIC implementations of Simon and Speck.

Size	Algorithm	Area (GE)	Throughput (Mbps)	Efficiency (kpbs/GE)	Clock (MHz)	Implementation
64/128	Simon	1751	870	497	625	iterative
		44322	34243	773	535	key-agile pipeline
		35948	45070	1254	704	non-key-agile pipeline
	Speck	2014	634	315	307	iterative
		48056	23908	498	374	key-agile pipeline
		39992	29722	743	464	non-key-agile pipeline
128/128	Simon	2342	1145	489	626	iterative
		146287	106961	731	836	key-agile pipeline
		104790	87798	838	686	non-key-agile pipeline
	Speck	3290	880	268	234	iterative
		98003	41531	424	324	key-agile pipeline
		86976	52162	600	408	non-key-agile pipeline
128/256	Simon	3419	1081	316	625	iterative
		233204	100078	429	782	key-agile pipeline
		110875	87193	786	681	non-key-agile pipeline
	Speck	5159	1287	249	382	iterative
		163770	51705	316	404	key-agile pipeline
		97432	52056	534	407	non-key-agile pipeline

and in particular to support efficient communication with large numbers of constrained devices, lightweight algorithms will need to perform well on both light-weight and “heavyweight” platforms.

High-throughput ASIC implementations

Table 7 shows a sample of higher-throughput implementations on the same 130 nm ASIC process used to generate the Simon and Speck data in table 3. Decryption is not supported in these implementations, but for Simon, in particular, it could be added at low cost due to the similarity of the encryption and decryption algorithms.

For each algorithm and block/key size, we present an iterative and two fully pipelined encryption implementations. In the iterative case, a single copy of the round function is used to loop over the data for a number of cycles equal to the total number of rounds.

In the fully pipelined case, a number of copies of the round function equal to the number of rounds is

implemented, with registers in between. This allows a complete block of ciphertext to be output every clock cycle, once the pipeline is full. One of the fully pipelined implementations is key-agile, meaning that every plaintext block to be encrypted can have its own associated key. The second fully pipelined implementation is not key-agile: It saves area by requiring that all blocks in the pipeline use the same key, so that only one instance of the key schedule is necessary, rather than one for each level of the pipeline. Changing key for this second sort of implementation requires the new round keys to be loaded and the pipeline to be flushed.

The flexibility of Simon and Speck enables all sorts of implementations in between these performance extremes (e.g., iterated versions computing multiple rounds per clock cycle, and pipelined implementations with multiple rounds between stages), but we do not have the space to include those results here.

Simon and Speck have compelling advantages for high-throughput ASIC applications. This seems clear,

even in view of the difficulties inherent in comparing implementations using different technologies and libraries. As a point of comparison, we consider the CLEFIA block cipher.³ The designers of that algorithm report on a joint implementation [55] of the encryption and decryption algorithms⁴ which has an efficiency of 401, using a 90 nm technology [9339 GE, 3.74 gigabits per second (Gbit/s) at 572 megahertz (MHz)]. This is excellent performance relative to other block ciphers; indeed CLEFIA realizes the “world’s highest hardware gate efficiency” [54].

We did ASIC implementations of Simon and Speck at this same 90 nm feature size. (Note that these results are not reported in table 7, where the feature size is 130 nm.) Speck has a 8089 GE (encrypt-only) implementation, running at 1.404 gigahertz (GHz), for a throughput of 10.6 Gbit/s and an efficiency of 1307. Simon is even better: For 8011 GE, an encrypt-only version runs at 3.066 GHz, for a throughput of 17.1 Gbit/s and an efficiency of 2130. There may be differences in cell libraries, etc. (and we note again that interconnect delays are not considered in our work or in the CLEFIA work), but a factor of 2130/401 > 5 improvement is surely significant.

x86 and ARM implementations

We have recently studied implementations of Simon and Speck as stream ciphers in counter mode on

several higher-end 32-bit and 64-bit processors. These processors are likely to be used in systems such as smartphones, tablets, and servers communicating with constrained devices. We considered the 32-bit Samsung Exynos 5 Dual (which includes NEON SIMD instructions), based on an ARM Cortex-A15, and two 64-bit Intel processors: the Xeon E5640 and Core i7-4770, representing the Westmere and Haswell architectures, respectively. Performance was benchmarked using SUPERCOP [12], making for fair comparison with the performance of highly optimized implementations of AES and ChaCha20, in particular. The Simon and Speck code, all written in C, is available on GitHub [63]. Figure 2 illustrates the detailed data produced by SUPERCOP.

The overall results are similar on the ARM and the x86 platforms. The C implementations of Simon have better overall performance than the C implementations of AES for 256-bit keys and slightly worse performance for 128-bit keys. The C implementations of Speck 128/256 have better overall performance than the best C implementations of ChaCha20, a stream cipher especially noted for its speed.

Finally, we note that extremely high-performance instantiations of AES are possible on certain processors, for example using Intel’s hardware AES-NI instructions. Despite this, Speck in software can come close to matching this high performance: On the Haswell

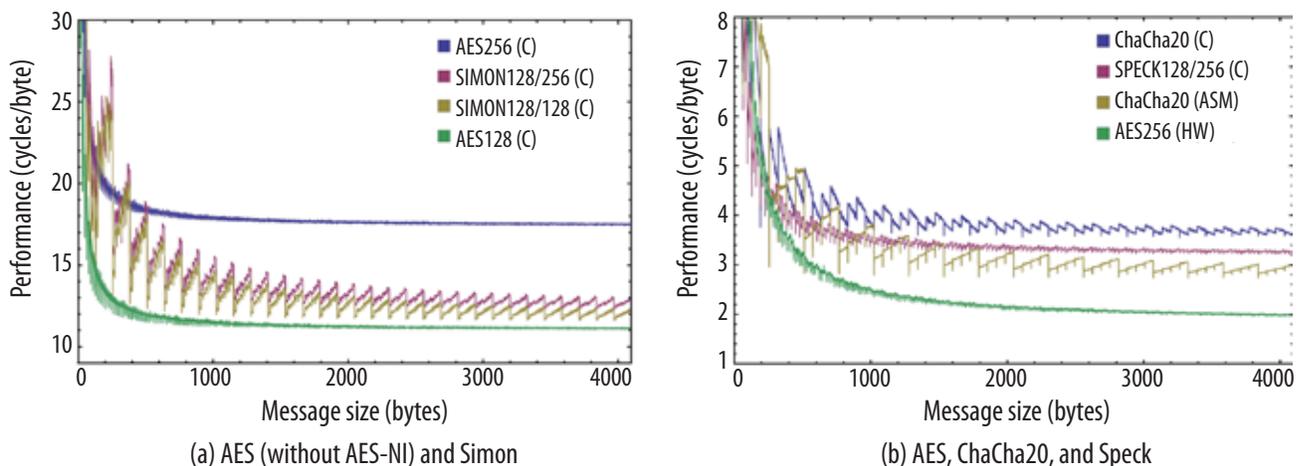


FIGURE 2. Intel Xeon E5640 throughput in cycles/byte (smaller is better) for messages from 1–4096 bytes.

