Protecting Endpoint Devices in IoT Supply Chain

(Invited Paper)

Kun Yang, Domenic Forte, and Mark M. Tehranipoor

ECE Department, University of Florida
{k.yang}@ufl.edu, {dforte, tehranipoor}@ece.ufl.edu

***Abstract*—The Internet of Things (IoT), an emerging global network of uniquely identifiable embedded computing devices within the existing Internet infrastructure, is transforming how we live and work by increasing the connectedness of people and things on a scale that was once unimaginable. In addition to increased communication efficiency between connected objects, the IoT also brings new security and privacy challenges. Comprehensive measures that enable IoT device authentication and secure access control need to be established. Existing hardware, software, and network protection methods, however, are designed against fraction of real security issues and lack the capability to trace the provenance and history information of IoT devices. To mitigate this shortcoming, we propose an RFID-enabled solution that aims at protecting endpoint devices in IoT supply chain. We take advantage of the connection between RFID tag and control chip in an IoT device to enable data transfer from tag memory to centralized database for authentication once deployed. Finally, we evaluate the security of our proposed scheme against various attacks.**

***Index Terms*—Internet of Things (IoT), Endpoint Device, Supply Chain Security, Traceability, Authentication**

I. Introduction

Ever since its appearance in 1982, when a modified Coke machine at Carnegie Mellon University becomes the first internet-connected appliance [1], the concept of the Internet of Things (IoT) has attracted more and more attention over the past few decades. The IoT corresponds to the interconnection of uniquely identifiable embedded computing devices within the existing Internet infrastructure. Facilitating information and service exchange between connected objects in global supply chain networks is one of the major missions of IoT. With IoT solutions, we are able to be aware of asset status from virtually anywhere, accurately position system failure, and dynamically capture, communicate, and analyze intelligence data. The pace of IoT adoption is accelerating because of rapid development of cloud computing, increased number of smart devices, and proliferated applications connecting supply chain owners, partners, and customers. Cisco's Internet Business Solutions Group (IBSG) predicts there will be 50 billion devices connected to the Internet by 2020 [2].

In addition to all the above-mentioned benefits, the IoT also raises new security and privacy challenges. While current Internet is a connection of rather uniform devices, the IoT will exhibit much *higher level of heterogeneity*, as objects of different functionality, technology and application fields will belong to the same communication environment. Furthermore, IoT devices are usually *resource constrained* in terms of computing, communication, and storage capabilities. As a result, security and privacy maintenance for IoT will be much harder compared with Internet and the more conventional embedded systems.

Connected IoT devices may be accessed or controlled by malicious network nodes. The authors in [3] demonstrated how it is possible for an external party to gain control over every connected device within a ZigBee network by taking advantage of security flaws in ZigBee standard, one of the most popular wireless communication standards used by IoT devices. Attackers may compromise IoT devices and build a botnet to launch cyber attacks, including sending spam, spreading viruses and worms, and running denial-of-service attacks. In 2014, Proofpoint uncovered the first proven IoT-based cyber attack, which involved more than 750,000 phishing and spam emails launched from more than 100,000 compromised IoT devices, including home-networking routers, connected multimedia centers, televisions and at least one refrigerator [4]. IoT devices infected with malware may disclose sensitive data (e.g., contactless payment information) to adversaries. Communication between IoT devices or between IoT device and trust center (i.e., the core device who is responsible for joining activities within a local area network) may suffer from eavesdropping. Data captured by sensors connected to IoT devices may be altered maliciously during communication. Credentials of IoT devices may be stolen by hackers to perform further cyber attacks [5]. Credentials assigned to one IoT device may be replayed by another device after its lifetime. On the other hand, user privacy of IoT devices is also at risk especially when current smart devices could collect much private information such as blood pressure and heart rate.

A lot of solutions have been proposed to defend IoT devices against cyber threats. By optimizing communication standards, improving device security configuration, upgrading firmware, setting strong passwords, installing patches, etc., the vast majority of cyber attacks can be prevented. Symmetric encryption algorithms such as Advanced Encryption Standard (AES) [6] are widely adopted to prevent eavesdropping on the communication between two resource constrained network devices. The keyed-hash message authentication code (HMAC) [7] can be used to simultaneously verify both the data integrity and the authentication of a message. To prevent credentials of one dead IoT device from being reused by another device, credentials need to be tied to lifetime. To preserve user privacy in a participatory sensing network, a Hot-Potato-Privacy-Protection algorithm (HP^3), in which data is delivered to the next hop until some user-defined threshold is reached before being uploaded to the server, has been proposed in [8]. To limit access to information content, an access control technique implemented within a network device was presented in the patent [9]. It determines whether to forward client requests for processing by comparing client source information against a database of Uniform Resource Locators (URLs), IP addresses, or other resource identification data.

However, hardware threats are rarely touched which are also critical to IoT security. The IoT devices may contain untrusted components. For example, counterfeit integrated circuits (ICs) (e.g., recycled or remarked ICs) or ICs containing hardware Trojans may have been mounted on the PCBs of IoT devices intentionally or unintentionally by the system integrators before they enter the supply chain. Authentic IoT devices may be mixed with clones or fakes during their distribution by untrusted supply chain partners. IoT devices may be lost or stolen during distribution or even after deployment. IoT devices may even be physically tampered by rogue elements who may have access to them after being provisioned.

