A Taxonomy for the Security Assessment of IP-based Building Automation Systems: The Case of Thread

Yu Liu, Zhibo Pang, *Senior Member, IEEE,* György Dán, *Senior Member, IEEE,* Dapeng Lan, and Shaofang Gong, *Member, IEEE*

***Abstract*—Motivated by the proliferation of wireless building automation systems (BAS) and increasing security-awareness among BAS operators, in this paper we propose a taxonomy for the security assessment of BASs. We apply the proposed taxonomy to Thread, an emerging native IP-based protocol for BAS. Our analysis reveals a number of potential weaknesses in the design of Thread. We propose potential solutions for mitigating several identified weaknesses and discuss their efficacy. We also provide suggestions for improvements in future versions of the standard. Overall, our analysis shows that Thread has a well-designed security control for the targeted use case, making it a promising candidate for communication in next generation BASs.**

***Index Terms*—Building Automation Systems; Thread; Security Analysis**

I. Introduction

The main purpose of a building automation system (BAS) is to provide functional services for security, access control, and for controlling and managing lighting, climate, and other mechanical and electrical systems in a commercial or residential building (home) [1]. In the past, BAS systems used to operate in isolation, but there is an increasing demand to interconnect BASs with the Internet and with corporate networks for ease of management. The integration of BASs and recent cyber-attacks against industrial automation systems and against BASs has triggered an increasing concern that BAS systems could become a target of cyber-attacks, or could be used for launching attacks targeted against the corporate network with severe consequences, e.g., as in the case of the 2014 attack against the retail company Target through an HVAC provider [2].

Motivated by the changing requirements, especially security concerns, the BAS industry has recently started to explore wireless, native IP-based solutions [3], as a complement to traditional wired, non-IP based BAS standards, such as KNX, BACnet, LonWorks, and DALI [4]. Wireless, native IP-based solutions offer three main advantages compared to legacy BAS standards. First, the adoption of wireless communication significantly reduces the cost of system deployment, simplifies maintenance, and provides long term reliability. Second, a native IP-based BAS can be easily integrated with other IT systems, providing interoperability among different communication protocols. Last, but not least, native IP-based BASs can better address the growing security concerns of customers. Together with ease of maintenance and interoperability, security has thus become a main driver for a new generation of BAS solutions.

A notable example of emerging native IP-based wireless BAS communication protocols is Thread [5] , which has been designed for providing secure and reliable wireless communication for building automation. Thread has found adoption in industry, and may have the potential to replace legacy BAS standards.

Surprisingly, however, even though one of the main drivers for new building automation standards is security, there does not exist a comprehensive security analysis taxonomy for BAS protocols, and there is no systematic security analysis of Thread within a comprehensive taxonomy. The lack of a security analysis taxonomy for BAS protocols hinders the development of an industry-wide BAS standard, while the lack of a security analysis of Thread may hinder its wide-spread adoption by the industry.

In this paper, we make three important contributions to address this gap. First, after reviewing the security-relevant characteristics of BASs, we propose a comprehensive security assessment taxonomy for BAS protocols, which covers all phases of the interaction between the devices that constitute the BAS, starting from when a device is commissioned into the network until it is decommissioned. Second, we provide a security assessment of the Thread protocol using the proposed taxonomy, and validate the assessment by experimental results. Finally, we propose improvements to address potential security issues in the Thread protocol, so as to strengthen its potential for wide-spread adoption in BASs.

The rest of the paper is organized as follows. In Section II we review related work, in Section III we characterize BASs and present the security analysis taxonomy for modern BASs. In Section IV we apply the taxonomy to give a qualitative assessment of the security of Thread. In Section V we propose and discuss security enhancements for Thread, and Section VI concludes the paper.

II. Related Work

There have been a number of recent works on BAS security. In [6], Mundt et al. concluded several weaknesses in modern

Y. Liu and S. Gong are with Linköping University, Sweden.
Z. Pang is with ABB Corporate Research, Sweden.
G. Dán is with KTH Royal Institute of Technology, Sweden.
D. Lan is with University of Oslo, Norway.

