Low-Energy Security: Limits and Opportunities in the Internet of Things

Wade Trappe, Richard Howard, and Robert S. Moore | Rutgers University

Many new "networkable" devices, which constitute the Internet of Things, are low energy and lightweight. These devices must devote most of their available energy and computation to executing core application functionality, making the task of affordably supporting security and privacy quite challenging.

Progress in wireless technologies, coupled with device miniaturization, has led to a pervasive communication fabric that encompasses everything from automobiles and environmental sensors to medical devices and personal communication. Consequently, computing is becoming centered on the huge amount of information captured and made accessible by the vast array of devices that form the Internet of Things (IoT). The IoT allows for new applications that will tackle societal needs through unprecedented access to data. However, as new technologies emerge at an ever-increasing rate, there will be a commensurate increase in cybersecurity attacks.

Unfortunately, it's impossible to design applications that engender trust in their effectiveness without trust in the underlying operation, connectivity, and data produced by the Internet of Things. The IoT will include some familiar devices such as smartphones, which are well-resourced and can be readily powered and recharged. However, it will also include embedded wireless devices in the previously unconnected world for which the resource paradigm is quite different. These devices must be highly affordable and thus will be limited in terms of energy, computing, size, and storage.

In this article, we present the viewpoint that current cryptographic tools can't easily secure the "low end" of the IoT, as the operating energy and computational regime aren't conducive to traditional security approaches. The low-end, low-energy, and lightweight computing that will characterize the edge of the IoT will come with considerable restrictions on function design. These devices must devote most of their available energy and computation to executing core application functionality and might have little left over to support security and privacy.

IoT Security

A natural starting point for identifying IoT security concerns is prior work on wireless ad hoc and sensor network security. Many security concerns that the IoT faces are identified in "The Resurrecting Duckling: Security Issues for Ad-Hoc Wireless Networks," which presented a look into the future of securing embedded wireless devices and examined confidentiality, integrity, authenticity, and availability threats.[1] Similarly, "SPINS: Security Protocols for Sensor Networks" provided a detailed exploration in the context of highly resource-constrained SmartDust sensors.[2]

Building on the framework established in these

 Copublished by the IEEE Computer and Reliability Societies

papers, we outline three types of threats that the low end of the future wireless Internet faces:

- Confidentiality. Wireless communication between entities is especially susceptible to confidentiality threats from attackers snooping for message content. As messages are broadcast over the air, malicious adversaries can easily intercept packets. For unencrypted communication, adversaries can decipher content by listening to broadcast packets.
- Integrity, authentication, and nonrepudiation. IoT devices must establish integrity, authentication, and nonrepudiation at various levels. This requires the ability to authenticate devices and thereby support nonrepudiation, ensure the integrity of message communication between senders and receivers, and ensure trust in the data values entering the IoT system.
- Availability. Adversaries can launch RF-specific denial-of-service attacks that target IoT radio devices' ability to transmit or receive messages. Similarly, attacks could involve reprogramming wireless devices to operate with a faster duty cycle, thereby increasing the channel utilization or flooding back-end servers.

The IoT involves an unprecedented inter-networking of objects, making their data more accessible to the broader Internet. Many of these devices will be deployed and forgotten, providing information in an unattended, semicontinual manner with very limited maintenance. As such, the IoT subsumes the previous and separate notions of sensor networks, mobile networks, and RFID systems; securing the IoT will require techniques that support these areas in clever new ways.

Energy Concerns Can't Be Easily Solved by Other Technologies

Although IoT devices range from lowly RFID transponders to more resourced smartphones and tablets, we focus on the most lightly resourced and inexpensive devices. Engineers working with platforms such as smartphones and tablets have a far easier task than those beginning to work with the new generation of IoT nodes.

Consider the difference between a modern smartphone, such as a Samsung Galaxy S5, and a miniature sensor tag capable of reporting information such as presence, temperature, and humidity for decades.[3] The smartphone has a 2.5-GHz quad core processor, 2 Gbytes of RAM (with up to 128 Gbytes of SD card), and a battery with 30 kJ of stored energy that's typically recharged daily. The tag has a single 16-bit processor, often running at 6 to 12 MHz to save energy, with 512 bytes (not megabytes or gigabytes, but bytes!) of RAM and 16 Kbytes of flash for program storage.

