Hardware and Embedded Security in the Context of Internet of Things

Arun Kanuparthi
Polytechnic Institute of NYU
Brooklyn, NY - 11201. USA
arun.kanuparthi@nyu.edu

Ramesh Karri
Polytechnic Institute of NYU
Brooklyn, NY - 11201. USA
rkarri@poly.edu

Sateesh Addepalli
Cisco Systems
San Jose, CA - 95134. USA
sateeshk@cisco.com

ABSTRACT

Internet of Things (IoT) is the interconnection of a large number of resource-constrained devices such as sensors, actuators, and nodes that generate large volumes of data which are then processed into useful actions in areas such as home and building automation, intelligent transportation and connected vehicles, industrial automation, smart healthcare, smart cities, and others. Important challenges remain to fulfill the IoT vision including *data provenance and integrity, trust management, identity management*, and *privacy*. We describe how embedded and hardware security approaches can be the basis to address these security challenges.

Categories and Subject Descriptors

B.4 [**Hardware**]: Input/output data communications; C.3 [**Computer Systems Organization**]: Special-purpose and application-based systems

Keywords

Internet of Things, Security Architecture, Secure IoT

1. INTRODUCTION

The way our society interacts with technology is rapidly heading towards a major paradigm shift. Computing is becoming centered on the vast amounts of data and information captured and made accessible as all humans and devices get connected into an Internet of Things (IoT) [8, 1]. IoT is an interconnection of a large number of networked devices. The interaction between smart machines and the environment results in the generation of large volumes of data that may be processed into useful commands to control actuators. IoT will encompass medical implants, alarm clocks, wearable systems, automobiles, washing machines, traffic lights, and the energy grid. It is expected that 50 billion devices will be interconnected by 2020, and this number is further expected to reach a trillion [9].

CyCAR'13, November 4, 2013, Berlin, Germany.

http://dx.doi.org/10.1145/2517968.2517976.

It will allow for new applications that tackle societal challenges by using unprecedented access to data. For instance, vehicular collisions, which kill thirty thousand people in the US annually and injure almost a million more, may be tackled by using embedded wireless sensors, monitors, and actuators in automobiles. IoT will make it possible for emergency workers to increase their effectiveness during disaster response by connecting to networks of robots. IoT is anticipated to play a critical role in future megacities that are instrumented with a myriad of sensors.

Security and privacy are key challenges to make the IoT a reality. They cannot be dealt with in an ad-hoc manner using reactive approaches. A proactive approach is required, where trustworthiness is engineered upfront into IoT. IoT must have strong security foundations built on a holistic view of security for all IoT components. Measures to address the realistic challenges of *data provenance and integrity, identity management, trust management*, and *privacy* must be implemented. Absent strong security foundations, attacks on and malfunctions in the IoT components will outweigh any of its benefits.

Data provenance and integrity, identity management, trust management, and privacy are four key challenges in designing a secure IoT. Data provenance ensures that the source of data is trustworthy. Data integrity ensures that the data has not been maliciously tampered with. Trust management ensures trust in the devices. Identity management refers to the administration of individual identities. Privacy is essential to ensure that the user's data and credentials are under his control and no one else's. Embedded and hardware security approaches can be leveraged to build a secure IoT. We focus on securing the resource-constrained embedded devices (the sensors that collect the information, the nodes that process this information, and the actuators that perform the physical action). First we propose to integrate sensing with PUF technology [13] for data provenance and integrity. Second, we propose to use PUFs for identity management. Third, we propose to use hardware performance counters [17] for trust management and to monitor the integrity of applications. Finally, we propose to use lightweight cryptography to provide privacy.

The rest of the paper is outlined as follows. A generic IoT architecture and its threat model are described in Section 2. The challenges involved in designing a secure IoT are described in Section 3. We also describe how embedded and hardware security approaches can be used to address these challenges. We conclude the paper in Section 4.