BASs ranging from field layer to automation and management layer, based on practical attacks against two office buildings where KNX and BACnet were deployed. Brauchli et al. in [7] conducted their security assessment on a real home automation system, digitalSTORM, and derived attack vectors of BASs involving servers, devices, the communication bus and remote access. These case studies did not consider all security aspects of BASs. Similarly, recent work [8] [9] [10] [11] analyzed particular security aspects and vulnerabilities of ZigBee based BASs. In [8] the authors formulated security requirements for ZigBee networks with a focus on communication security, while [9] presented an experimental evaluation of the security of gateways in home automation. In [10] the authors explored an energy depletion attack against ZigBee based wireless networks and built an analytical model to evaluate the impact of the vulnerability. In [11] the authors showed that it is possible to access secret keys in the TI CC2430, which was designed for use in ZigBee networks, through physical access to the device, highlighting the importance of protecting secrets from physical access. None of these works, however, followed a mature security analysis taxonomy to perform the analysis, nor did they provide a system level security analysis of BASs.

Related to BAS security are recent works on the security analysis of wireless sensor networks (WSN) and the Internet of Things (IoT). In [12] the authors classified threats targeting WSNs into passive and active attacks. Authors in [13] surveyed the security of three wireless mesh networks, ZigBee PRO, WirelessHART and ISA100.11a, using a taxonomy that distinguishes active and passive attacks, and insider and outsider attacks, and examined each threat in terms of its impact on confidentiality, integrity and availability. In [14] the authors compared the security features of KNX-RF, EnOcean, ZigBee, Z-Wave and Thread, with emphasis on the cryptographic methods and on implementation issues. However, in lack of a mature threat model, these coarse-grained taxonomies can hardly capture all threats and thus they cannot ensure the comprehensiveness of the security analysis.

In [15]–[19] the authors analyzed the security of WSNs and home area networks (HAN) following the layers of the communication protocol stack. In [15], the authors matched the threats in each layer of the OSI model with corresponding countermeasures. Authors in [16], [17] identified potential risks in the physical, the media access control (MAC), network, routing and application layers, illustrated with representative protocols used in each layer, including IEEE 802.15.4, 6LowPAN, RPL, CoAP, etc. Authors in [18] analyzed the security of HANs using a three-layer stack, namely perception layer, network layer and application layer, and classified threats into privacy, physical, disaster and damage categories. Such a layer-based analysis taxonomy is less relevant to BASs for two reasons. First, the security analysis of layers in isolation does not provide a comprehensive view of a BAS protocol, as the weakness of one layer can be compensated by another layer. Second, these taxonomies focus only on communication security, and disregard other security aspects in BASs.

A series of works by Granzer and Kastner provided an early outline of security analysis of BASs. In [20], the authors classified attacks against BASs into network attacks and device attacks, and proposed approaches to securing communication and devices in a BAS. Communication security encompasses traditional aspects like data integrity, freshness, confidentiality, and in addition it includes new concepts, such as *initial configuration*, *secure binding*, and *secure unbinding*. Device-level security is based on a two-level firmware: a generic operating system that allows the execution of customized applications. In [21], various BAS domain-specific challenges were described that distinguish BAS from other wireless networks. A comparison and a security assessment of KNX, LonWorks, BACnet and ZigBee was presented in [20]–[22].

While the aforementioned studies cover many important aspects of BAS security, they fall short on considering the complete life cycle of BASs, in particular, the security of commissioning new devices and of decommissioning devices from the BAS network, which are equally important as the security of communication and of devices during operation. In this paper we address this shortcoming by considering security during the entire life-cycle of a BAS.

III. BAS SECURITY TAXONOMY

A BAS consists of a network of devices for providing services, and is enabled by a BAS communication protocol [1], [3]. In this section we discuss security-relevant characteristics of BASs, and propose a taxonomy for BAS protocol security assessment.

A. Security Relevant Characteristics of BASs

Among the characteristics and requirements of BASs discussed in [3], [20] we identified four key characteristics that make securing BASs particularly challenging.