To detect hardware Trojans contained in the ICs, a series of techniques have been developed using side-channel signal analysis, functional test, etc [10]. To combat die and IC recycling (CDIR), the authors in [11] proposed a suite of solutions including light-weight, on-chip structures based on ring oscillators (RO-CDIR), anti-fuses (AF-CDIR) and fuses (F-CDIR). To eliminate counterfeit ICs from the electronics supply chain, the Defense Advanced Research Projects Agency (DARPA) launched the Supply Chain Hardware Integrity for Electronics Defense (SHIELD) program to develop a hardware root of trust, called dielet, to enable IC authentication [12]. The dielet in essence is a microscopic-scale chip that combines strong encryption, sensors, near-field power and communications. The dielet can be inserted into the package of an IC to capture any tampering attempt and communicate with the server via the RF channel. To prevent theft of an item from a building, a device which can activate the alarm when taken through the exit was patented in [13]. The patented device is extremely small and can be associated with a vast majority of products. To protect the secrecy of internal states of cryptographic hardware against an adversary who may modify the values of an unbounded number of wires anywhere in the circuit, an efficient transformation of a circuit realizing an arbitrary functionality into a private circuit realizing the same functionality has

978-1-4673-8388-2/15/$31.00 ©2015 IEEE

been proposed in [14].

However, all the above measures are directed against only one type of risk associated with the IoT. Some approaches depend on extra circuitry to be added to chip design, which is not always available especially when legacy chips are used. Also, none of them treat the supply chain of IoT devices as a whole (i.e., from a starting point to the end points) when considering security and privacy issues. In this case, it becomes particularly important to develop a low-cost full-fledged solution that ensures the security of IoT even under complex global supply chains and given limited available hardware resources.

In this work, we present an RFID-enabled solution that aims at protecting endpoint devices in IoT supply chain. Central to our solution is an RFID-enabled Supply Chain management and traceability scheme called ReSC-2 (the original ReSC approach was presented in [15], where ReSC stands for RFID-enabled Supply Chain). Compared with the original ReSC approach, ReSC-2 has the following features and advantages: (i) ReSC-2 is specific to the supply chain of IoT devices; (ii) ReSC-2 enables mutual authentication between RFID readers and tags; (iii) ReSC-2 does not require RFID readers to upload backups of tag traces to the centralized database. Our main contributions are as follows:

- IoT security and privacy challenges are systematically analyzed.
- ReSC-2 enables traceability and authentication of IoT devices across the entire supply chain. Its authentication procedure consists of two steps: (i) verifying the matching between RFID tag and IoT device; (ii) validating tag trace.

The remainder of this paper is organized as follows: Section II introduces the related work. Section III discusses the security and privacy challenges associated with the IoTs. Section IV describes ReSC-2 in details and how it can address many of the IoT security challenges. In Section V, we evaluate the security of ReSC-2. Finally, we conclude in Section VI.

II. Related Work

Two areas of prior work have particular relevance to our study: hardware security primitives and RFID-enabled supply chain management.

Hardware Security Primitives: Resource constrained embedded devices (e.g., RFID tags and wireless sensor nodes) are subject to physical and side-channel attacks due to the lack of standard security-enhanced hardware. In this context, lightweight and cost-effective hardware security primitives are considered to be integrated into cryptographic schemes and security protocols to build trust in remote embedded devices by assuring those devices are functioning in the predefined states. Typical hardware security primitives include physical unclonable functions (PUFs) and true random number generators (TRNGs). When issued a challenge, an ideal PUF produces a unique and reliable response that depends on uncontrollable process variations during IC manufacturing [16]. Among different types of PUFs, SRAM PUF becomes popular due to its convenience of using commonly available and integrated SRAM rather than include a dedicated primitive in the circuit [17]. TRNGs produce a sequence of random and uniformly distributed symbols by measuring a random physical phenomenon, such as thermal noise or other quantifiable electromagnetic and quantum phenomena [18]. TRNGs are widely used for confidentiality (e.g., one-time pads, session keys, seeds, initial vectors, etc.), integrity (e.g., nonce generation), and authenticity (e.g., challenges for authentication).

RFID-enabled Supply Chain Management: RFID technologies have been widely explored to enhance visibility and enable traceability in the supply chain over the past decade. The authors in [19] proposed to detect cloning attacks by verifying the correct sequence of tag observations related to transport processes, which does not rely on global knowledge of supply chain structures or product flows, and thus is robust to supply chain dynamics, recalls, and misdeliveries. A tailing mechanism was proposed in [20] to detect cloning attacks by writing random numbers to tags as they pass through the supply chain and verifying tail (composed of random numbers) divergence between genuine and cloned tags over time. However, all the above measures have the following limitations: (i) the tags lack inherent connection to the objects they are attached to and thus are vulnerable to split attacks (i.e., separating tag from product, swapping tags, etc.); (ii) the tag trace has no necessary relation to the tag itself and thus is vulnerable to duplication attack (i.e., duplicating tag trace); (iii) readers have to be connected to the centralized database to perform rule verification and clone detection; and (iv) they cannot prevent counterfeit or stolen products from being used by end-users.

By combining the important features of both set of techniques (hardware security primitives and RFID technologies), we propose an RFID-enabled solution that aims at efficiently protecting endpoint devices in IoT supply chain.

III. IoT Security and Privacy Challenges

In this section, we first introduce the properties of IoT and then discuss its security and privacy challenges.

A. Properties of IoT

Different from Internet, the IoT has its own unique properties that increase difficulty to security and privacy maintenance. The properties of IoT are summarized as follows:

Heterogeneity: The IoT exhibits much higher level of heterogeneity than Internet, as objects with totally different functionality and originated from various technology and application fields will belong to the same communication environment. IoT device types range from small RFID tags with limited processing power to large connected servers running sophisticated operating systems. Hence, corresponding security and privacy measures should be interface-friendly and compatible with various types of IoT hardware.