3. CLEFIA is a lightweight ISO standard which supports high-throughput ASIC implementations.

4. CLEFIA’s symmetry means that there is little overhead in providing decryption functionality. On the other hand, the area won’t go down by much for an encrypt-only version.

architecture our C implementation of Speck 128/256 is only 33% slower than the AES-NI version of AES 256.

Side-channel mitigations

The most secure algorithm can become vulnerable to attack if it is implemented in a way that leaks information because power usage or execution time (or something else) is correlated to secret key values. Understanding these sorts of *side-channels* and how to eliminate them is an important line of research, and it's particularly relevant for constrained devices, which tend to lack physical countermeasures.

We very briefly discuss side-channel attacks and mitigations, and note some work in this area involving Simon and Speck.

One sort of side-channel attack exploits key-dependent variations in encryption times to recover secret information. Algorithms which are implemented using look-up tables, e.g., AES, on processors with cache memory can be particularly vulnerable to these *cache-timing attacks* [18]. Since Simon and Speck have no look-up tables, they are naturally immune to this type of attack.

Perhaps the most important type of side-channel attack uses key-dependent power emanations. Implementations of block ciphers typically are susceptible to such *differential power analysis (DPA)* attacks unless countermeasures are taken. Because of Simon's low-degree round function, *masking* countermeasures are especially efficient; see [49, 50]. In particular, the second of these papers demonstrates a threshold implementation of Simon 128/128 which provides resistance to first-order DPA for 87 slices on a Spartan-3 FPGA. This makes it less than half the size of the smallest reported unprotected Spartan-3 implementation of AES, and 25% smaller than unprotected implementations of PRESENT-128. (And PRESENT-128 is not exactly a comparable algorithm, since it has a block size of 64 bits, and the version of Simon they consider has a block size of 128 bits.)

We are not aware of similar work to protect Speck, but there are other countermeasures that apply equally to both Simon and Speck. One such measure aims to confound DPA by partially unrolling an algorithm [14]. We've done such implementations of Simon and

Speck, but don't have the space in this paper to discuss them. Briefly, for the 64-bit block and 128-bit key size, there is an ASIC implementation of Simon that computes four full rounds per clock cycle and requires 3290 GE. A similar implementation of Speck computes three rounds per clock cycle and has an area of 3120 GE. We have not done side-channel analysis for these implementations.

Another mitigation uses frequent key updating [58]. The tiny hardware implementations of Simon and Speck in tables 3 and 5 are key agile, meaning the key can be changed with each run without incurring a significant performance penalty, and so they would be good candidates for use with this strategy.

Conclusion

We have sought in this paper to demonstrate the sort of performance that Simon and Speck can achieve. Most importantly, Simon and Speck have an edge over other algorithms *not* in terms of head-to-head comparisons on particular platforms (although it appears that on most platforms one of Simon or Speck is the best existing algorithm, and the other is not far behind), but by virtue of their flexibility. This flexibility is a consequence of the simplicity of the designs, and means the algorithms admit small ASIC, FPGA, microcontroller, and microprocessor implementations, but can also achieve very high throughput on all of these platforms. Their flexibility makes Simon and Speck ideal for use with heterogeneous networks, where algorithms optimized for particular platforms or usages will not be appropriate.

The simplicity of Simon and Speck has additional benefits. First, they are very easy to implement, and efficient implementations can be had for minimal work; this is in marked contrast to the situation for algorithms such as AES, where a decade of research was required to find near-optimal implementations. Coding errors are much easier to avoid for simple algorithms. In addition, simplicity enables relatively cheap side-channel mitigations, and makes the algorithms attractive for unanticipated uses (such as homomorphic encryption). Last, but not least, simplicity makes the algorithms attractive targets for cryptanalysis. Complexity in this regard presents a barrier to entry, and this tends to limit the amount of scrutiny that

an algorithm receives. Because of their simplicity (and perhaps because of their source!), Simon and Speck have been quite thoroughly vetted by the cryptographic community in the two years since their publication.

Simon and Speck are also unique among existing lightweight block ciphers in their support for a broad range of block and key sizes, allowing the cryptography to be precisely tuned to a particular application.

We are hopeful that the approach we have taken to the design of Simon and Speck means they will continue to offer high performance on tomorrow's IoT devices. 

Bibliography

- [1] Abdelraheem MA, Alizadeh J, Alkhzaimi HA, Aref MR, Bagheri N, Gauravaram P, and Lauridsen MM. “Improved linear cryptanalysis of reduced-round SIMON.” *Cryptology ePrint Archive*, Report 2014/681, 2014. Available at: <http://eprint.iacr.org/2014/681.pdf>.
- [2] Abed F, List E, Lucks S, and Wenzel J. “Differential and linear cryptanalysis of reduced-round Simon.” *Cryptology ePrint Archive*, Report 2013/526, 2013. Available at: <http://eprint.iacr.org/2013/526.pdf>.
- [3] Abed F, List E, Lucks S, and Wenzel J. “Differential cryptanalysis of round-reduced Simon and Speck” In: Cid C, Rechberger C, editors. *Fast Software Encryption, FSE 2014*, 2014 Mar 3-5; London, UK: LNCS, p. 525-545. Volume 8540 of LNCS; Springer;2014. Available at: http://link.springer.com/chapter/10.1007/978-3-662-46706-0_27.
- [4] Alizadeh J, Alkhzaimi H, Aref MR, Bagheri N, Gauravaram P, Kumar A, Lauridsen MM, and Sanadhya SK. “Cryptanalysis of SIMON variants with connections.” In: Saxena and Sadeghi [48], p. 90–107.
- [5] Alizadeh J, Bagheri N, Gauravaram P, Kumar A, and Sanadhya SK. “Linear cryptanalysis of round reduced Simon.” *Cryptology ePrint Archive*, Report 2013/663, 2013. Available at: <http://eprint.iacr.org/2013/663.pdf>.
- [6] Alkhzaimi HA and Lauridsen MM. “Cryptanalysis of the SIMON family of block ciphers.” *Cryptology ePrint Archive*, Report 2013/543, 2013. Available at: <http://eprint.iacr.org/2013/543.pdf>.
- [7] Ashur T. “Improved linear trails for the block cipher Simon.” *Cryptology ePrint Archive*, Report 2015/285, 2015. Available at: <http://eprint.iacr.org/2015/285.pdf>.
- [8] Aysu A, Gulcan E, and Schaumont P. “SIMON says, break the area records for symmetric key block ciphers on FPGAs.” *Cryptology ePrint Archive*, Report 2014/237, 2014. Available at: <http://eprint.iacr.org/2014/237.pdf>.