1) Limited resource availability [3] [20] [21]: Sensors and actuators in BASs are usually energy constrained embedded devices with limited computational resources, and the available memory and storage are usually limited to a few kilobytes to meet necessary runtime requirements. Furthermore, battery performance may degrade over time and due to environmental changes, such as temperature, and may affect the computational or communication capability of a device. To ensure security under limited resources, computationally inexpensive cryptographic algorithms and protocols have to be used [20], together with adaptive communication solutions.

2) Diverse topologies [3] [21]: A single BAS could consist of as many as thousands of devices that exchange data and commands with each other using point-to-point (unicast) and group (multicast) communication to provide a variety of services. Customization for different buildings implies that BASs may have different topologies depending on the layout of the building, and thus the network structure and the security solutions need to be scalable and adaptive, so that they can support a large number of point-to-point and group communication sessions [21].

3) Physical access [20]: Devices in a BAS typically operate in an open environment, hence they can be exposed to physical access. If an adversary manages to compromise a node that stores sensitive data or network credentials in

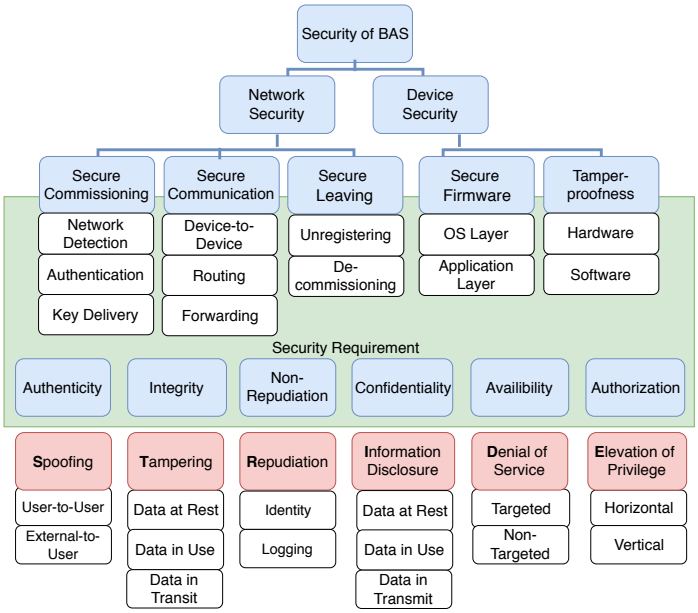

Fig. 1. Proposed Security Analysis Taxonomy for BASs.

memory or in flash storage, the node may potentially leak those data [11].

4) Continuous maintenance [20]: BASs can have long life cycles, especially large scale BASs that consist of thousands of sensors and actuators and are used in commercial buildings for as long as a decade. Device and system maintenance in the form of firmware upgrades is thus essential for deploying new functionalities and for protecting the devices and the system from new attacks that exploit algorithmic or software vulnerabilities. Firmware upgrades need, however, to be performed in a secure manner so as to avoid system compromise.

B. Taxonomy for Security Assessment

Extending the precursory work by Granzer and Kastner [20] [21], in what follows we present a comprehensive taxonomy for BAS protocol security analysis. Besides considering network security and device security as in [20], the proposed taxonomy covers all five phases of the interaction between individual devices and the building automation network (BAN), starting from when a device joins the network until it leaves the network. Furthermore, we divide each phase into smaller units, so as to enable a fine-grained security analysis that covers the whole life cycle of BASs. The taxonomy considers threats from both an active and a passive adversary that has access to the communication channels and the devices that constitute the network, according to the STRIDE threat classification model, and formulates desirable system properties under these threats. Fig. 1 shows an overview of the proposed taxonomy.

1) Five Phases of Interaction:

a) Secure Commissioning: Commissioning in a BAS is the process through which a device locates the correct BAN that it should attach to, mutually authenticates with the authorized commissioner in the BAN, it is entrusted with network data and credentials, and finally it establishes a secure communication channel with other entities in the network. We choose to use the term *Secure Commissioning* instead of *Initial Configuration* and *Secure Binding* used in [20], because the act of uploading prepared configuration data (e.g., network address) to a device is not a necessary step in all BASs, as some devices can either auto-configure or could receive default data during manufacturing.