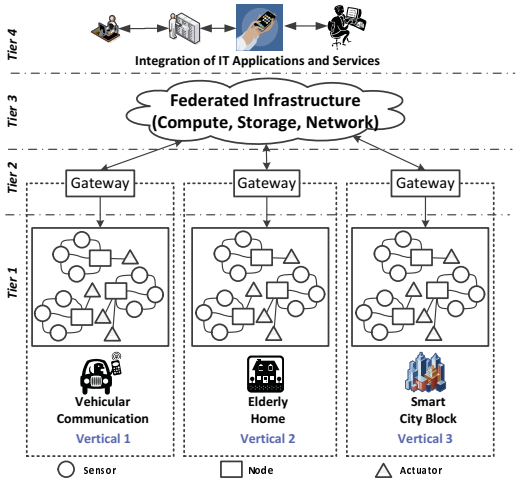

Figure 1: An IoT architecture with three verticals of vehicular communication, elderly home, and smart city block. The four tiers in an IoT architecture are sensors network (consisting of sensors, nodes, and actuators), gateway, and federated infrastructure (with compute, network, and storage capabilities), and integration of IT applications and services.

2. AN IOT ARCHITECTURE

A typical IoT architecture performs: (i) sensing and data collection (using sensors), (ii) local embedded processing (at the node and gateway), (iii) activating devices based on commands sent from the nodes (using actuators), (iv) wired and/or wireless communication (using low power wireless protocols), (v) automation (using software), and (vi) remote processing (federated compute-network-storage infrastructure). The IoT ecosystem is divided into four tiers to accomplish the above mentioned tasks [5]. Figure 1 shows one IoT architecture that includes use cases of vehicular communication, elderly home, and smart grid.

The first tier consists of sensors, actuators, and processing nodes. *Sensors* collect data. They typically have very low processing capability. *Processing nodes* process the data collected by the sensor network to take necessary action. The node has limited storage, low processing capability and power budget. The second tier consists of the gateway. *Gateway* interfaces tier 1 to the outside world via the internet. It has good processing power and memory. Most state-of-the-art gateways also provide wireless communication [10][1]. In some cases gateway can be subsume the functionality of a node. The federated infrastructure that can belong to Enterprise domain or Service provider domain has compute, network, and storage capabilities. This forms the third tier. In addition to performing various aggregation, management and service delivery functions it is capable of processing strong cryptographic algorithms that consume a lot of power. IT applications and services are integrated in the fourth tier.

2.1 Vertical 1: Vehicular Communication

Vehicular communication offers a rich variety of connectivity and interactions: cars to cars, cars to access points (Wi-Fi, 4G, LTE, and Smart Traffic Lights (STLs)), and access points to access points to deliver a rich menu of services such as saftey, traffic support, mobility and location awareness, and support for real-time interactions. For instance, a vehicle that is in the blind spot of another vehicle can sense a collision and communicate the alert the driver to apply the brake. An STL may interact with other sensors to detect pedestrians, bikers, and measure the speed of approaching vehicles. It may also interact with neighboring STLs to coordinate the green traffic wave.

2.2 Vertical 2: Elderly Home

The instrumented home (of a Grandma) as it may evolve in the next 10 years will be a controllable and programmable platform. An elderly home may be instrumented with technologies such as pill bottles to ensure that medicines are taken at the right time and the right dosage is administered, wearable devices to track the gait to detect falls and to monitor balance issues and sensors that track food stored in the kitchen. They may be supported by apps to recommend recipes based upon best-by dates and dietary recommendations, and to automatically request delivery to replenish food [16, 11]. However, without addressing security and privacy issues, such systems have not found traction.

2.3 Vertical 3: Smart City Block

Co-optimization of water, electricity, temperature control, and noise at the city block level is an example. The flow of people in the city block (for example, around large buildings) can be monitored to optimize foot traffic on one hand and to schedule street cleaning to minimize disruption on the other hand. Within a building, elevator patterns can be monitored and adapted to conserve energy and/or reduce wait time based on the time of the day (peak versus off-peak). At the individual worker level, one can monitor the individual life style and usage patterns to program his/her computing and communication devices to variable power input and output modes to balance her needs in the context of critical environmental factors. An app can use the customer calendar (meetings, lunch, desk time and gym time) to adjust the ambient environment parameters. Flow of emergency responders can be optimized to ensure their priority access to transit in emergencies.