[9] Beaulieu R, Shors D, Smith J, Treatman-Clark S, Weeks B, and Wingers L. “The SIMON and SPECK families of lightweight block ciphers.” *Cryptology ePrint Archive*, Report 2013/404, 2013. Available at: <http://eprint.iacr.org/2013/404.pdf>.

[10] Beaulieu R, Shors D, Smith J, Treatman-Clark S, Weeks B, and Wingers L. “The SIMON and SPECK block ciphers on AVR8-bit microcontrollers.” In: Eisenbarth T and Öztürk E [32].

[11] Benadjila R, Guo J, Lomné V, and Peyrin T. “Implementing lightweight block ciphers on x86 architectures.” *Cryptology ePrint Archive*, Report 2013/445, 2013. Available at: <http://eprint.iacr.org/2013/445.pdf>.

[12] Bernstein DJ and Lange T. “eBACS: ECRYPT benchmarking of cryptographic systems.” Available at: <http://bench.cr.yp.to>.

[13] Bhasin S, Graba T, Danger J, and Najm Z. “A look into SIMON from a side-channel perspective.” In: *Hardware-Oriented Security and Trust, HOST2014*; 2014 May 6-7; Arlington, VA; p. 56–59. IEEE;2014. doi:10.1109/HST.2014.6855568.

[14] Bhasin S, Guilley S, Sauvage L, and Danger J-L. “Unrolling cryptographic circuits: A simple countermeasure against side-channel attacks.” In: Pieprzyk J, editor. *Topics in Cryptology – CT-RSA 2010*; 2010 Mar 1-5; San Francisco, CA volume 5985 of LNCS, p. 195–207. Springer;2010. Available at: http://link.springer.com/chapter/10.1007/978-3-642-11925-5_14.

[15] Biryukov A, Roy A, and Velichkov V. “Differential analysis of block ciphers SIMON and SPECK.” In: Cid C, Rechberger C, editors, *Fast Software Encryption, FSE 2014*, 2014 Mar 3-5; London, UK; Springer;2014. Volume 8540 of LNCS, p. 546-570. Available at: http://link.springer.com/chapter/10.1007/978-3-662-46706-0_28.

[16] Biryukov A and Velichkov V. “Automatic search for differential trails in ARX ciphers.” In: Benaloh J, editor, *Topics in Cryptology - CT-RSA 2014*, volume 8366 of LNCS, p. 227–250. Springer;2014. Available at: http://link.springer.com/chapter/10.1007/978-3-319-04852-9_12.

[17] Bogdanov A, Knudsen LR, Leander G, Paar C, Poschmann A, Robshaw MJB, Seurin Y, and Vikkelsoe C. “PRESENT: an ultra-lightweight block-cipher.” In: Paillier P, Verbauwhe I, editors. *Cryptographic Hardware and Embedded Systems - CHES 2007*, volume 4727 of LNCS, p. 450–466. Springer;2007. Available at: http://link.springer.com/chapter/10.1007/978-3-540-74735-2_31.

[18] Bonneau J and Mironov I. “Cache-collision timing attacks against AES.” In: Goubin L, Matsui M, editors. *Cryptographic Hardware and Embedded Systems – CHES 2006*, volume 4249 of LNCS, p. 201–215. Springer;2006. Available at: http://link.springer.com/chapter/10.1007/11894063_16.

- [19] Borghoff J, Canteaut A, Güneysu T, Kavun EB, Knežević M, Knudsen LR, Leander G, Nikov V, Paar C, Rechberger C, Rombouts P, Thomsen SS, and Yalçın T. “PRINCE—a low-latency block cipher for pervasive computing applications (full version).” *Cryptology ePrint Archive*, Report 2012/529, 2012. Available at: <http://eprint.iacr.org/2012/529.pdf>.
- [20] Boura C, Naya-Plasencia M, and Suder V. “Scrutinizing and improving impossible differential attacks: Applications to CLEFIA, Camellia, LBlock and Simon (full version).” *Cryptology ePrint Archive*, Report 2014/699, 2014. Available at: <http://eprint.iacr.org/2014/699.pdf>.
- [21] Buhrow B, Riemer P, Shea M, Gilbert B, and Daniel E. “Block cipher speed and energy efficiency records on the MSP430: System design trade-offs for 16-bit embedded applications.” *Cryptology ePrint Archive*, Report 2015/011, 2015. Available at: <http://eprint.iacr.org/2015/011.pdf>.
- [22] Cannière CD, Dunkelman O, and Knezevic M. “KATAN and KTANTAN—A family of small and efficient hardware-oriented block ciphers.” In: Clavier C, Gaj K, editors. *Cryptographic Hardware and Embedded Systems-CHES 2009*, volume 5747 of LNCS, p. 272–288. Springer;2009. Available at: http://link.springer.com/chapter/10.1007/978-3-642-04138-9_20.
- [23] Carmer B and Archer DW. “Block ciphers, homomorphically.” Galois, Inc. Blog, December 2014. Available at: <http://galois.com/blog/2014/12/block-ciphers-homomorphically/>.
- [24] Chen H and Wang X. “Improved linear hull attack on round-reduced SIMON with dynamic keyguessing techniques.” *Cryptology ePrint Archive*, Report 2015/666, July 2015. Available at: <http://eprint.iacr.org/2015/666.pdf>.
- [25] Chen Z, Wang N, and Wang X. “Impossible differential cryptanalysis of reduced round SIMON.” *Cryptology ePrint Archive*, Report 2015/286, 2015. Available at: <http://eprint.iacr.org/2015/286.pdf>.
- [26] Chu J and Benaissa M. “Low area memory-free FPGA implementation of the AES algorithm.” In: Koch D, Singh S, and Tørrensén J, editors, *Field Programmable Logic and Applications (FPL) 2012*, 2012 Aug. 29–31; Oslo, Norway; p. 623–626. IEEE;2012. Doi: 10.1109/FPL.2012.6339250.
- [27] Courtois N, Mourouzis T, Song G, Sepehrdad P, and Susil P. “Combined algebraic and truncated differential cryptanalysis on reduced-round Simon.” In: M. S.Obaidat, A. Holzinger, and P. Samarati, editors, *SECRYPT2014*; 2014 Aug. 28–30; Vienna, Austria; p. 399–404. SciTe Press;2014. doi:10.5220/0005064903990404.
- [28] Dinu D, Corre YL, Khovratovich D, Perrin L, Großschädl J, and Biryukov A. “Triathlon of lightweight block ciphers for the Internet of Things.” *Cryptology ePrint Archive*, Report 2015/209, 2015. Available at: <http://eprint.iacr.org/2015/209.pdf>.
- [29] Dinur I. “Improved differential cryptanalysis of round-reduced Speck.” In: Joux A and Youssef AM, editors, *Selected Areas in Cryptography - SAC2014*; 2014 Aug. 14–15; Montreal, Quebec; volume 8781 of LNCS, p. 147–164. Springer;2014. Available at: http://link.springer.com/chapter/10.1007/978-3-319-13051-4_9.
- [30] Dinur I, Dunkelman O, Gutman M, and Shamir A. “Improved top-down techniques in differential cryptanalysis.” *Cryptology ePrint Archive*, Report 2015/268, 2015. Available at: <http://eprint.iacr.org/2015/268.pdf>.
- [31] Eisenbarth T, Kumar SS, Paar C, Poschmann A, and Uhsadel L. “A survey of lightweight-cryptography implementations.” *IEEE Design & Test of Computers*, 24(6): 522–533, IEEE;2007. doi:10.1109/MDT.2007.178.
- [32] Eisenbarth T and Öztürk E, editors. *Lightweight Cryptography for Security and Privacy—Light Sec2014*; 2014 Sep. 1–2; Istanbul, Turkey; volume 8898 of LNCS. Springer;2014.
- [33] Gong Z, Nikova S, and Law YW. “KLEIN: A new family of lightweight block ciphers.” In: Juels A and Paar C, editors, *RFID Security and Privacy-RFIDSec 2011*; 2011 June 26–28; Amherst, MA; volume 70555 of LNCS, p. 1–18. Springer;2011. Available at: http://link.springer.com/chapter/10.1007/978-3-642-25286-0_1.
- [34] Gulcan E, Aysu A, and Schaumont P. “A Flexible and compact hardware architecture for the SIMON block cipher.” In: Eisenbarth and Öztürk [32].
- [35] Guo J, Peyrin T, Poschmann A, and Robshaw MJB. “The LED block cipher.” In: Preneel B, Takagi T, editors. *Cryptographic and Embedded Systems - CHES 2011*; 2011 Sep 28 – Oct 1; Nara, Japan; volume 6917 of LNCS, p. 326–341. Springer;2011.
- [36] Juels A and Weis SA. “Authenticating pervasive devices with human protocols.” In: Shoup V, editor. *Advances in Cryptology - CRYPTO 2005*; 2005 Aug 14–18; Santa Barbara, CA; volume 3621 of LNCS, p. 293–308. Springer;2005.
- [37] Kölbl S, Leander G, and Tiessen T. “Observations on the SIMON block cipher family.” *Cryptology ePrint Archive*, Report 2015/145, 2015. Available at: <http://eprint.iacr.org/2015/145.pdf>.
- [38] Lepoint T and Naehrig M. “A comparison of the homomorphic encryption schemes FV and YASHE.” In: Pointcheval D and Vergnaud D, editors, *AFRICACRYPT 2014*; 2014 May 28–30; Marrakesh, Morocco; volume 8469 of LNCS, p. 318–335. Springer;2014. Available at: http://link.springer.com/chapter/10.1007/978-3-319-06734-6_20.
- [39] Maene P and Verbauwhede I. “Single-cycle implementations of block ciphers.” *Cryptology ePrint Archive*, Report 2015/658, July 2015. Available at: <http://eprint.iacr.org/2015/658.pdf>.