The phone runs roughly 10 hours while Web browsing, but the IoT node must deliver data over a wireless link for at least 10,000 hours on a coin cell battery with less than 1/15 the energy of that in the phone—and it never gets recharged. A few million instructions to execute security protocols is an insignificant drain on a phone's resources but would be intolerable for a tag.

Three approaches address this problem. The first approach, currently widely used, is providing minimum security to these ubiquitous sensing and actuating nodes in the belief that such information is of little concern to attackers. Several recent examples bring this viewpoint into question. One demonstration showed that attackers can hack passive tire pressure sensors in cars and use this as a gateway into the automotive system.[4] In another case, attackers passively read wireless electric meters in residential applications at distances of hundreds of meters, developing a detailed pattern of resident energy use and activity and thereby undermining personal privacy. These are just a few issues to emerge in the very near term as IoT devices are more widely deployed.

Another approach to address this problem is to depend on electronic and battery performance increases that will allow existing security approaches to adequately adapt to such miniature devices. After nearly 50 years of Moore's law scaling and creating ever smaller devices in ever denser circuits, we're reaching fundamental limits. The official Semiconductor Roadmap (www.itrs.net) shows feature sizes as small as 7 nm, placing limits on what can be produced reliably on a wafer. Of course, performance progress will continue, and new materials such as graphene will be incorporated. However, the simple and relatively low-cost path of scaling to smaller sizes is reaching a natural end, and continued progress will occur at a slower and costlier pace. Devices such as low-cost IoT nodes will probably be the last to benefit from these advances until the technology becomes mainstream—long after there are hundreds or thousands of nodes per person.

The next area where performance increases are expected, but disappointment likely, is in energy sources. Unlike semiconductor technology, batteries are a mature technology with centuries of engineering behind them. The periodic table offers a limited number of elements, and their potential for making batteries has been long known. As Figure 1 shows, over the past three decades, careful engineering has brought an increase in useful battery capacity of only about 7 percent per year, and we're already utilizing some of the highest-energy materials available.[5] Getting just one of the many decades of performance needed in this way is highly unlikely, even with alternate energy storage technologies (see Figure 2).

The final approach is to harvest energy from the

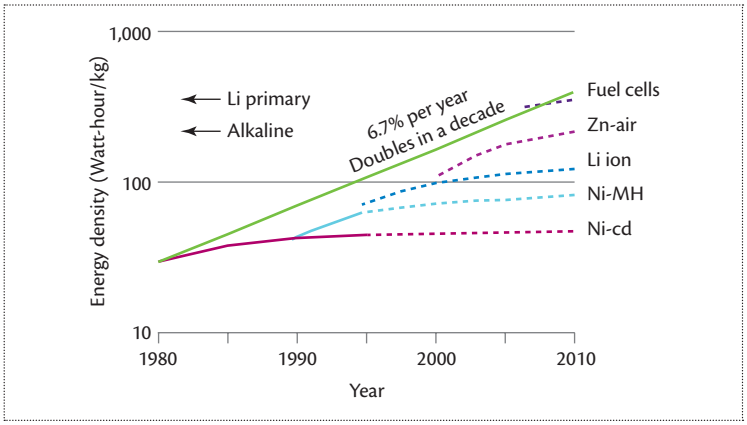

Figure 1. Energy densities for different battery technologies over the past three decades. For example, a Tesla automobile's Li ion battery is reported to have a little more than 100 Watt-hour/kg, consistent with the Li ion curve. (Brijesh Vyas, private communication, Alcatel-Lucent, 2000.)