Specificity: Vast majority of current IoT devices (e.g., SmartBand) are designed for a particular use and could collect sensitive personal information (e.g., blood pressure, heart rate, living habit, etc.), in which case how to effectively protect user privacy will be a big concern. In addition to consumer electronics, IoT devices are more and more used in industrial and agricultural automation. For example, IP surveillance cameras are widely used to monitor asset status in the inventories. Compromised IoT devices could disclose significant trade secrets.

Resource Constrained: Most IoT devices are low-cost hardware with constrained resources in terms of computing, communication, and storage capabilities, which requires corresponding security and privacy measures to be lightweight and cost-effective. For example, passive RFID tags used to track and trace commodities in the supply chain are usually equipped with simple read/write operations, XORing with random numbers, and cyclic redundancy check (CRC) capabilities. Wireless sensor network (WSN) sensors are usually equipped with low-cost microcontrollers with small bit width.

Wireless: A large number of IoT devices are equipped with wireless communication modules (e.g., WiFi, Bluetooth, ZigBee, etc.) and have the capability to communicate with neighboring devices or network nodes through the air channel. As a result, malicious readers or network nodes could easily intercept those packets being communicated between IoT devices or between IoT device and trust center without being noticed. Here, we refer to the core device (e.g., the core router in the home security network) who is responsible for joining activities within a local area network as the trust center.

Infectivity: Since most of the time IoT devices are connected to the network and usually share the same network key or group key within a seemingly trusted area (e.g., theme parks, music concerts, sports games, etc.), if one device is compromised, the adversary could easily hack its neighboring devices with the deciphered network/group key. For example, IoT devices within the same ZigBee network will encrypt packets using the shared network key after authenticating themselves to the trust center with their link keys [21].

Mobility: Many IoT devices (e.g., smart phones) are mobile and would move together with their users. As a result, their communication neighborhood will be transformed aperiodically. Dynamic communication neighborhood will be a challenge to authentication methods based on fixed IP addresses or interaction with neighboring devices.

Scalability: The number of IoT devices on the earth have been growing exponentially. The management of such a huge number of IoT devices will be a big challenge. Furthermore, the number of IoT device types is also on the rise, which raises a new challenge to device authentication.

B. Security Challenges

The above-mentioned properties raise new challenges to security and privacy maintenance for IoTs. Conventional approaches to hardware, software and network security cannot be simply adopted to resolve the security and privacy issues associated with IoTs. New lightweight and cost-effective solutions specific to global supply chains of IoT devices need to be proposed to clear the way for large-scale deployment of IoT devices in various areas. The security and privacy challenges associated with IoTs are listed as follows:

Component Trust: Against the backdrop of global supply chains, more and more IoT devices are assembled at overseas manufacturing plants. Components on IoT devices are provided by different vendors and pass through many entities on multiple continents before they are installed in their final applications. In this context, counterfeit ICs [22] or ICs containing hardware Trojans [10] may have been mounted on the PCBs of IoT devices intentionally or unintentionally by assemblers before they enter the supply chain. Mechanisms ensuring trust towards components provided by untrusted vendors should be integrated into security protocols. As an example, DARPA launched the SHIELD program to develop a dielet that enables IC authentication by integrating strong encryption, sensors, near-field power and communications into a microscopic-scale chip capable of being inserted into the package of an IC [12].

Device Authentication: Before arriving at end-users, IoT devices usually have to pass through many entities across the global supply chain. Cloned or fake IoT devices may also enter the supply chain and be mixed with the authentic ones. Some customers may purchase cloned or fake IoT devices from grey market knowingly or unknowingly to save money. More than 700 seizures of counterfeit Cisco network hardware and labels with an estimated retail value of more than $143 million were reported by Department of Justice in 2010 [23]. The cloned IoT devices may have additional malicious functionality to collect personal information, spoof the network, etc. Therefore, when plugged into the network, IoT devices and the remote server should authenticate each other before obtaining network services (e.g., downloading necessary firmware updates).

Hardware Theft: IoT devices and sometimes expensive components on them (e.g., central processing units) may be lost or stolen in inventories, during distribution or even after deployment. In 2012, 117 electronic thefts were reported in the US with the average loss of $382,500 per theft incident [24]. In 2014, 1 million dollars worth of expensive central processing units were replaced with cheaper parts before they were stolen from Hewlett-Packard warehouse in Andover [25].

Access Control: We refer to the selective restriction of access to certain hardware or software resources as access control. When plugged into the network, IoT devices may be accessed by malicious network nodes. Legitimate communicating parties may also try to access contents exceeding their access privileges. Access control prevents activities that could jeopardize system security by constraining what a user can do directly, as well as what programs running on behalf of the users are allowed to do [26]. Role-based access control models assign minimum privileges to system components to finish their jobs. In this case, if any component is compromised or its credential is stolen, the intruder will have minimal access to other parts of the system and the impact of security breach will be minimized.

Data Confidentiality: Communications between IoT devices and between IoT device and trust center may suffer from eavesdropping since they usually operate on the air channel and are protected by weak protocols (e.g., ZigBee [21], EPC C1G2 [27], etc.). Communication between gateway and remote server is more secure since it is usually protected by strong protocols (e.g., TLS [28], IPsec [29], etc.).

Data Integrity: Sensor data and authentication information may be maliciously altered in transit for denial-of-service attack. Digital signatures and message authentication codes can be used to protect data integrity.