[40] Minematsu K. “TWINE block cipher.” Personal communication regarding results from [57], July 2014.

[41] Moradi A, Poschmann A, Ling S, Paar C, and Wang H. “Pushing the limits: A very compact and a threshold implementation of AES.” In: Paterson KG, editor. *Advances in Cryptology - EUROCRYPT2011; 2011 May 15-19; Tallinn, Estonia*; volume 6632 of LNCS, p. 69. Springer;2011. Available at: http://link.springer.com/chapter/10.1007/978-3-642-20465-4_6.

[42] Mourouzis T, Song G, Courtois N, and Christofii M. “Advanced differential cryptanalysis of reduced-round SIMON64/128 using large-round statistical distinguishers.” *Cryptology ePrint Archive*, Report 2015/481, 2015. Available at: <http://eprint.iacr.org/2015/481.pdf>.

[43] Osvik DA. “Fast implementations of AES on various platforms.” Personal communication regarding results from [44], June 2014.

[44] Osvik DA, Bos JW, Stefan D, and Canright D. “Fast software AES encryption.” In: Hong S, Iwata T, editors. *Fast Software Encryption, FSE 2010*; 2010 Feb 7-10; Seoul, Korea; volume 6147 of LNCS, p. 75–93. Springer;2010. Available at: http://link.springer.com/chapter/10.1007/978-3-642-13858-4_5.

[45] Papagiannopoulos K. “High throughput in slices: The case of PRESENT, PRINCE and KATAN 64 ciphers.” In: Saxena and Sadeghi [48], p. 137–155.

[46] Poschmann AY. *Lightweight Cryptography: Cryptographic Engineering for a Pervasive World*. PhD thesis, Bochum, Germany: Ruhr University Bochum, 2009.

[47] Rabbaninejad R, Ahmadian Z, Salmasizadeh M, and Aref MR. “Cube and Dynamic Cube Attacks on SIMON32/64.” In: 2014 11th International ISC Conference on Information Security and Cryptology (ISCISC 2014), p. 98–103 2014 Sep. 3-4; Tehran, Iran; IEEE;2014. doi: 10.1109/ISCISC.2014.6994030.

[48] Saxena N and Sadeghi A, editors. *Radio Frequency Identification: Security and Privacy Issues-RFIDSec2014*; 2014 July 21-23; Oxford, UK; volume 8651 of LNCS. Springer;2014. Available at: <http://link.springer.com/book/10.1007/978-3-319-13066-8>.

[49] Shahverdi A, Taha M, and Eisenbarth T. “Silent Simon: A threshold implementation under 100 slices.” *Cryptology ePrint Archive*, Report 2015/172, 2015. Available at: <http://eprint.iacr.org/2015/172.pdf>.

[50] Shanmugam D, Selvam R, and Annadurai S. “Differential power analysis attack on SIMON and LED block ciphers.” In: Chakraborty RS, Matyas V, and Schaumont P, editors. *Security, Privacy, and Applied Cryptography Engineering, SPACE 2014*; 2014 Oct 18-22; Pune, India; volume 8804 of LNCS, p. 110–125. Springer;2014. Available at: <http://link.springer.com/book/10.1007/978-3-319-12060-7>.

[51] Shi D, Hu L, Sun S, Song L, Qiao K, and Ma X. “Improved linear (hull) cryptanalysis of round-reduced versions of SIMON.” *Cryptology ePrint Archive*, Report 2014/973, 2014. Available at: <http://eprint.iacr.org/2014/973.pdf>.