2.4 Possible Threats in IoT

IoT's distributed nature and use of resource-constrained embedded devices in public areas make them easily exploitable. Easily accessible sensors and actuators in unprotected zones, such as city streets, are vulnerable to physical damage.

Figure 2 shows the threat model of the IoT architecture. Sensors can be tampered with to provide incorrect data to the nodes, while the actuators may be sent commands from unauthorized sources to perform some physical action. For instance, a malicious temperature sensor always reports a fixed value, a tampered security camera may always replay outdated video streams. Authentication failure at the sensor may give an attacker unauthorized access to private/confidential information. For instance, a faulty home security sensor may not trigger an alarm and lets a burglar access into the building. At the node level, the application running on the microcontroller may be compromised and may leak encryption keys, etc. Denial of service attacks may be launched at the gateway, thereby preventing information to be transmitted or received through the internet. A secure IoT architecture must ensure end-to-end security.

[1]In some cases, the node may also be the gateway, capable of directly communicating with the federated infrastructure. This is not shown in Figure 1.

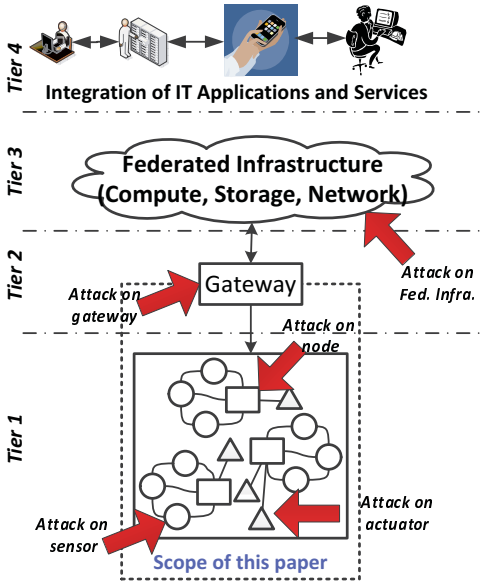

Figure 2: Possible threats in IoT. Attacks can be launched on the sensors, nodes, gateway, and cloud.

3. KEY CHALLENGES

Although IoT is potentially transformative, with a 14 trillion dollar projected market [8], there has been little progress towards its vision beyond limited deployments in certain verticals. The main reason for this lack of progress stems from serious concerns about the security, privacy, and trustworthiness of such systems [12]. IoT elements monitor almost every aspect of a person's life. Hence, citizens have legitimate privacy concerns. Moreover, companies fear reputational damage from data getting into the wrong hands and governments worry about security risks. Security in the IoT has been studied in the literature [12, 3, 7]. These studies focus on the security at remote processing locations. They propose using lightweight cryptographic primitives in the resource-constrained embedded devices.

The design of a secure IoT architecture involves addressing the challenges of *data provenance and integrity, identity management, trust management*, and *privacy*. We outline these challenges and explain how embedded and hardware security support can address these challenges.

3.1 Data Provenance and Integrity

Trust in data is trust in the system. Ensuring the trustworthiness of data coming from IoT to applications that analyze that data and potentially actuate controls based on this data requires that trust be addressed at both the producer and the consumer side of this data [18]. The main questions here are: *How can the data coming from the sensor be trusted?*, and *How can we ensure that the integrity of the data has not been compromised?*.

AttackExample 1: A sensor is maliciously modified to report incorrect values. For instance, a temperature sensor may be tampered to always report a certain value irrespective of the actual temperature. This problem can be addressed at hardware level using sensor PUFs [13].

A traditional PUF [15] takes in a challenge and ideally produces a response with the following properties:

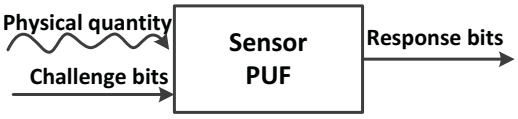

Figure 3: Traditional PUF produces the response based on the challenge. Sensor PUF produces the response based on the challenge as well as the sensed physical quantity.

- For a given binary challenge, a PUF always produces the same response.
- One challenge-response pair leaks nothing about other pairs.
- The manufacturer of the PUF cannot predetermine the mapping.