Service Availability: IoT devices may suffer from denial-of-service and rogue access point attacks. For example, network service will not be available to customers if their routers are infected with computer viruses. The connectivity between IoT devices will accelerate the virus propagation in the victim population. The virus simulation infected 2000 routers in London in a matter of six or seven weeks [30]. Networking equipment containing ICs with hardware Trojans would also malfunction when the hardware Trojans are triggered [10].

Finite Lifetime: IoT devices have finite lifetime. Credentials of one dead IoT device may be reused by another device, which will impact security and cause economic loss to service providers. Therefore, credentials are recommended to be tied to lifetime of IoT devices.

Physical Tampering: IoT devices may be physically tampered by adversaries who have access to them. Debug ports, test pads, visible tracks and pins on PCBs of IoT devices may provide convenience for physical tampering and side-channel attacks. The authors in [31] successfully hacked the Nest Thermostat by bypassing the firmware verification done by the Nest software stack, installing malicious software into the unit, and altering the device behavior (i.e., transforming the thermostat into a beachhead for a remote attacker to allow for introducing rogue services).

User Privacy: Current smart IoT devices may collect more and more sensitive personal information such as blood pressure, heart beat, and living habit. Users of IoT devices may suffer from being profiled. In addition to personal health information, other individual privacy may also suffer from invasion. For example, compromised IP cameras or smart phones may disclose personal information in terms of family life and social interaction. Other than individual privacy, enterprise privacy is also at risk. For example, compromised RFID systems deployed in the supply chain could disclose significant business information such as manufacturing capabilities, pipeline throughput, inventory capacities, sales conditions, etc.

Table I lists all the above-mentioned security and privacy challenges together with corresponding mitigation measures. Those challenges that can be addressed by our proposed solution ReSC-2 with either similar or different alternative approaches are marked with $\surd$. Section 4 will discuss in details how ReSC-2 addresses most of these security challenges.

Table I: IoT security and privacy challenges

Challenge	Regular Mitigation	Covered by ReSC-2
Component Trust	1. Hardware Trojan detection [10]. 2. Die and IC recycling detection (e.g., RO-CDIR, AF-CDIR, F-CDIR [11]).	×
Device Authentication	Authenticate IoT devices based on unique IP addresses or MAC addresses.	$\surd$
Hardware Theft	Alarm mechanisms based on RFID or other RF techniques [13].	$\surd$
Access Control	1. Mandatory or role-based access controls built in OS [32]. 2. Access control mechanisms integrated into IoT hardware [9].	$\surd$
Data Confidentiality	Encrypting data using symmetric encryption algorithms (e.g., SIMON and SPECK ciphers [33]).	$\surd$
Data Integrity	1. Message authentication code. 2. Digital signature.	$\surd$
Service Availability	1. Firewall and anti-virus software. 2. Hardware Trojan detection [10].	$\surd$
Finite Lifetime	Credentials are tied to device lifetime.	$\surd$
Physical Tampering	1. Tamper-proof or tamper-resistant circuitry design [14]. 2. Detecting illegal tampering/replacement of chips on the PCB with the board ID [34].	×
User Privacy	Privacy protection algorithms (e.g., HP^3 [8]).	×

IV. ReSC-2

The proposed RFID-enabled solution that aims at protecting endpoint devices in IoT supply chain is presented in this section. First, we briefly introduce the original ReSC approach. Secondly, we describe supply

chain framework for IoT devices, ReSC-2 hardware architecture, and how ReSC-2 could address most of the security and privacy challenges for IoTs. Finally, we discuss authentication procedure of ReSC-2 in details.

A. Original ReSC Approach

The original ReSC approach is aimed at protecting the supply chain of network devices [15]. At the system integration stage, the one-to-one mapping between RFID tag identity (*tag ID*) and control chip identity (*CC ID*) in the network device will be enrolled into the centralized database for future authentication. During distribution, RFID readers on the tag's distribution path will write their signatures to the tag memory and jointly create a unique tag trace. RFID reader at the current stage authenticates the tag by validating the old signatures in the tag memory. RFID readers need to upload the backups of tag traces to the centralized database when plugged into the network. When the network device is deployed at end-users, the unique tag trace stored in the tag memory will be read out by the control chip and sent to the centralized database for authentication. As an improvement, ReSC-2 enables mutual authentication between RFID readers and tags and does not require RFID readers to backup tag traces. ReSC-2 is specific to the supply chain of IoT devices and considers more security issues associated with IoTs.

B. Overview of ReSC-2

Figure 1(a) demonstrates our proposed RFID approach aimed at addressing different challenges/issues in the supply chain for IoT devices. Our proposed RFID system for IoT devices would consist of the following: (i) a front-end composed of RFID tags and readers; (ii) IoT devices equipped with RFID tags that include read-only tag identities (*tag IDs*) stored in the locked memories; (iii) locations associated with each reader; (iv) a back-end consisting of a centralized database (DB) that stores information (e.g., tag identities, control chip identities, tag traces, etc.) and authenticates IoT devices. Figure 1(b) illustrates the hardware architecture of ReSC-2, including the entities involved and their connections. Central to our approach are two new features: i) the RFID tag and the control chip are bound together with a one-to-one mapping to prevent potential split attacks; ii) tag identity, control chip identity, and tag trace can be sent to the database for authentication over encrypted Ethernet. We divide the supply chain for IoT devices into three states (S1, S2, and S3) and there are two possible state transitions (T1 and T2) between states as shown in Figure 1(a). Figure 1(c) illustrates the state transition graph of our proposed scheme.