In order for commissioning to be secure, a BAS standard has to guarantee security during network detection, authentication and during the delivery of network secrets. It has to protect the network from spoofing, man-in-middle and denial-of-service attacks and it should not leak any secret about the network to an unauthorized third party.

b) Secure Communication: After commissioning, a device implements or contributes to functions through communicating over the established channels using the network credentials. In order for the communication to be secure, routing information exchange, message forwarding and device-to-device data delivery should be secure.

c) Secure Leaving: Once a successfully commissioned device is forced to leave or decides to join another BAN, the removal of the device from the network (unregistering) and destruction of sensitive information about the BAN (de-commissioning) have to be performed in a secure manner, as exposure of the network credentials could be used for violating the security of other BAN devices.

d) Secure Firmware: A typical BAS device executes code stored in non-volatile memory, i.e., firmware. The code can be divided in two levels: a low level operating system (OS) that manages hardware and provides interaction between resources and applications, and high level applications that contain functional tasks. A compromise at any of these two levels implies compromise of the device [20]. Furthermore, upgrading functional tasks of the device requires upgrading the firmware. Thus, secure OS and application code development, as well as secure configuration and firmware upgrades are essential.

e) Tamper-proofness: A BAS device needs to be protected from compromise through tampering, as a compromised device may be used for obtaining the network secrets or for manipulating data. Protection is unlikely to be achieved by physical isolation, thus tamper protection needs to be applied on the chips and the memory inside the devices, to protect the hardware and the software.

2) Security Requirements: Our taxonomy formulates security requirements for all five phases, in face of threats structured according to a refined STRIDE threat model, and considers both active and passive adversaries. In the refined STRIDE model, we distinguish between spoofing by an internal and an external adversary, we consider tampering and information disclosure attacks against data at rest, in use and in transit, we consider repudiation attacks through affecting identity management and logging, we distinguish between targeted and non-targeted DoS attacks, and between horizontal and vertical elevation of privilege attacks.

a) Authenticity: Authenticity requires that data are sent from an authorized sensor or a command is sent by a legitimate origin from within the BAN. Authenticity extends integrity as it allows to verify the identity of the sender of data or

commands, thus it can prevent unauthorized devices from crafting messages for causing malfunction.

b) Integrity: The integrity of all information, data and commands, transmitted over the BAN must be verifiable and valid at the time of reception, so as to mitigate injection, modification, delay and replay of messages, as modified or crafted data and commands may damage control systems while replayed messages allow adversaries to gain control of BAS facilities in a building [23].

c) Non-repudiation: The system should be able to counter repudiation attacks, e.g., if a user denies having performed an action then the system should be able to verify the validity of the claim. Non-repudiation requires secure identities, but also secure logging and auditing.

d) Confidentiality: All information, data and commands, transmitted over the BAN must be kept confidential, i.e., they must not be disclosed to any unauthorized nodes, as the leak of commands and sensor data could provide an adversary with the knowledge of the control commands for functional devices in the BAS, and could reveal environmental information and current conditions of the building, with privacy invasion as a potential consequence.

e) Availability: Availability ensures that the BAS makes information available whenever it is needed, i.e., a legitimate device in a BAN is able to realize its functions by providing data to or receiving data or commands from authorized nodes despite component failures or adversarial activity, such as a DoS attack.

f) Authorization: Authorization ensures that access to BAS resources, services and actions is constrained, and prohibits adversaries from exploiting and exploring the system.

IV. SECURITY IN THREAD

Before we apply the security analysis taxonomy to the Thread protocol, we provide a brief description of the entities and protocols used in Thread [5].

A. Thread Architecture Overview

A Thread Network consists of Thread devices, some of which may become Routers, and one or more Border Routers that connect the Thread Network to external networks, e.g., to other Thread networks. Each Router is assigned a 16 bit address by a Routing Leader, which is elected automatically among the routers, while the address of regular devices is derived from the address of the Router they are connected to. In addition, devices can have multiple IPv6 Unique Local Addresses and Global Unicast Addresses.