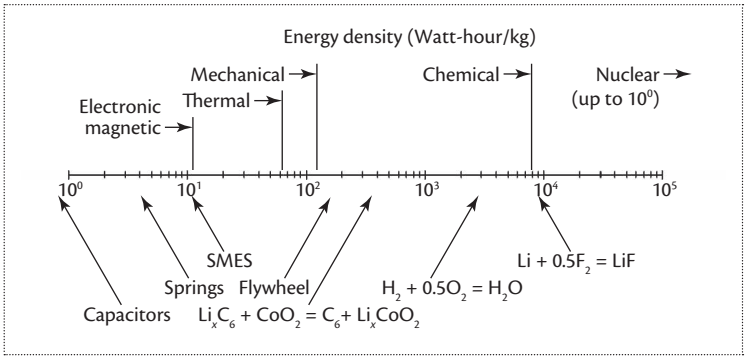

Figure 2. There are strict limits on the amount of energy that can be stored using conventional methods, and any dramatic shift requires exploiting new and potentially risky methods. We can verify these limits by examining standard material science references in combination with conventional physical principles. For example, we can verify the lithium fluoride calculation by referencing the heat of formation in the National Institute of Standards and Technology's Chemistry WebBook (http://webbook.nist.gov) and converting it to Watt-hour/kg. (Brijesh Vyas, private communication, Alcatel-Lucent, 2000.)

environment—from either manmade or natural sources. In the first case, a system carefully adds extra energy to the environment in a form that IoT nodes can use efficiently. These systems fall into two broad categories. The first of these added energy approaches is based on low-frequency electromagnetic energy in which a base station is much closer than a wavelength to the sensor node. The coupling's efficiency decreases rapidly with distance; a good rule of thumb is that it's useful only for distances a few times the size of the antenna, making it of limited utility for powering remote IoT nodes.

The second type of added energy system uses broadcast radio frequency electromagnetic radiation to energize an IoT node. Powering a system through radio energy is an old idea, and it has practical limitations when applied to low-cost, small IoT nodes in complex environments. The most visible commercial applications of this type are passive RFID tags, wherein a base station transmits radio energy at a frequency of a few hundred megahertz to a few gigahertz to a passive (battery-free) tag. The tag collects the energy with its antenna and converts it into DC electrical energy that can power the microprocessor and RF electronics. To communicate back to the base station, the tag modulates the characteristics of its antenna, changing the amount of RF energy reflected back to the base station.

Unfortunately, the reality of radio propagation and safety regulations severely limits the capability of such a system. The fundamental constraint is that radio energy decreases in density by at least the square of the distance. Because real-world attenuation for omnidirectional radiation is typically worse than $1/r^2$ and antennas must be properly oriented relative to the base station, performance is low except at ranges of a few meters. For instance, a base station emitting 4 W to power IoT RFID tags would support distances of up to 3 meters in a $1/r^3$ environment, which frequently occurs in indoor and urban settings, whereas 100 W is necessary for a range of roughly 10 meters—far in excess of safe and legal exposure limits.[6] This large radio energy is needed just to allow a node to modulate the radio signal and reflect it back. The situation becomes even more dire when one must perform basic security functions. We can reduce the power levels by using a directed beam along a line of sight, but this is dangerous and severely limits the IoT nodes' deployment options.

The second approach for energy harvesting is to try to tap into existing energy sources. A cursory look at most environments reveals an apparent wealth of wasted energy—light, heat, vibration, wind, water, and so forth. Similarly, there are many examples of successful harvesting of these resources. All these have practical limitations in terms of IoT application, but we focus here on light harvesting, as it appears to be easier to implement at the scale needed for IoT nodes.

At noon on a clear day, the sun provides nearly 100 mW/cm^2. Photovoltaic cells can convert this energy directly into usable electricity with an efficiency that ranges from a high of nearly 25 percent in single crystals to approximately 1 percent for low-cost, flexible organic films.

Although light harvesting is useful for many applications, some significant complications limit its usefulness with miniature, low-cost sensor nodes. First, the sun is up only in the daytime. Beyond this, there are clouds, shadows, the sun's motion, and accumulated dust on the collector, which further limit the output to a small fraction of the theoretical maximum. For instance,

the observed daily solar energy gathered in New York City averages approximately 12 mW/cm^2 for a cell aimed at the sun in winter. Indoor lighting is usually more dependable than outdoor, but the power levels are much lower; further reduction occurs because photovoltaic cells lose efficiency (as much as a factor of 10) at low light levels, and the rechargeable batteries needed to integrate the energy have their own internal leakage.