1) S1. System Integration: This state is essentially the start of the IoT device's life and occurs in an untrusted environment. Measures to ensure that system integrators of IoT devices will follow our proposed procedures (i.e., functional and reliability testing, physical inspection, initialization steps, etc.) and will not behave immorally or illegally (e.g., overproduce IoT devices, enroll wrong pairs of tag and control chip identities, etc.) are going to be developed in our future work. Functional and reliability testing will be finished to ensure both hardware and software work as expected. Initialization steps are also finished and would include extracting tag identities (*tag IDs*) and control chip identities (*CC IDs*) from RFID tags and control chips separately and storing them in the centralized database. The system integrator will also assign a set of session keys (i.e., k_1, k_2, ..., k_N) to the tag and store them in the tag memory as part of initialization steps, where N corresponds to the number of readers on the tag's distribution path. All this information will be used later to track IoT devices as they move through the supply chain and verify their identities. IoT devices would eventually move into the next states which are also untrusted (susceptible to attacks). We refer to this as transition T1.

2) S2. Distribution: In this state, IoT devices are stored in inventories and transported between supply houses, distributors, retailers, and installers. In an RFID-enabled supply chain, IoT devices can be tracked using an *offline* (unplugged and disconnected from the network) mode. Tags are powered by the readers and communicate their authentication information (more in subsection C). RFID readers on the distribution path will jointly create a unique tag trace that is stored in the tag memory. To address the drawbacks of existing protocols as mentioned in Section II, a more secure and practical tag trace enrollment and validation procedure

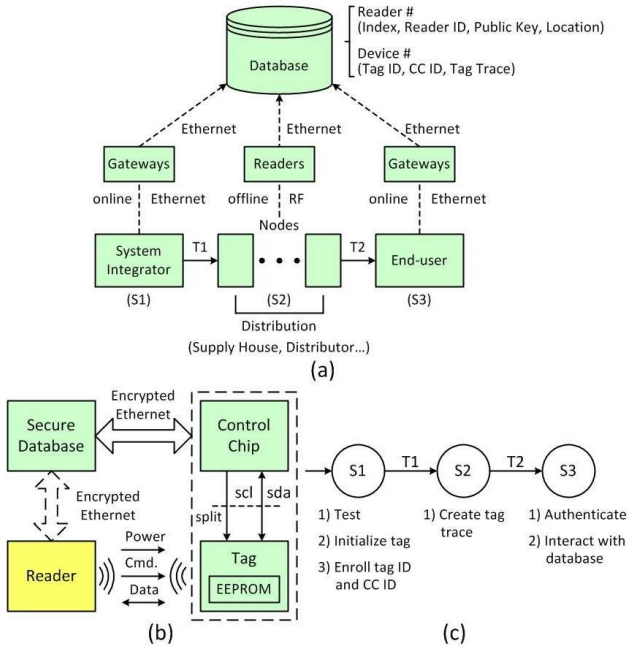

Figure 1: (a) Supply chain framework, (b) Hardware architecture, and (c) State transition graph.

will be presented in subsection C. Since the communication with RFID tag is done only by readers in the distribution and the devices are unplugged, we refer to this as *offline* mode. If IoT devices are stolen from any intermediate stage of the supply chain, the tag traces stored in the tag memories of those devices would be incomplete since they cannot contain signatures of readers at all subsequent stages. Therefore, hardware theft could be detected by verifying the completeness of tag trace when the IoT device is plugged into the network.

3) S3. End-user: Eventually, IoT devices will be deployed in the homes or businesses of end-users. We refer to this as transition T2. In this state, the IoT device will interact over encrypted Ethernet with the centralized database. Since this occurs when the IoT device is powered and operates over cable or wireless WiFi, we refer to this as *online* mode. Authentication procedures will be performed before network service (e.g., downloading necessary firmware updates, etc.) is available. Access control is available by rejecting service requests from suspicious IoT devices that fail the authentication procedures.

C. Authentication of ReSC-2

Overall, the authentication procedure of ReSC-2 can be split into two phases: (i) verification of the matching between tag identity (*tag ID*) and control chip identity (*CC ID*); (ii) verification of the integrity of tag trace to make sure that the IoT device has passed through the valid supply chain before arriving at the end-user. Compared with the original ReSC approach, ReSC-2 follows the similar authentication phases but adopts different protocols and implementations.

1) ReSC-2 I: Tag Matching with Device: At the system integration stage, the control chip identity (*CC ID*) will be generated from the start-up signature (SRAM PUF) of embedded SRAM inside the control chip. The control chip identity together with the tag identity will compose a 2-tuple (*CC ID*, *tag ID*) and is stored in the centralized database in *online* mode for future device authentication. The communication between the control chip and the database is assumed protected by cryptographic protocols such as TLS [28]. Potential split attacks can be detected since we bind the RFID tag and the IoT device together with the one-to-one mapping between tag identity and control chip identity. Even if the attacker could probe the I2C channel [35] connecting tag and control chip,

intercept the packets being transmitted between the tag and the control chip, and program them into the cloned tag, we can still detect this type of eavesdropping since the tag identity stored in the tag memory only matches one specific control chip identity (which is never communicated in plaintext). The tag memory includes three parts: one unique, read-only tag identity, a set of session keys (i.e., k_1, k_2, ..., k_N), and one unique tag trace composed of the signatures of readers on the distribution path of that tag.