Sensor PUF is a Physical Unclonable Function (PUF) that co-mingles sensing with the challenge response processing of a PUF. A sensor PUF extends the functionality of conventional physical unclonable functions to provide authentication, unclonability, and verification of a sensed value. The variation that a sensor PUF (shown in Figure 3) provides, extends conventional PUFs by including two inputs: a physical quantity and a traditional binary challenge. A sensor PUF has the following properties:

- For a given challenge and a given sensed quantity, the sensor PUF always produces the same response.
- One challenge-quantity-response triple leaks nothing about other triples.
- The manufacturer of the sensor PUF cannot predetermine the challenge-quantity-response mapping.

This new class of sensors addresses the vulnerability in typical sensing systems, in which an attacker can spoof measurements by interfering with the analog signals that pass from the sensor element to the embedded microprocessor. By merging sensing with cryptography, sensor PUF provides assurances about data integrity and forms the basis of data provenance and integrity.

Example 1- Defense: For a given challenge and a given sensed quantity, the sensor PUF always produces the same response. This attack can be thwarted if the node can issue different challenges to the sensor. This produces a different responses (depending on the sensed quantity), those the tampered sensor cannot generate. Thereby, providing assurance of the sensed value.

3.2 Challenge 2: Identity Management

Identity management refers to the administration of individual identities within a system. Without unique, unforgeable, and easily verifiable identities, there is no accountability or deterrence. An identification system for IoT should scale to trillions of nodes. Not every IoT identity should be directly accessible by external entities unless they are authenticated [14]. The question here is: *Is a sensor authorized to send data to the node?*

Attack Example 2- Device A fakes the identity of Device B In this scenario, the device could be a sensor, node, or an actuator. For instance, a malicious node fakes it's identity as that of a genuine node and sends malicious commands to the actuator to perform some actions.

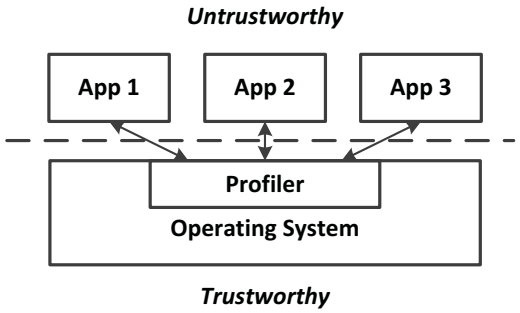

Figure 4: Integrity checking using HPCs [17]

Sensor PUFs can provide unique IDs. Therefore, by exploiting the fact that the PUF can have exponential number of challenge response pairs, where the response is unique for each IC and each challenge, the threat of fake identities can be neutralized by application of randomly chosen challenge-response pairs [15].

Identity management at the higher levels (i) maintains a repository of legitimate users, (ii) adds, modifies, and deletes the contents of the repository, (iii) regulates user access, enforces security policies and access privileges, (iv) reports system activities and audits to verify past activities.

The nodes have some processing power and are capable of running applications on them to process the data. A lightweight identity management application that performs the above mentioned tasks can be installed on the node. This application collects data only from authorized sources (sensors). The node must maintain a list of sensors which are authorized to send data. It also facilitates the addition of new sources and removal of retired sources. Some amount of storage can be dedicated to log activities, or this data can be transmitted to tier-3 for storage.

Example 2 - Defense: Since each device is augmented with a PUF, it has its own unique identity. By using different challenge-response pairs, the true identity of the device can be found. Thereby, differentiating a genuine device from a fake device.

3.3 Challenge 3: Trust Management

The distributed nature of the IoT and the strong human component makes the creation of appropriate trust models and trust managements systems challenging [4]. The question here is: *Can the device that is transmitting the data be trusted?* A root of trust is necessary to build a chain of trust and ensure trustworthiness. The root of trust begins at the hardware level [6].

Much of the power of IoT, comes from applications that process the data that is sent by the sensors and from the actuators that exert physical action. These applications must be trustworthy and must be protected against attackers trying to exploit the vulnerabilities such as unchecked buffers in these applications [2]. Also, the actuators must be protected from bogus inputs from unauthorized sources.