Even so, this might be a useful amount of energy for a node[7] if the collector is properly installed and avoids shadowing and if its surface is kept clean. However, the applications in which such harvesting would help are those in which IoT nodes won't be exposed to much light; examples include bridge corrosion, water leaks, object tracking, and other tasks in which IoT nodes are best placed in a space generally unoccupied by people.

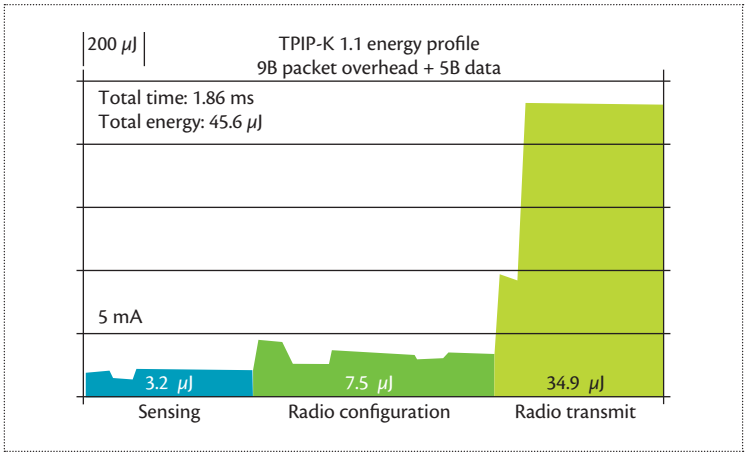

Figure 3. Measured energy profile for a simple IoT device that performs sensing for temperature and open or closed doors and immediately broadcasts.

An Example IoT Device Energy Requirement Breakdown

How do IoT nodes use their precious joules to accomplish their tasks? Let's walk through the energy budget of a low-power, long-lived sensor node.[3] For simplicity, assume that the energy to collect data is insignificant. Estimating the energy to send the signal is easy, but a working node must do more. The data must be read from memory, and then framed into a packet with enough information for the receiver to capture and decode it. The information must be moved from the processor to the radio—a complex device that typically has its own state machine—which must be powered up, stabilized, and even calibrated to meet frequency regulations. Then, the actual data must be sent over the air. The last step is typically under 10-percent efficient for commercial chips, although some prototype research devices have shown nearly 50-percent efficiency.

We examine a simple IoT device based on the MSP430g2553 16-bit microcontroller (MCU) and CC1150 radio—both representative of leading low-energy components (www.inpointsys.com). This device is characteristic of the simplest class of IoT device: it measures data and periodically broadcasts this data. Based on Texas Instruments' data sheets,[8] at 12 MHz, the MSP430 requires approximately 2.8 mA, and the CC1150 requires approximately 23 mA to broadcast at 900 MHz and 6 dBm. At 250 Kbps, a 14-byte frame (8-byte sync/preamble, 1-byte length, 3-byte ID, 2-byte data) requires 448 μs, which at 23 mA (plus 2.8 mA for the microcontroller) would be 34.7 μJ of energy.

To validate this estimation, we attached one such device to a regulated 3-V power supply with an inline 10-Ω resistor and measured the delivered current on an oscilloscope. Figure 3 shows the resulting measurement and includes an 8.4 mA 75 μs frequency synthesizer settle time. The radio broadcast uses approximately 75 percent of the energy, and the MCU framing the data and delivering it to the radio uses approximately 25 percent. There's no provision for encoding or protecting the data in any way.