[52] Shibutani K, Isobe T, Hiwatari H, Mitsuda A, Akishita T, and Shirai T. “Piccolo: an ultra-lightweight blockcipher.” In: Preneel B, Takagi T, editors. *Cryptographic and Embedded Systems - CHES 2011*; 2011 Sep 28 – Oct 1; Nara, Japan; volume 6917 of LNCS, p. 342–357. Springer;2011. Available at: http://link.springer.com/chapter/10.1007/978-3-642-23951-9_23.

[53] L. Song L, L. Hu L, B. Ma B, and D. Shi D. “Match box meet in-the-middle attacks on the SIMON family of block ciphers.” In: Eisenbarth T and Öztürk E [32]. Available at: http://link.springer.com/chapter/10.1007/978-3-319-16363-5_9.

[54] Sony Corporation. “CLEFIA: The 128-bit block cipher.” Available at: <http://www.sony.net/Products/cryptography/clefia/>.

[55] Sugawara T, Aoki N, and Satoh A. “High-performance ASIC implementations of the 128-bit block cipher CLEFIA.” In: *International Symposium on Circuits and Systems (ISCAS) 2008; 2008 May 18-21; Seattle, WA*; p. 2925–2928. IEEE;2008. doi: 10.1109/ISCAS.2008.4542070.

[56] Sun S, Hu L, Wang P, Qiao K, Ma X, and Song L. “Automatic security evaluation and (related-key) differential characteristic search: Application to SIMON, PRESENT, LBlock, DES(L) and other bit-oriented block ciphers.” In: Sarkar P, Iwata T, editors. *Advances in Cryptology-ASIACRYPT 2014*; 2014 Dec 7-11; Kaoshiung, Taiwan, R.O.C.; volume 8874 of LNCS, p. 158–178. Springer;2014. Available at: http://link.springer.com/chapter/10.1007/978-3-662-45611-8_9.

[57] Suzaki T, Minematsu K, Morioka S, and Kobayahi E. “Twine: A lightweight block cipher for multiple platforms.” In: Knudsen LR and Wu H, editors. *Selected Areas in Cryptography, SAC2012*; 2012 Aug 15-16; Windsor, Ontario; volume 7707 of LNCS, p. 339–354. Springer;2012. Available at: http://link.springer.com/chapter/10.1007/978-3-642-35999-6_22.

[58] Taha MMI and Schaumont P. “Key updating for leakage resiliency with application to AES modes of operation.” *IEEE Transactions on Information Forensics and Security*, 10(3):519–528, IEEE;2015. doi: 10.1109/TIFS.2014.2383359.

[59] Todo Y. “Structural evaluation by generalized integral property.” *Cryptology ePrint Archive*, Report 2015/090, 2015. Available at: <http://eprint.iacr.org/2015/090.pdf>.

[60] Wang N, Wang X, Jia K, and Zhao J. “Improved differential attacks on reduced SIMON versions.” *Cryptology ePrint Archive*, Report 2014/448, June 2014. Available at: <http://eprint.iacr.org/eprint-bin/versions.pl?entry=2014/448>.

[61] Wang N, Wang X, Jia K, and Zhao J. “Differential attacks on reduced SIMON Versions with dynamic key-guessing techniques.” *Cryptology ePrint Archive*, Report 2014/448, February 2015. Available at: <http://eprint.iacr.org/2014/448.pdf>.

[62] Wang Q, Liu Z, Varici K, Sasaki Y, Rijmen V, and Todo Y. “Cryptanalysis of reduced-round SIMON 32 and SIMON 48.” *Cryptology ePrint Archive*, Report 2014/761, 2014. Available at: <http://eprint.iacr.org/2014/761.pdf>.

[63] Wingers L. “Software for SUPERCOP benchmarking of SIMON and SPECK.” Available at: http://github.com/lrwinge/simon_speck_supercop.

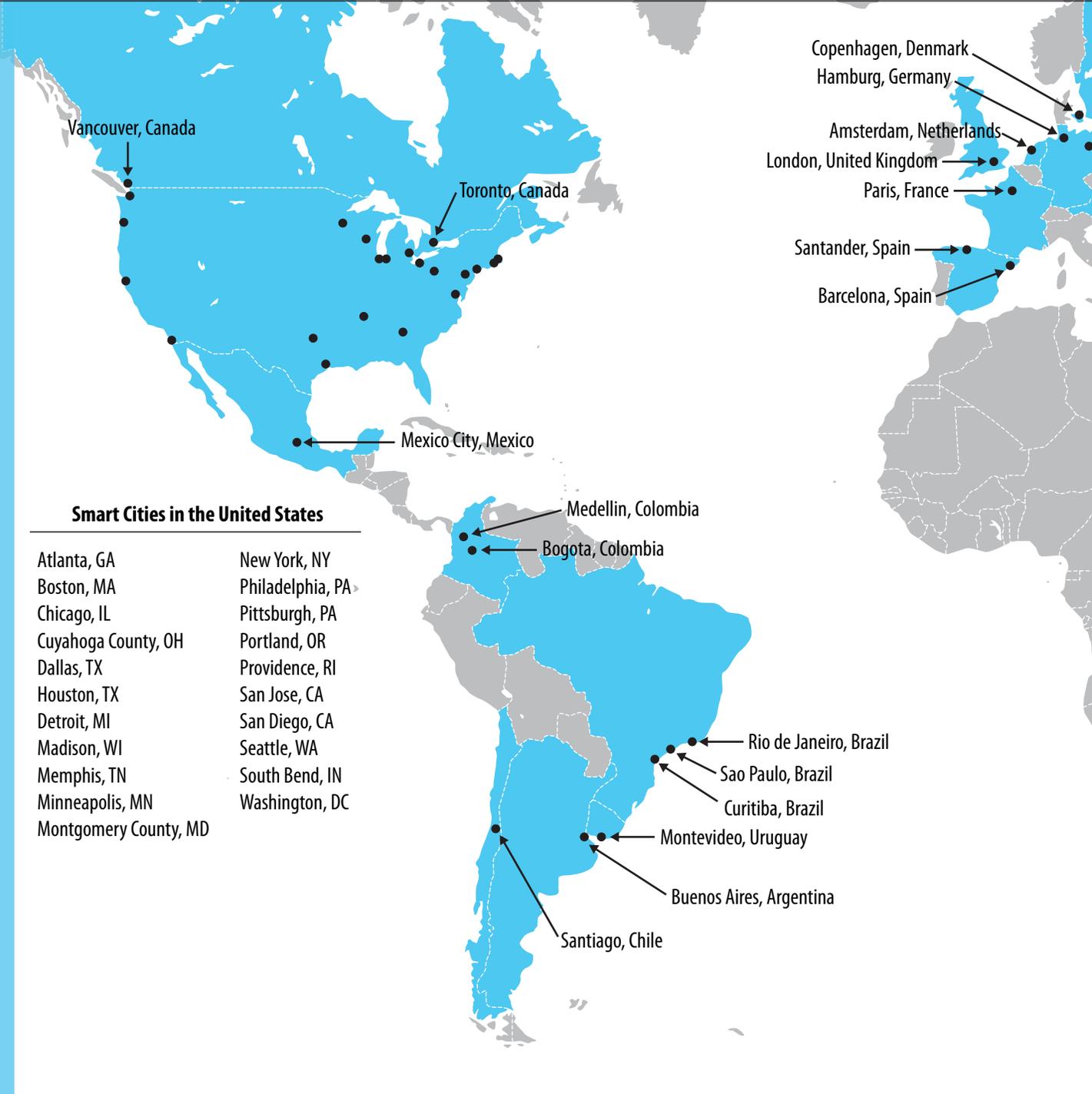
[64] Yalla P and Kaps J-P. “Lightweight cryptography for FPGAs.” In *Reconfigurable Computing and FPGAs, ReConFig '09*; 2009 Dec 9-11 Cancun, Mexico; p. 225–230, IEEE; December 2009. doi: 10.1109/ReConFig.2009.54.