Attack Example 3- Tampering the application on the node: An attacker may attempt to exploit vulnerabilities in the applications running on the node, which process the data collected from the sensors and send commands to the actuators. Alternatively, an attacker may attempt to surreptitiously execute a rootkit.

Legacy systems have all the infrastructure and software in place and making radical changes to the hardware is usually not allowed. In order to provide trustworthiness in legacy as well as low-cost systems, one can leverage hardware performance counters (HPCs) that are present in all commodity processors. HPCs are registers that can monitor certain events that occur during the lifetime of a program.

Table 1: NumChecker detection capabilities. The numbers are deviations (%) from uninfected executions. Deviation of more than 5% suggests a malicious modification. For each rootkit, the bold number indicates the largest deviation [17].

Rootkit	Events counted	System calls monitored	
		sys_open	sys_getde -nts64
SucKIT 1.3b	INST	836.1	242.9
	RN	676.5	483.3
	BR	**1294.2**	1028.1
Adore 0.42	INST	99.4	427.7
	RN	123.5	650.0
	BR	119.9	**1313.1**
Sk2rc2	INST	363.4	39.8
	RN	**488.2**	95.8
	BR	359.2	66.9
Superkit	INST	827.8	244.4
	RN	535.3	483.3
	BR	**1399.5**	1014.4

The counters facilitate monitoring of the programs [17]. As seen in Figure 4, when a program begins to run, the counters are activated by the Operating System. Depending on the model, the events can be counted periodically or at the end of program execution. Using a mix of different events we can generate a model that is program and platform dependent. This model is used to monitor the software integrity. One downside of using HPCs is that the approach is not very accurate and may produce false positives.

In order to introduce new functionality, rootkits usually modify the original system calls. The difference in the number of events between normal and infected executions is notable. This abnormality helps in detecting rootkits. Table 1 shows how various rootkits modified the original system calls in a Linux 2.4 kernel were detected.

Defense Example 3: Tampering program execution introduces significant deviation and can be detected using HPCs.

3.4 Challenge 4: Privacy

The foundation of IoT applications is sensitive data provided by users and their devices. Privacy enhancing technologies can protect users sensitive data while still preserving the functionality of higher-level applications.

Attack Example 4- Protection against eavesdropping attack: An adversary attempts to eavesdrop on the communication between the devices 6and retrieve private information.

Confidentiality of data can be ensured by using lightweight encryption algorithms. Implementing policies that require approval from the user to participate in the IoT can alleviate the privacy concerns of users. For instance, sensors send push notifications to users before collecting their private data.

Example Defense 4: All communication from the sensor to the node, or from the node to the actuator is encrypted using lightweight encryption algorithms. This communication is confidential to an adversary attempting to eavesdrop on this communication. This way, privacy is ensured.

Table 2: Summary of security challenges in IoT and corresponding hardware/embedded security support.

Challenges	Hardware/Embedded Security Support
Data Provenance and Integrity	Sensor PUF
Identity Management	Sensor PUF, PUF
Trust Management	PUF, HPCs
Privacy	Lightweight encryption

Table 3: Recommendations on lightweight cryptographic primitives to be used at each tier of IoT

	Sensor	Node	Gateway	Fed. Infr.
Data size	< 10 B	< 1 MB	< 1 GB	1 GB
Enc/Dec	PRESENT mCRYPTON	CLEFIA AES	AES ECC	RSA
Hash	DM-PRESENT	PROP	HMAC	SHA-3
Key Ex.	DH-512	DH-512	ECDH	DH
Digital Sign.	ECDSA-163	ECDSA, -233	DSA	ECDSA 409

In short, sensor PUFs address the challenge of data provenance and integrity. Sensor PUFs and PUFs can be used for identity management; PUFs and hardware performance counters can be used for trust management. Lightweight encryption algorithms can support confidentiality and privacy to users. Table 2 summarizes the challenges and hardware/embedded security solutions.