At the microcontroller, each 16-bit operation at 12 MHz uses approximately 0.7 μJ. Suppose we double the MCU's overhead to allow for security mechanisms for each packet. This would give a security engineer less than 20,000 operations, at 16 bit, to ensure data security, privacy, and validation. To understand the enormity of this challenge, we highlight a recent lightweight SSL implementation for a low-end device, which required roughly 16 million operations![9]

Conventional Cryptography Fails to Port

Conventional cryptography is essential to secure the Internet. We're big fans of using cryptography in protocol suites such as IPSEC, TLS, and HTTPS on well-resourced devices. Unfortunately, many aspects of modern cryptography make porting over to ultra-low-end devices difficult. Perhaps foremost is the challenge of implementing a basic encryption algorithm, such as a block cipher, on such a low-end device. Again, a typical IoT device might have only 512 bytes of RAM, very limited code space, and very limited resources to devote to security. Some of this memory will have to be dedicated to storing any cipher keys, which can be a large fraction of memory to dedicate to noncore functionality. Further, many encryption algorithms require lookup tables for efficient implementation—notably, the old and familiar Data Encryption Standard algorithm involves large S-boxes and permutation tables, the typical Advanced Encryption Standard (AES) Rijndael involves 800 bytes of lookup tables, and more efficient implementations require even more storage for lookup tables.

The RC5 algorithm has been a popular choice in the RFID and sensor security literature for avoiding storage penalties associated with lookup tables, as it has small code size and is computationally efficient. Unfortunately, it involves 32-bit rotations, which can be a challenge to implement on 8-bit or 16-bit devices. In spite of its efficiency, recent results have shown that in a typical RFID setting using energy harvesting, the RC5 algorithm failed to complete its computations because not enough energy was harvested to perform flash writes.[10] Thus, even for a favorite of strong conventional ciphers, the energy requirements are still beyond what can be performed on lightweight IoT devices.

Researchers have made considerable efforts to develop lightweight cryptographic algorithms suitable for resource-constrained devices such as RFID tags.[11] The track record isn't all that encouraging, as almost an equal quantity of papers identify lightweight algorithms' vulnerabilities.[12] The shortcomings detailed in much of the lightweight wireless security literature can be categorized as weaknesses in the actual cryptographic algorithm, problems with initial bootstrapping assumptions, and protocol flaws. Speaking to the first weakness, we reiterate cryptographic security's all-or-nothing nature: using an algorithm with a small key size—say 40 to 64 bits—is essentially no security.

Although we might overcome these challenges eventually, we must remember that the cryptographic community is inherently pessimistic and particularly creative when it puts on its adversarial hat. In general, security—and cryptography in particular—is difficult with an infinite resource budget, and thus we revisit the popular use of RC5 for IoT-type devices. Even good algorithms like this, with its data-dependent rotations, could be subject to timing attacks because its execution time depends on the input data. Device-level IoT designers aren't likely to consider factors such as side-channel attacks during cipher selection and implementation. The challenges of rebuilding a cryptographic algorithm implementation to remove information leakage due to timing would greatly use up the available resources (notably, code space) on resource-limited devices. Thus, we're skeptical that cryptography—and lightweight cryptography in particular—will successfully migrate and be implemented without weakness on a low-end IoT device.

Another approach to using cryptography warrants discussion—the use of dedicated cryptographic circuitry, either on chip or in a cryptographic coprocessor. Some studies, such as "Energy Comparison of AES and SHA-1 for Ubiquitous Computing," have shown that dedicated cryptographic hardware can in principle significantly reduce cryptographic algorithms' energy costs and don't incur the code space storage penalties associated with algorithm lookup tables.[13] However, it poses challenges for data formatting as well as presentation in the short word-length environment typical for IoT devices.

To get a sense of the energy savings available in custom cryptographic circuitry, note that the circuitry described in "Energy Comparison of AES and SHA-1 for Ubiquitous Computing" does AES encryption on 29-byte packets at an estimated power consumption of only 24 μW, whereas just turning on a 16-bit (2-byte) ultra-low-power microcontroller (TI MSP430g2553) at the same clock frequency consumes roughly 10 times the power.

The benefits of such dedicated circuitry are adequate to reach the IoT energy goals we outlined but would have to be implemented in integrated circuit technology compatible with the IoT systems' other requirements (for example, low cost and ultra-low sleep current), possibly integrated with highly efficient radios, and compatible with standardized interfaces and algorithms that would allow the production volumes to bring the costs into the range necessary for ubiquitous applications. This is an exciting research direction that cuts across digital, cryptographic, and RF domains and would greatly benefit from close interaction between research and the industries responsible for implementation at scale.