Thread relies on IEEE Standard 802.15.4-2006 at the physical and at the link layer. It uses the Mesh Link Establishment (MLE) protocol [24] for link configuration, network-wide parameter dissemination and neighbor detection, and for maintaining link reliability information. Thread uses the 6LoWPAN protocol for adaptation to IPv6. A distance vector routing protocol, in which link cost is determined by the Received Signal Strength Indicator (RSSI), is used to allow for up to 32 Routers in the Thread Network, due to the 802.15.4 frame length limit. For multicast delivery Thread relies on the Multicast Protocol for Low Power and Lossy Networks (MPL). Multiple Thread Networks can be connected by Border Routers to allow for more devices. On top of IP Thread uses the CoAP protocol over UDP.

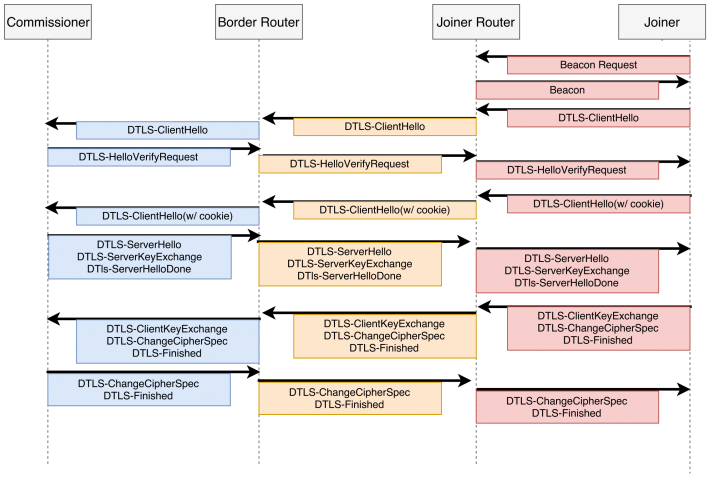

Fig. 2. Thread Commissioning Handshake Procedure [27]

B. Security Analysis

In what follows we use the proposed security analysis taxonomy to analyze the Thread protocol.

1) Commissioning: Commissioning in Thread consists of two phases. First, a commissioner candidate needs to be authenticated to the Leader; once authenticated, it becomes the sole Commissioner in the network, and the Leader informs all Joiner Routers how to reach the Commissioner. This phase is referred to as petitioning. Second, a new device is authenticated by the commissioner, and entrusted with network credentials (e.g., master key) by the Joiner Router [5]. This phase is referred to as joining. These two phases ensure that only an authorized device can become the sole commissioner and perform commissioning.

Petitioning and joining are both done using datagram transport layer security (DTLS) combined with Password-Authenticated Key Exchange by Juggling (J-PAKE) for key establishment [25]. J-PAKE relies on the Schnorr Non-interactive Zero Knowledge Proof (NIZK) for providing mutual authentication and key agreement based upon a low-entropy pre-shared secret, which in the case of Thread is the Pre-Shared Key for Commissioner (PSKc) for petitioning, and the Pre-Shared Key for Device (PSKd) for joining. J-PAKE was recently proven to provide off-line and on-line dictionary attack resistance (attacker cannot guess session key), perfect forward secrecy (compromise of pre-shared secret does not disclose session key), and known session security (session keys are independent) [26].

Fig. 2 shows the message exchange between the Commissioner, a Border Router and the Joiner device during joining. DTLS checks the genuineness of the Joiner device using a cookie, which protects the Joiner Router from simple DoS attacks. During the last step, the Joiner Router entrusts the Joiner device with the master key protected by the pre-established key encryption key (KEK), which is delivered by the Commissioner.

Given this commissioning procedure, Thread can generally achieve *Secure Commissioning* as *Authenticity*, *Integrity*, *Confidentiality* and *Authorization* are fulfilled during authentication and key delivery, except for a few targeted flooding attacks during the network detection process, which can degrade the network performance as follows.