3.5 Other Security Requirements

In addition to the challenges mentioned in Section 3, a A secure architecture must support confidentiality, integrity, availability, authenticity, and non-repudiation; the IoT is no different. These are accomplished using cryptographic primitives such as encryption algorithms, hash functions, digital signatures, and key exchange algorithms.

It is crucial to choose the appropriate cryptographic algorithm that does not consume too much power. For instance, if the amount of data to be processed is less than 1 KB, the processing can be done on the sensor itself, else it can be sent to the node for processing. The node is capable of processing data under 1 MB. The gateway and federated infrastructure can process data upto 1 GB and greater than 1 GB, respectively. By doing localized processing, data processing in tier 3 can be avoided. This localized processing results in faster response times. Table 3 shows the cryptographic primitives that can be used at each tier of the IoT.

4. CONCLUSION

We identified four key challenges in designing a secure IoT: data management, identity management, trust management, and privacy. We describe how embedded and hardware security approaches can be used to address these challenges in the context of an IoT. We propose the use of Sensor PUFs to address the challenge of data provenance and integrity. Sensor PUFs and PUFs can be used for identity management; PUFs and hardware performance counters can be used for trust management. Lightweight encryption algorithms can be used to provide confidentiality and privacy to the users.

5. REFERENCES

[1] Internet of Things - Architecture. www.iot-a.eu/public, 2013.

[2] Aleph One. Smashing the stack for fun and profit. *Phrack magazine*, 7(49):365, 1996.

[3] S. Babar, A. Stango, N. Prasad, J. Sen, and R. Prasad. Proposed embedded security framework for internet of things (iot). In *Wireless Communication, Vehicular Technology, Information Theory and Aerospace Electronic Systems Technology (Wireless VITAE), 2011 2nd International Conference on*, pages 1–5, 2011.

[4] M. Blaze, J. Feigenbaum, and J. Lacy. Decentralized trust management. In *Security and Privacy, 1996. Proceedings., 1996 IEEE Symposium on*, pages 164–173. IEEE, 1996.

[5] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli. Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pages 13–16. ACM, 2012.

[6] D. Champagne and R. B. Lee. Scalable architectural support for trusted software. In *High Performance Computer Architecture (HPCA), 2010 IEEE 16th International Symposium on*, pages 1–12. IEEE, 2010.

[7] D. Chen, G. Chang, L. Jin, X. Ren, J. Li, and F. Li. A novel secure architecture for the internet of things. In *Genetic and Evolutionary Computing (ICGEC), 2011 Fifth International Conference on*, pages 311–314, 2011.

[8] M. Chui, M. LÃűffler, and R. Roberts. The Internet of Things. *McKinsey and Co. Quarterly Journal*, 2010.

[9] Cisco. The Internet of Things - How the Next Evolution of the Internet is Changing Everything, 2011.

[10] Cisco Systems. Cisco 819 4G LTE M2M Gateway Integrated Service Router.

[11] J. Heitzeberg. Lively: Smart Sensors for Elderly Loved Ones, 2013.

[12] R. Roman, P. Najera, and J. Lopez. Securing the internet of things. *Computer*, 44(9):51–58, 2011.

[13] K. Rosenfeld, E. Gavas, and R. Karri. Sensor physical unclonable functions. In *Hardware-Oriented Security and Trust (HOST), 2010 IEEE International Symposium on*, pages 112–117. IEEE, 2010.

[14] J. S. Shapiro, J. M. Smith, and D. J. Farber. EROS: A Capability System. 1999.

[15] G. E. Suh and S. Devadas. Physical unclonable functions for device authentication and secret key generation. In *Proceedings of the 44th annual Design Automation Conference*, pages 9–14. ACM, 2007.

[16] The Economist. Care for the elderly: An age old problem, 2011.

[17] X. Wang and R. Karri. Numchecker: Detecting kernel control-flow modifying rootkits by using hardware performance counters. In *Design Automation Conference (DAC), 2013 50th ACM / EDAC / IEEE*, pages 1–7, 2013.

[18] K. Xu, H. Xiong, C. Wu, D. Stefan, and D. Yao. Data-provenance verification for secure hosts. *Dependable and Secure Computing, IEEE Transactions on*, 9(2):173–183, 2012.