How to Change the Game

The question remains: What can we do to secure low-end IoT devices where they touch the broader Internet? If we want to support security at the low-end device, an alternative approach is to reuse existing functions and thereby not introduce additional energy burden or be very selective about what additional functionality we employ. We might also attempt to exploit the inherent asymmetry in the deployment scenario, in which low-end devices typically communicate to more powerful base stations or back-end servers that don't have the low-end devices' energy and computational restrictions.

Free Security Might Be All We Can Afford

An ideal approach is to use existing functions for security; a natural choice is to leverage radio communication. We can apply signal processing at the receiver to authenticate whether a transmission came from the expected transmitter in the expected location. For example, it's possible to exploit the uniqueness of the channel between an IoT node (Alice) and its receiver (Bob)—such as arises from multipath fading—as an authenticator to distinguish between a legitimate and illegitimate transmitter. Alternatively, we can use the specific analog characteristics of a transmitter or an auxiliary circuit to effectively encode analog information. Because these analog nuances can't be predicted or controlled in manufacturing, they essentially act

as physically unclonable functions and can serve as a unique key, thereby supporting new authentication styles and even nonrepudiation services. This type of physical-layer authentication has little or no energy overhead because it allows reuse of radio signals—including preambles and pilots, which are needed for the channel equalization necessary in modulation decoding—to support authentication.[14,15] The practical challenge is to provide physical-layer characteristics that are stable or predictable over the whole operation range of temperature, voltage, and decades of aging.

Building on a similar concept, it might be possible to reuse other mechanisms in the communication. In particular, we might support the communications' integrity through anomaly detection, in which we strive to gain confidence that the communication between a low-end device and its receiver is legitimate. For example, many wireless protocols use sequence numbering to reject duplicate packets, and the sequence number field tends to be incremented each time a packet is transmitted. In this case, the sequence numbering's monotonicity supports easy anomaly detection. Even if a spoofing adversary can create a false packet with the next sequence number that a legitimate device would use, the legitimate device will follow its own sequence progression, and thus we would observe repeated duplicate sequence numbers. This simple observation has been used for anomaly detection in Wi-Fi networks and can easily be applied to other settings.

> **The IoT's future will rely on our ability to adequately secure hard-to-secure, resource-sparse devices.**

The adversary's challenge is to launch a doppelganger attack by determining how to imitate the original legitimate device, and then terminate that legitimate device and seamlessly assume its role and functions. This task is quite challenging, reminiscent of the famous scene from *Raiders of the Lost Ark* in which Indiana Jones attempted to swap the idol for a bag of sand. In the IoT setting, if the monitoring station has some knowledge of the next expected IoT device transmission (for example, periodic beacons), then an attacker must disable the legitimate device and time the assumption of the roles precisely.

There's something to be said for safety in numbers, and it might be possible to use these low-end IoT devices' affordability and abundance for a security advantage. Using many IoT devices—each submitting its own data from potentially differing sources and data types—enables processing at the data repository to determine whether the data is potentially suspect. Ensuring the IoT data's integrity then involves a form of consistency checking to determine whether one or even a few data records fall outside a region of likely valid measurements. For example, we might employ outlier detection schemes to identify suspect data or use a rule-based method to perform classification. These schemes could look for consistencies across time, spatial deployment, or even different measurement modalities. Should some measurements appear suspect, an appropriate administrator could be notified, or the data could be purged or cleansed.

Supporting the data integrity reported by IoT devices can be very powerful as many physical properties exhibit correlation with other physical properties, which might corroborate the values produced by a different type of measurement. This approach places the burden on the data consumer, and one might envision a cloud service analyzing IoT data and tagging suspicious data with confidence measures.