2) ReSC-2 II: Valid Tag Trace: Leading manufacturers are moving away from basic make-to-stock (MTS) approach to make-to-order (MTO), configure-to-order (CTO) and engineer-to-order (ETO) production models [36]. For example, most Cisco products use the CTO production model [37]. It is a trend that more and more manufacturers will give up MTS model and embrace more MTO, CTO and ETO models. Actually, only MTO, CTO and ETO models could represent the benefits of IoT (i.e., reduced inventory and lead time, increased throughput, etc.). ReSC-2 is designed against MTO, CTO and ETO production models, in which case the system integrator could know the tag's distribution path in advance. We define a tag trace as valid when it carries all the necessary signatures of authorized readers on its distribution path. Before entering the supply chain, the system integrator will assign a set of session keys (i.e., k_1, k_2, ..., k_N) to the tag and store them in the tag memory, where N corresponds to the number of readers on the tag's distribution path. Each session key (k_i) will be used to encrypt the communication between the tag and one specific reader (R_i). The session key (k_i) will be computed as follows:

$$k_i = AES_{key_{R_i}}(tagID) \tag{1}$$

where key_{R_i} is the master key shared between the reader R_i and the centralized database. Measures to ensure that system integrators of IoT devices will follow our proposed procedures (i.e., functional and reliability testing, physical inspection, initialization steps, etc.) and will not behave immorally or illegally (e.g., overproduce IoT devices, enroll wrong pairs of tag and control chip identities, etc.) are going to be developed in our future work. When the IoT device is distributed in the supply chain, RFID readers dispersed at different locations will join up to create a unique tag trace and store that trace in the tag memory. The tag memory will include a static read-only tag identity, a set of session keys, and a unique tag trace composed of the signatures of readers on the tag's distribution path. Public key cryptography based digital signature technique (e.g., GMR signatures [38], etc.) is used to generate reader's signature. The centralized database can look up the public key (pk_i) of each reader (R_i) using the index ($Index_i$) of that reader. Figure 2 illustrates our proposed light-weight RFID protocol with cyclic redundancy check (CRC) and XORing with random numbers omitted for brevity of expression. The entire communication flow between RFID reader and tag can be divided into the following three steps:

Step 1: When the IoT device arrives at the next intermediate stage, the reader R_i at that stage will first issue a *Query* command to the tag together with a random number RN_1.

Step 2: After receiving the *Query* command and the random number RN_1, the tag will encrypt the 2-tuple (RN_1, RN_2) using the session key k_i and reply with its identity (*tag ID*) and the ciphertext $AES_{k_i}(RN_1, RN_2)$, where RN_2 is a new random number generated by the tag.

Step 3: After receiving the 2-tuple ($tag\ ID$, $AES_{k_i}(RN_1, RN_2)$), the reader R_i will generate the session key k_i locally by encrypting the tag identity (*tag ID*) using its master key key_{R_i}. The reader R_i authenticates the tag by decrypting $AES_{k_i}(RN_1, RN_2)$ and validating RN_1. If the tag passes the authentication, the reader R_i will first generate a signature

$$SIGN_i = H_{sk_i}(tagID||Index_i||TS_i) \tag{2}$$

where $Index_i$ is the index associated with the i_{th} reader, TS_i denotes the specific time when reader R_i updates the tag, $||$ indicates the concatenation operation, and $H_{sk_i}(X)$ indicates encrypted hash value of input argument X using sk_i as the private key of reader R_i. Next, the reader R_i will encrypt the quad (RN_2, $SIGN_i$, $Index_i$, TS_i) using the session key k_i and send the ciphertext $AES_{k_i}(RN_2, SIGN_i, Index_i, TS_i)$ to the tag to update the tag trace. The tag authenticates the reader R_i by decrypting $AES_{k_i}(RN_2, SIGN_i, Index_i, TS_i)$ and validating RN_2. If the reader R_i passes the authentication, the tag will store the reader update ($SIGN_i$, $Index_i$, TS_i) in the tag memory.

When the IoT device is installed at the end-user, the control chip will read out the chain of readers' signatures from the tag memory and transfer it to the centralized database for validation in *online* mode. The database has stored the correct supply chain trace associated with each tag. It will use the index $Index_i$ of each reader to look up its public key pk_i and then validate the signature $SIGN_i$ using that public key pk_i. If the chain of readers' signatures is incomplete or does not match the expected trace (stored in the database), the service request from the suspicious IoT device will be rejected by the server.

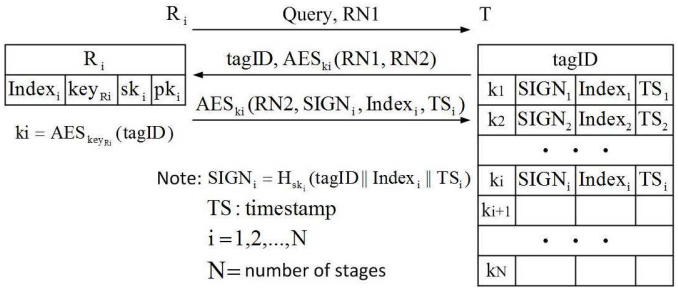

Figure 2: Light-weight RFID protocol.

3) Authentication at the End-user: The authentication at the end-user involves the following two steps: (i) the control chip authenticates itself to the centralized database based on its identity; (ii) the control chip reads out the contents of tag memory (i.e., tag identity and tag trace), encrypts them, and transmits them to the centralized database for validation. All these two steps are performed in *online* mode. The second step can ensure not only that the device has passed through the legal supply chain (as described above) but also that the tag is genuine and bound to that specific device. Service is only available to the end-user after all the authentication procedures (i.e., tag matching with device, valid tag trace including all the necessary signatures of authorized readers on the distribution path, etc.) are passed. This shall prevent stolen and/or counterfeit products from being used, making them worthless. The authentication procedures can also be performed in retail stores before purchase.