[65] Yap H, Khoo K, Poschmann A, and Henricksen M. “EPCBC—A block cipher suitable for electronic product code encryption.” In: Lin D, Tsudik G, and Wang X, editors, *Cryptology and Network Security, CANS 2011*; 2011 Dec 10-12; Sanya, China; volume 7092 of LNCS, p. 76–97. Springer; 2011.

[66] Song L, Huang Z, Yang Q. “Automatic differential analysis of ARX block ciphers with application to Speck and LEA.” *Cryptology ePrint Archive*, Report 2016/209. 2016 March. Available at: <http://eprint.iacr.org/2016/209.pdf>.

Smart cities are those urban environments working to leverage Big Data and automation through the use of information and communication technologies (ICT) to enhance inhabitants' living conditions. (In the United States, some county-level governments have also established "smart city" programs.) Because each region has established varied infrastructure, communication, and urban services goals, the program criteria used in the selection of these smart cities included (see bottom of next page):

GLOBE AT A GLANCE



SMART CITIES



- ▶ Low-carbon economy initiatives
- ▶ Smart urban transit systems
- ▶ Telematics and vehicle-to-grid (V2G) technologies
- ▶ Emergency services programs
- ▶ Expansion of wireless communication services
- ▶ Sustainable energy programs
- ▶ Internet of Things and smart city solutions research and development partnerships with leading universities

GLOBE AT A GLANCE REFERENCES

[1] Newcombe T. "Santander: The smartest smart city." *Governing*. May 2014. Available at: <http://www.governing.com/topics/urban/gov-santander-spain-smart-city.html>.

[2] Cohen B. "The top 10 smartest cities on the planet." *FastCoexist*. 2012 Jan 11. Available at: <http://www.fastcoexist.com/1679127/the-top-10-smart-cities-on-the-planet>.

[3] Andrews J. "Singapore to Pilot Global Smart Cities Index." *Cities Today*. 2015 Oct 13. Available at: <http://cities-today.com/singapore-joins-global-smart-cities-index-pilot/>.

[4] Larson S. "Inside Amsterdam's efforts to become a smart city." *The Kernel*. 2015 Jan 4. Available at: <http://kernelmag.dailydot.com/issue-sections/features-issue-sections/11313/amsterdam-smart-city/>.

[5] Wardhani D. "Jakarta launches smart city program." *The Jakarta Post*. 2014 Dec 16. Available at: <http://www.thejakartapost.com/news/2014/12/16/jakarta-launches-smart-city-program.html>.

[6] Cohen B. "The 10 smartest cities in Europe." *FastCoexist*. 2014 Jan 13. Available at: <http://www.fastcoexist.com/3024721/the-10-smartest-cities-in-europe>.

[7] Cohen B. "The 8 smartest cities in Latin America." *FastCoexist*. 2013 Dec 3. Available at: <http://www.fastcoexist.com/3022533/the-8-smartest-cities-in-latin-america>.

[8] Rizvi M. "Dubai becomes first Internet of Things network city in Middle East." *Khaleej Times*. 2015 Sep 17. Available at: <http://www.khaleejtimes.com/business/technology/uae-all-set-for-internet-of-things-network>.

[9] Office of the Press Secretary, the White House. "US Government to invest \$160mn in smart cities initiative." 2015 Sep 14. Available at: <https://www.whitehouse.gov/the-press-office/2015/09/14/fact-sheet-administration-announces-new-smart-cities-initiative-help>.

[10] Office of the Press Secretary, the White House. "FACT SHEET: Administration announces new "smart cities" initiative to help communities tackle local challenges and improve city services." 2015 Sep 14. Available at: <https://www.whitehouse.gov/the-press-office/2015/09/14/fact-sheet-administration-announces-new-smart-cities-initiative-help>.

FROM LAB TO MARKET

News from the Technology Transfer Program

NSA's NiFi available via open source, improves flow of Big Data

In recent years, NSA has used open-source software (OSS) releases as an important way to move technology to the marketplace. These releases help NSA promote technologies' further development by opening them up to global review; they also help industry spur economic growth by building the OSS into products and services.

To date, one of NSA's most successful OSS releases is Niagarafiles (NiFi), a platform that eliminates artificial delays in identifying and transmitting critical, high-volume data across multiple networks. By automating and prioritizing data flows, NiFi responds to the technical and regulatory challenges Big Data has created for the public and private sectors alike.

NiFi was originally developed for NSA mission use but can help all kinds of organizations control, manage, and analyze data flow more quickly and effectively, even across geographically dispersed sites. This in turn supports real-time business intelligence—comprehensive situational awareness based on data-in-transit supporting corporate decision-making.

With the help of the NSA Technology Transfer Program (TTP), inventor Joe Witt released NiFi to the Apache Software Foundation incubator in November 2014. NSA had previously worked with Apache on the OSS release of the Agency's Accumulo project for data storage and retrieval. Apache NiFi was quickly elevated to a "top level" project, the highest status, indicating a mature technology with an active community. So far, more than 60 nonfederal OSS contributors have



developed new features, which NSA could choose to integrate into its own NiFi usage.

Joe Witt left NSA shortly after the technology's release to start Onyara, a firm built around Apache NiFi; the company grew rapidly to 10 employees. In August 2015, less than a year after NiFi's release, Onyara was acquired by Hortonworks, a Silicon Valley technology firm. Hortonworks' DataFlow product, powered by Apache NiFi, is its platform for data-in-transit, to complement Hadoop as the platform for data-at-rest. The rapid commercial success and industry adoption show the significant downstream benefits generated by NSA technology.

To learn more about technology transfer activities at NSA, please contact the TTP at tech_transfer@nsa.gov or 1-866-680-4539. 