Safety in numbers can support IoT availability during interference or denial of service. It's absolutely a bad idea to combat jamming using the puny capabilities of a low-end device. Incorporating any form of anti-jam countermeasures, such as modulation with robust anti-jam margins, comes with a cost and implies that the device is less efficient than it otherwise could be in the absence of an attack. Incorporating adaptivity in the IoT node is also a bad idea, as it leads to the well-known sleep deprivation attack.[1]

A better place to cope with interference is at the receiver. With a single receiver, one simple approach is to employ a "bit checkpointing" strategy motivated by standard fragmentation mechanisms, so that if a portion of the packet is not interfered with, it might successfully be identified and used. With multiple base stations at different locations, distributed signal processing techniques in the vein of multiple input, multiple output methods become possible and might provide sufficient processing gain against the jammer. These methods require implementing challenging synchronization and sharing of raw signals to be processed at a centralized location. Researchers are exploring such techniques in the context of cloud radio access networks.

If You Have to Spend, Be Thrifty

If you can spend a little energy to introduce new security features or components in a low-end device, then you must be very selective. There are several useful application-oriented observations to keep in mind:

although low-end IoT devices will be widespread, each individual device won't likely be responsible for too much, the data dynamics will likely be limited (for instance, how much does temperature change in five minutes?), and the data value might rapidly decay with time.

The lack of dynamics suggests that we can be selective with what's encrypted or hashed and transmitted and thereby achieve energy economy by performing cryptographic operations only when significant new data needs to be exchanged. The amount of savings is directly related to the data stream's novelty!

This idea offers great potential, particularly for data such as that of a fire alarm in which changes occur rarely, but careful consideration is necessary as there's also a semantic security concern. When a fire occurs, the fire alarm message will be sent and, even if it's encrypted, will tell an eavesdropper that there's a fire. Keeping this type of event private while ensuring efficiency and low latency is a challenge, and thus is a direction for exploration in the context of IoT security.

Similar to the idea of reusing the physical layer for authentication, we might consider using the physical layer to support confidentiality between IoT node Alice and receiver Bob in the presence of eavesdropper Eve. Conventional cryptography exploits an advantage that Alice and Bob have over Eve—namely their shared encryption key—to support secrecy. In many settings, it's possible to find other advantages that Alice and Bob have. For example, when Alice and Bob share a channel that's better than Eve's, they can use proper coding to guarantee they can secretly share information that Eve can't understand.

This concept of secrecy has been around for some time,[16] and quite recently, these ideas have been rejuvenated in the context of secret wireless communications. Generally, confidential communications can be supported when the Alice–Bob communication channel is somehow better than the Alice–Eve channel; recent results have exploited the fading process in a typical wireless scenario to harm an adversary's ability to eavesdrop.[14] These methods are promising for supporting confidentiality in communications because they're built directly from a step that an IoT device must perform anyway—namely, communication.

However, we must carefully consider reusing the physical layer to support confidentiality; this is an opportunity for further research. Notably, using physical-layer secrecy has its own costs; perhaps the most obvious is the message expansion needed for confidentiality.

Similar to the message expansion problems associated with probabilistic encryption,[17] a source coding approach to physical-layer secrecy typically involves coded cipher text that's larger than the plaintext. Because the message is larger, the transmission is longer, placing more strain on a system's energy. Approaches, such as ciphers or encoding for physical-layer confidentiality, that are efficient and have little to no message expansion—and thereby mitigate the burden of supporting confidentiality for a transmitting IoT node—are promising directions for investigation.

Ultimately, the IoT's future will rely on our ability to adequately secure hard-to-secure, resource-sparse devices. Without this, a vast population of security-disenfranchised devices will be connected to the Internet, creating a wide array of unforeseen security and privacy breaches. ■