V. Security Evaluation

The quality of control chip identity (SRAM PUF) and RF communication efficiency between RFID reader and tag have been verified by prior work [15]. In this section, we evaluate the security of ReSC-2 in the context of IoT. We divide all the potential attacks/risks into five categories in terms of attack targets and discuss them respectively.

RFID tag: By binding the RFID tag and the identified IoT device together with a one-to-one mapping between tag identity (*tag ID*) and control chip identity (*CC ID*), cloning tag ID can be detected and service request will be rejected by the server when the cloned IoT device is installed at the end-user. Since the tag trace of ReSC-2 depends on both reader information (i.e., reader's index and private key), tag information (i.e., tag identity), and the specific time (i.e., *timestamp*) when the tag trace is updated, it is unique for each device and thus is resistant to duplication attack (duplicating tag trace by untrusted entities involved in the supply chain). When a rogue employee uses an authorized reader to update a stolen device, we can catch him or her since the *timestamp* embedded in the reader's signature could indicate who is on duty at that time. Protecting tag privacy is out of the scope of this paper.

Control chip / Device component: Illegal substitute of control chip or other device component (i.e., replacing original control chip or device component with a counterfeit or tampered IC/component) can be detected with unclonable control chip identity (SRAM PUF). Recycled ICs/components can be detected by a suite of solutions including light-weight, on-chip structures based on ring oscillators (RO-CDIR), anti-fuses (AF-CDIR) and fuses (F-CDIR) [11]. Hardware Trojans contained in the ICs can be detected by a series of techniques based on side-channel signal analysis, functional test, etc [10]. All the above-mentioned hardware

Trojan and die/IC recycling detection approaches can be simply integrated into ICs/components mounted on IoT devices and be compatible with ReSC-2.

IoT device: IoT devices stolen from inventories or shelves can be detected since their tag traces are either incomplete or fake and will fail the tag trace validation procedure. IoT devices stolen from homes/businesses can be detected by authentication among neighboring devices, which is going to be developed in our future work. Illegal access by malicious network nodes can be prevented by access control mechanisms integrated into IoT hardware [9]. Physical tampering can be detected or prevented by adding tamper-proof or tamper-resistant circuitry [14] to IoT hardware. ReSC-2 supports or at least be compatible with all the above-mentioned protection mechanisms.

RF channel: Sensitive information (e.g., reader's private key) is never transmitted in clear and thus is resistant to eavesdropping. Duplication attack can be prevented since reader update (i.e., new signature generated by current reader) is generated based on one specific tag identity and cannot be simply duplicated to be used for another tag. In the worst case, even if the adversary performs duplication attack and tag ID cloning simultaneously, the copies of reader updates obtained by duplication attack could match the cloned tag identities. Those cloned tag identities would not match the control chip identities. ReSC-2 is resistant to replay attack since freshly generated random numbers are used as the challenges for mutual authentication to verify whether both sides possess the shared session key and would be used only once. ReSC-2 is also resistant to man-in-the-middle attack. When the adversary intercepts the 2-tuple ($tag\ ID$, $AES_{k_i}(RN_1, RN_2)$) sent by the legal tag to the authorized reader, changes the tag identity ($tag\ ID$) contained in the 2-tuple to any other wanted tag identity ($tag\ ID'$), and sends the forged 2-tuple ($tag\ ID'$, $AES_{k_i}(RN_1, RN_2)$) to the authorized reader to swindle reader update associated with that wanted tag identity, the forged 2-tuple will fail the authentication by the authorized reader since RN_1 cannot be recovered by decrypting $AES_{k_i}(RN_1, RN_2)$ using a different session key k_i' computed based on $tag\ ID'$. When the adversary alters the reader update $AES_{k_i}(RN_2, SIGN_i, Index_i, TS_i)$ sent by the authorized reader, the forged reader update will be detected by the tag since RN_2 cannot be recovered from it. ReSC-2 is resistant to denial-of-service attack since the RFID reader can only update the tag memory after passing the tag's authentication.

RFID reader: Spoofed tags usually originate from an illegal channel and cannot possess correct session keys. As described in Section IV, the reader R_i can authenticate the tag by decrypting $AES_{k_i}(RN_1, RN_2)$ and validating RN_1.

VI. Conclusion

In this paper, we have presented an RFID-based solution that enables traceability and authentication of IoT devices across the supply chain called ReSC-2. Compared with existing approaches, ReSC-2 has the following merits: (1) By binding the RFID tag and the identified device together with a one-to-one mapping, potential split attacks (i.e., separating tag from product, swapping tags, etc.) can be detected; (2) By combining two techniques (i.e., one-to-one mapping between tag identity and control chip identity, unique tag trace composed of signatures of readers on the distribution path) together, ReSC-2 can address most of security and privacy challenges for IoT supply chain; (3) The fabrication cost is quite low since the vast majority of components (e.g., voltage regulator, control chip with embedded SRAM, etc.) in this design already exist in many modern IoT devices.

References

[1] Carnegie Mellon University. The "Only" Coke Machine on the Internet, 1982.

[2] Dave Evans. The Internet of Things: How the Next Evolution of the Internet Is Changing Everything. *CISCO white paper*, 1, 2011.

[3] Tobias Zillner and Sebastian Strobl. ZigBee Exploited - The good, the bad and the ugly. In *Black Hat USA 2015*, 2015.

[4] Proofpoint. Proofpoint Uncovers Internet of Things (IoT) Cyberattack, Jan 2014.

[5] Eduard Kovacs. Attackers Use Stolen Credentials to Hack Cisco Networking Devices, August 2015.