POINTERS



Homeland Security thinks start-ups are best source for IoT security solutions

The Department of Homeland Security (DHS) wants to be able to detect all devices connected to its network in a particular location, such as an airport, and thinks start-ups know how to do it. DHS, which in 2015 set up an office in Silicon Valley, is looking for companies whose technology not only detects devices and sensors, but also verifies and authenticates them, prevents spoofing, and updates devices' security systems. Ideally, the system would map out the location of those devices, work well for nontechnical users, and not disrupt other devices.

As part of its plan to entice start-ups, in December 2015 the Agency unveiled plans to use small, short-term technology contracts to bypass the lengthy administrative

process associated with traditional contracting. This would involve small awards (\$50,000 to \$200,000) to companies for three- to six-month performance periods. According to the notice, a project that gets four rounds of funding could get up to \$800,000 over 24 months, at the end of which the technology could be deployed for testing or acquired by another group. DHS also plans to use the new contracting system to collect technology related to first responders, aviation and drone security, and fighting biological threats.

For more information, visit <http://www.nextgov.com/emerging-tech/2016/01/dhs-think-start-ups-know-how-best-protect-internet-things/124888/>.

Consumers' security fears are curtailing IoT sales

Although consumers flocked to the gadget-laden Consumer Electronics Show (CES 2016), held in Las Vegas, Nevada from 6–9 January, a recent survey by Accenture suggests that fears about security may keep them away from IoT devices. The study was conducted between October and November 2015, with 28,000 consumers in 28 countries participating. Forty-seven percent of participants cited privacy and security concerns as a barrier to adoption of IoT products. Price and ease of use were also cited as reasons for not adopting new technology.

The study also found that of consumers who are aware of recent security breaches, about 66% were less likely to adopt or keep an IoT device. Of these persons, 18% stopped using such a product until better security could be guaranteed because the risk of ownership was not

worth the potential reward. An additional 24% reported delaying a purchase until security improves.

Levels of concern vary. Thirty-seven percent of participants said they would be more cautious when using an IoT device, while 21% said they are not concerned about security breaches and hackers.

For more information, visit <http://www.scmagazine.com/consumers-security-fears-are-curtailing-iot-sales-report/article/463226/>.



Nokia's IoT security tool takes in the whole network

Nokia is taking the principle of “united we stand” to the IoT with a platform that can harness systems from multiple vendors for network-wide security. The company's NetGuard Security Management Center is designed to monitor and control all the security components on a network. This will help carriers and other IoT service providers take a more holistic approach to preventing and responding to attacks. NetGuard will be demonstrated at the Mobile World Congress in Barcelona and is expected to ship later this year.

NetGuard is a new product, mostly software, intended for any large organization that operates an IoT application, collects the data from IoT end nodes, or is building

connectivity for an IoT network. The platform monitors all IoT devices, analyzes activity using a malware database from F-Secure, draws correlations between events in different parts of the network, and can set security parameters to minimize the chance of successful attacks. There is a decision-making engine that can automatically configure security settings and decide how to respond to threats. Alternatively, administrators can just monitor all network security information on a unified dashboard and make changes manually.

For more information, visit <http://www.pcworld.com/article/3029072/nokias-iot-security-tool-takes-in-the-whole-network.html>.

ITU to develop IoT standards

Members of the International Telecommunications Union (ITU) have established a new ITU-T Study Group to address the standardization requirements of IoT technologies, with an initial focus on IoT applications in smart cities. ITU-T Study Groups develop international standards (ITU-T Recommendations) that underpin the interconnection and interoperability of ICT networks and devices.

The new group is titled “ITU-T Study Group 20: IoT and its applications, including smart cities and communities.” It will be responsible for international standards to enable the coordinated development of IoT technologies, including machine-to-machine communications and ubiquitous sensor networks. The group will develop standards that leverage IoT technologies to address urban-development challenges. A key part of this study will be the standardization

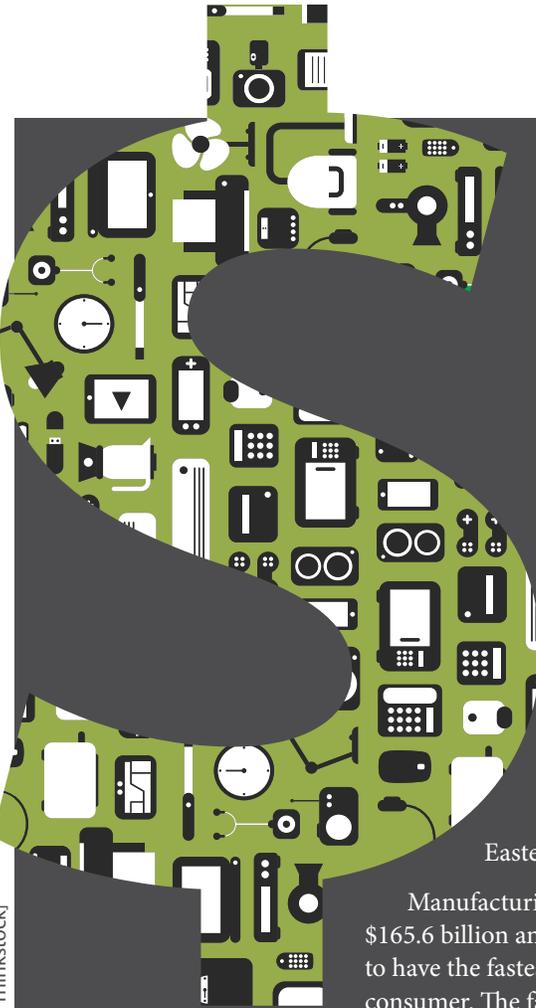
of end-to-end architectures for the IoT and mechanisms for the interoperability of IoT applications and data sets employed by various vertically oriented industry sectors.

The decision to create a new ITU-T Study Group was made by the Telecommunication Standardization Advisory Group (TSAG) at its meeting at ITU headquarters in Geneva, 2–5 June 2015. TSAG has the authority to modify ITU-T's structure and work program between quadrennial World Telecommunication Standardization Assemblies, giving ITU-T the agility required to reflect the changing priorities of its membership.

In May 2015, Dubai became the world's first city to assess the efficiency and sustainability of its operations using the key performance indicators developed by the ITU-T Focus Group on Smart Sustainable Cities (FG-SSC). The two-year pilot project will evaluate the feasibility of the indicators with the aim of contributing to their international standardization.

For more information, visit <http://www.lightreading.com/iot/iot-strategies/itu-to-develop-iot-standards/d/d-id/716263> and <http://www.itu.int/en/ITU-T/about/groups/Pages/sg20.aspx>.





IoT spending expected to reach nearly \$1.3 trillion in 2019

According to a new International Data Corporation (IDC) Spending Guide, worldwide spending on the IoT will grow at a 17% compound annual growth rate (CAGR) from \$698.6 billion in 2015 to nearly \$1.3 trillion in 2019.