References

1. F. Stajano and R. Anderson, "The Resurrecting Duckling: Security Issues for Ad-Hoc Wireless Networks," *Proc. 7th Int'l Workshop Security Protocols*, 2000, pp. 172–194.
2. A. Perrig et al., "SPINS: Security Protocols for Sensor Networks," *Wireless Networks*, vol. 8, no. 5, 2002, pp. 521–534.
3. B. Firner et al., "Towards Continuous Asset Tracking: Low-Power Communication and Fail-Safe Presence Assurance," *Proc. 6th Ann. IEEE Conf. Sensor, Mesh and Ad Hoc Communications and Networks*, 2009, pp. 1–9.
4. I. Rouf et al., "Security and Privacy Vulnerabilities of In-Car Wireless Networks: A Tire Pressure Monitoring System Case Study," *Proc. 19th Usenix Conf. Security*, 2010, pp. 21–21.
5. R. Van Noorden, "The Rechargeable Revolution: A Better Battery," *Nature*, vol. 507, no. 7490, 2014, pp. 26–28.
6. "Questions and Answers about Biological Effects and Potential Hazards of Radiofrequency Electromagnetic Fields," Office of Eng. and Technology, Federal Communications Commission, Aug. 1999; http://transition.fcc.gov/Bureaus/Engineering_Technology/Documents/bulletins/oet56/oet56e4.pdf.
7. M. Gorlatova et al., "Challenge: Ultra-Low-Power Energy-Harvesting Active Networked Tags (EnHANTs)," *Proc. 15th Ann. Int'l Conf. Mobile Computing and Networking*, 2009, pp. 253–260.
8. "MSP430g2553," Texas Instruments, 2014; www.ti.com/product/msp430g2553.
9. W. Jung et al., "SSL-Based Lightweight Security of IP-Based Wireless Sensor Networks," *Proc. Int'l Conf. Advanced Information Networking and Applications*, 2009, pp. 1112–1117.
10. H. Chae et al., "Maximalist Cryptography and Computation on the WISP UHF RFID Tag," *Wirelessly Powered Sensor Networks and Computational RFID*, Springer, 2013, pp. 175–187.

11. S. Karthikeyan and M. Nesterenko, "RFID Security without Extensive Cryptography," *Proc. 3rd ACM Workshop Security of Ad Hoc and Sensor Networks*, 2005, pp. 63–67.
12. E. Vahedi, R. Ward, and I. Blake, "Security Analysis and Complexity Comparison of Some Recent Lightweight RFID Protocols," *Computational Intelligence in Security for Information Systems*, LNCS 6694, Springer, 2011, pp. 92–99.
13. J. Kaps and B. Sunar, "Energy Comparison of AES and SHA-1 for Ubiquitous Computing," *Proc. Int'l Conf. Emerging Directions in Embedded and Ubiquitous Computing*, 2006, pp. 372–381.
14. R. Liu and W. Trappe, *Securing Wireless Communications at the Physical Layer*, Springer, 2010.
15. L. Xiao et al., "Using the Physical Layer for Wireless Authentication in Time-Variant Channels," *IEEE Trans. Wireless Communications*, vol. 7, no. 7, 2008, pp. 2571–2579.
16. A.D. Wyner, "The Wire-Tap Channel," *Bell Systems Tech. J.*, vol. 54, no. 8, 1975, pp. 1355–1387.
17. S. Goldwasser and S. Micali, "Probabilistic Encryption," *J. Computer and System Sciences*, vol. 28, no. 2, 1984, pp. 270–299.

Wade Trappe is an associate director at the Wireless Information Network Laboratory and a professor in the Department of Electrical and Computer Engineering at Rutgers University. His research interests include wireless security, wireless networking, and networking security. Trappe received a PhD in applied mathematics and scientific computing from the University of Maryland. He's a coauthor of *Introduction to Cryptography with Coding Theory* and an IEEE Fellow. Contact him at trappe@winlab.rutgers.edu.

Richard Howard is a research professor at the Wireless Information Network Laboratory at Rutgers University. His research interests include ultra-low-power wireless systems and sensors. Howard received a PhD in applied physics from Stanford University. He's a Fellow of the American Physical Society and the American Association for the Advancement of Science as well as a senior member of IEEE. Contact him at reh@winlab.rutgers.edu.

Robert S. Moore is a researcher in the Computer Science Department at Rutgers University. His research interests include computer networking, low-energy wireless sensors, and passive and active indoor localization. Moore received a PhD in computer science from Rutgers University. Contact him at romoore@cs.rutgers.edu.

Selected CS articles and columns are also available for free at http://ComputingNow.computer.org.