[6] Joan Daemen and Vincent Rijmen. *The design of Rijndael: AES-the advanced encryption standard*. Springer Science & Business Media, 2013.

[7] Hugo Krawczyk, Ran Canetti, and Mihir Bellare. HMAC: Keyed-hashing for message authentication. 1997.

[8] Ling Hu and Cyrus Shahabi. Privacy assurance in mobile sensing networks: go beyond trusted servers. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on*, pages 613–619. IEEE, 2010.

[9] Steven Shannon. Access control of networked data, May 15 2001. US Patent 6,233,618.

[10] Mohammad Tehranipoor and Farinaz Koushanfar. A survey of hardware trojan taxonomy and detection. 2010.

[11] Ujjwal Guin, Xuehui Zhang, Domenic Forte, and Mohammad Tehranipoor. Low-cost On-Chip Structures for Combating Die and IC Recycling. In *Proceedings of the 51st Annual Design Automation Conference*, pages 1–6. ACM, 2014.

[12] Kerry Bernstein. Supply Chain Hardware Integrity for Electronics Defense (SHIELD), March 2014.

[13] Donald G Robinson, Michael W Geatz, and Michael J Corcoran. Retail theft prevention and information device, December 31 1996. US Patent 5,589,820.

[14] Yuval Ishai, Manoj Prabhakaran, Amit Sahai, and David Wagner. Private circuits II: Keeping secrets in tamperable circuits. In *Advances in Cryptology-EUROCRYPT 2006*, pages 308–327. Springer, 2006.

[15] Kun Yang, Domenic Forte, and Mark Tehranipoor. ReSC: RFID-enabled Supply Chain Management and Traceability for Network Devices. In *The 11th Workshop on RFID Security*, 2015.

[16] G Edward Suh and Srinivas 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.

[17] Daniel E Holcomb, Wayne P Burleson, and Kevin Fu. Power-up SRAM State as an Identifying Fingerprint and Source of True Random Numbers. *Computers, IEEE Transactions on*, 58(9):1198–1210, 2009.

[18] Berk Sunar, William J Martin, and Douglas R Stinson. A provably secure true random number generator with built-in tolerance to active attacks. *Computers, IEEE Transactions on*, 56(1):109–119, 2007.

[19] David Zanetti, Leo Fellmann, and Srdjan Capkun. Privacy-preserving clone detection for RFID-enabled supply chains. In *RFID, 2010 IEEE International Conference on*, pages 37–44. IEEE, 2010.

[20] Davide Zanetti, Srdjan Capkun, and Ari Juels. Tailing RFID Tags for Clone Detection. In *NDSS*, 2013.

[21] ZigBee Alliance. ZigBee Specification, January 2008.

[22] Ujjwal Guin, Domenic Forte, and Mohammad Tehranipoor. Anti-counterfeit techniques: from design to resign. In *Microprocessor Test and Verification (MTV), 2013 14th International Workshop on*, pages 89–94. IEEE, 2013.

[23] Department of Justice. Departments of Justice and Homeland Security Announce 30 Convictions, More Than $143 Million in Seizures from Initiative Targeting Traffickers in Counterfeit Network Hardware, May 2010. http://www.justice.gov/opa/pr/departments-justice-and-homeland-security-announce-30-convictions-more-143-million-seizures.

[24] FreightWatch International Supply Chain Intelligence Center. 2013 Global Cargo Theft Threat Assessment, 2013.

[25] John R. Ellement. Three men face charges in $1m computer parts theft ring. *The Boston Globe*, 2014.

[26] Ravi S Sandhu and Pierangela Samarati. Access control: principle and practice. *Communications Magazine, IEEE*, 32(9):40–48, 1994.

[27] EPCglobal Inc. EPC Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz-960 MHz Version 1.2.0, May 2008.

[28] Tim Dierks. The Transport Layer Security (TLS) Protocol Version 1.2. 2008.

[29] Network Working Group. Security Architecture for the Internet Protocol. 2005.

[30] Jonny Milliken, Valerio Selis, and Alan Marshall. Detection and analysis of the Chameleon WiFi access point virus. *EURASIP Journal on Information Security*, 2013(1):1–14, 2013.

[31] Grant Hernandez, Orlando Arias, Daniel Buentello, and Yier Jin. Smart nest thermostat: A smart spy in your home. *Black Hat USA*, 2014.

[32] Ravi S Sandhu, Edward J Coyne, Hal L Feinstein, and Charles E Youman. Role-based access control models. *Computer*, (2):38–47, 1996.

[33] Ray Beaulieu, Douglas Shors, Jason Smith, Stefan Treatman-Clark, Bryan Weeks, and Louis Wingers. The SIMON and SPECK Families of Lightweight Block Ciphers. *IACR Cryptology ePrint Archive*, 2013:404, 2013.

[34] Kun Yang, Domenic Forte, and Mark Tehranipoor. An RFID-based Technology for Electronic Component and System Counterfeit Detection and Traceability. In *Technologies for Homeland Security, 2015 IEEE International Symposium on*, 2015.

[35] NXP Semiconductors. I2C Bus Specification and User Manual, Apr. 2014.

[36] Christopher Holmes. Designing and Implementing the Factory of the Future at Mahindra Vehicle Manufacturers, April 2015.

[37] Cisco. How Cisco Transformed Its Supply Chain, May 2014.

[38] Shafi Goldwasser, Silvio Micali, and Ronald L Rivest. A Digital Signature Scheme Secure Against Adaptive Chosen-message Attacks. *SIAM Journal on Computing*, 17(2):281–308, 1988.