Asia/Pacific is the clear leader in IoT spending, with more than 40% of the worldwide total in 2015. North America and Western Europe are the second and third largest regions, with combined spending of more than \$250 billion in 2015. The regions expected to experience the fastest growth in IoT spending over the five-year forecast period are Latin America (26.5% CAGR), followed by Western Europe and Central and Eastern Europe.

Manufacturing and transportation led in worldwide IoT spending, with 2015 totals of \$165.6 billion and \$78.7 billion, respectively. Over the next five years, industries forecast to have the fastest IoT spending growth will be insurance (31.8% CAGR), healthcare, and consumer. The fast-expanding consumer IoT market will be the third largest spending category by the end of the forecast period.

In addition to use cases identified above, “connected vehicles” was among the fastest growing IoT use cases across five of the six geographic regions. This broad category includes emergency, infotainment, security, vehicle-to-vehicle (V2V), and vehicle-to-infrastructure (V2I) applications.

For more information, visit <https://www.idc.com/getdoc.jsp?containerId=prUS40782915>.

Machina Research: M2M's share of roaming doubled in 12 months

A new study from Starhome Mach and Machina Research said mobile network operators should pay more attention to the impact of machine-to-machine (M2M) roaming on their networks, as M2M's share of roaming doubled in the last 12 months. Machina Research estimates that there are now 350 million cellular-based connections worldwide, and it expects this to grow to 1.3 billion over the next five

years. The company added, however, that the proportion of M2M connections accounted for by roaming is growing even faster.

Though this growth is good news for network operators, caution is required in analyzing the market implications. Roaming M2M devices have different usage profiles from human roamers; some devices are heavy users of data, but

others may send and receive very little while still being heavy consumers of “free” signaling resources on the visited network.

For more information, visit <http://www.fiercewireless.com/europe/story/machina-research-m2ms-share-roaming-doubled-12-months/2016-01-11>.

Gartner says smart cities will use 1.6 billion connected things in 2016

Gartner, Inc. estimates that 1.6 billion connected things will be used by smart cities in 2016, an increase of 39% from 2015. “Smart commercial buildings will be the highest user of the Internet of Things (IoT) until 2017, after which smart homes will take the lead with just over one billion connected things in 2018,” said Bettina Tratz-Ryan, research vice president at Gartner.

Business applications fueling the growth of the IoT in commercial buildings are handled through building information management systems that drive operations management, especially around energy efficiency and user-centric service environments. In 2016, commercial security cameras and webcams as well as indoor LEDs will drive total growth, representing 24% of the IoT market for smart cities.

Smart homes will represent 21% of total IoT use in smart cities in 2016 and will record the highest increase over the next five years. In smart homes, the consumer IoT applications that are fueling growth are smart TVs, smart set-top boxes, smart bulbs, and various home automation tools, such as smart thermostats, home security systems, and kitchen appliances.

More detailed analysis is available in the Gartner report “Forecast: Internet of Things—endpoints and associated services, worldwide, 2015.”

For more information, visit <http://www.gartner.com/newsroom/id/3175418>.



[Photo credit: Comomolas/Stock/Thinkstock]

In Denmark, RFID triggers traffic lights when bikes approach

At an intersection in Aarhus, Denmark's second largest city, traffic lights are automatically going green for cyclists. This is thanks to 2Green—a passive radio frequency identification (RFID) solution provided by Danish firm ID-advice. Two hundred local cyclists are taking part in the pilot study, part of a wider European IoT development effort known as Radical that is slated to run in six countries through February 2016.

RFID tags have been attached to each bicycle's front wheel and to RFID readers installed at the intersection. When a cyclist approaches, the system prompts the traffic-signaling software to switch the light facing the bike to green, while turning the cross-traffic light to red. In certain instances the traffic light can override 2Green; for example, when an emergency vehicle is passing through. The RFID reader also forwards data via a cellular connection to the Radical program's server as well as to CKAN, a web-based open-source data management platform where city managers and



[Photo credit: AndrewJShearer/iStock/Thinkstock]

the public can access traffic data.

For more information, visit <http://www.rfidjournal.com/articles/view?13849/3>.

Virginia start-up raises funds for battery-free IoT

PsiKick Inc. (Charlottesville, Virginia), a semiconductor start-up working on sub-threshold voltage operation wireless circuits, has raised \$16.5 million in Series B financing led by Osage University Partners and joined by existing investors. The firm was founded in 2012 by professors at the University of Virginia and the University of Michigan. In 2014, the company raised a Series A round of finance reportedly worth \$5.25 million. Total funding is now over \$22 million.

The company claims its proof-of-concept chip design would consume between 100 and 1,000 times less than any comparable chip. PsiKick is working on systems that scavenge energy from multiple sources including indoor light, radio-frequency rectification, thermal gradient,

and piezoelectric vibration. One such system is a battery-less electrocardiogram (EKG) sensor that supports a one megabit per second data rate over a 10 meter distance.

Other companies in this field include fabless start-up Ambiq Micro Inc. (Austin, Texas) and ARM Holdings plc (Cambridge, England). Ambiq has launched the Apollo line of Cortex-M4F based microcontrollers, claiming they offer a 10 time reduction compared with other microcontrollers. ARM has been working on a processor core optimized for operation close to the threshold voltage of complementary metal-oxide semiconductor transistors and at clock frequencies of the order of tens of kilohertz.

For more information, visit http://www.eetimes.com/document.asp?doc_id=1328565.

Thread and Zigbee standards will start cooperating to simplify smart home control

Consumers have more connected-home products than ever to choose from, but the technology is often so complicated that only tech enthusiasts buy it. The gear needs to become simpler and easier to use, but there are so many standards in play that streamlining a user's experience can be hard to do. Thread and ZigBee hope to make the market less fragmented. The Thread Group and the ZigBee Alliance plan to make their technologies work together in connected home products, which could help bring more order to the IoT. The organizations announced that they are collaborating to make a ZigBee application layer run over Thread networks.

ZigBee Cluster Library (ZCL), an application layer used in ZigBee products, will be able to work with Thread's mesh networking protocol. ZCL defines how connected devices, such as light bulbs and thermostats, tell each other what they are and what they can do. Many different application layers can run on top of Thread, and the partnership is not exclusive for either party. But the combined

technology might become an attractive package for streamlined development of home IoT products.

The two groups have been talking about a partnership for a long time, said Chris Boross, who is president of Thread Group and technical product marketing manager at Google's Nest division. Branding strategy for products that use both technologies has yet to be determined, he said. Thread was founded in 2015 by large vendors including Google, Samsung Electronics, and ARM Holdings, and has since added Huawei Technologies, Whirlpool, and Philips. The organization now has more than 80 members.

For more information, visit <http://www.pcworld.com/article/2905692/thread-and-zigbee-snap-pieces-together-to-boost-home-iot.html>.



NATIONAL SECURITY AGENCY



CENTRAL SECURITY SERVICE

Defending Our Nation. Securing The Future