- Beacon request flooding attack (Attack C/1): As the first step of commissioning, Joiner device will implement a scan by sending beacon request on every channel and wait for Joiner Router sending back beacons containing network steering data [5]. This mandatory mechanism can be used in DoS attack by sending rogue beacon requests towards the Joiner Router and thus affect its performance.
- DTLS handshake flooding attack [28] (Attack C/2): As we show later in Table II, the average joining handshake time for one node is around 10 s, while multiple nodes joining simultaneously will increase the joining time by orders of magnitude with many handshake errors happening. This is due to the complexity of the J-PAKE protocol combined with limited memory resources and processing power of the devices. Even though all the nodes will join the network eventually, this limitation can still be exploited for a DoS attack against the Joiner Router.

2) Communication: In a Thread network, device-to-device communication security relies on MAC layer security and MLE layer security.

a) Physical - Medium Access Control (PHY-MAC) Layer: Thread relies on IEEE 802.15.4 for link layer security: it uses AES-CCM mode, i.e., counter with cipher block chaining message authentication code (CBC-MAC), applied to both the frame header and the payload, as detailed in [29]. Combined with a so called latest frame counter maintained in each device for every neighbor, link layer security provides access control, message integrity, confidentiality and replay protection [10], and makes Thread resilient to attacks like external message spoofing, tampering, and information disclosure. Without knowing the network credentials, an adversary cannot inject or alter network traffic, thus it can neither perform an elevation of privilege attack nor can it impersonate a legitimate device. Due to its design choices, Thread is also resilient to 802.15.4 MAC specific attacks described in [30] [31].

- Same-Nonce attack [30]: Nonce is an initialization vector used in AES-CCM mode of operation. When two messages m1 and m2 are encrypted with the same nonce and the same key into cypher texts c1 and c2, an attacker can obtain m1 XOR m2 by computing c1 XOR c2. Thread is resilient to this attack, as the nonce consists of the source address, frame counter and the security level [29], and the frame counter is incremented for each frame, and is forbidden to be reused for a given key.
- Replay attack [31]: The frame counter consists of 4 octets, thus even if a node transmits one packet per millisecond, it still takes up to 49.7 days to use up all counter values, while a typical key rotation period is far shorter than that.
- Replay-protection attack [31]: By setting the frame counter or key index in a frame to a large value, an adversary could make a node ignore all frames with smaller counter or key index. Nonetheless, since Thread includes both the frame counter and the key index into the CBC-MAC computation, a replay-protection attack would result in authentication failure.
- Guaranteed Time Slot(GTS) attacks [31]: A GTS attack targets a Contention Free Period (CFP) of superframe duration in 802.15.4 [31], but according to [5], there is no coordinator implemented in Thread that would assign a CFP. Since Thread does not make use of the optional CFP in 802.15.4, Thread is not vulnerable to GTS attacks.
- PAN ID conflict attack [31]: IEEE 802.15.4 defines a PAN ID conflict resolution algorithm when a conflicting network PAN ID is detected. This mechanism can be abused by attackers through sending a fake PAN ID conflict notification. However, in a Thread network, the Thread commissioner utilizes a Thread management command on top of CoAP to detect a PAN ID conflict with neighboring networks, and it does not accept conflict notifications from outside networks directly. This design choice suggests that Thread does not implement the PAN ID conflict resolution algorithm of 802.15.4.

Yet, Thread is vulnerable to the following attacks.

- Radio jamming [32] (Attack R/1): Thread is inevitably prone to physical layer radio jamming, which is a form of a DoS attack.
- Link layer jamming and node-specific flooding [31] (Attack R/2): Unlike radio jamming, link layer jamming creates a DoS attack by crafting link layer frames in a Thread network, so as to reduce network performance and throughput; node-specific flooding sends frames towards a particular node, either to drain its battery or to affect its functionality.
- Back-off manipulation and Clear Channel Assessment (CCA) manipulation [31](Attack R/3): An adversary could deviate from the CSMA/CA channel access mechanism used by IEEE 802.15.4, by either using a shorter back-off time or even skipping CCA. Doing so would deteriorate the throughput of the legitimate nodes in the Thread network.
- Acknowledgment(ACK) attack [30](Attack R/4): IEEE 802.15.4 does not mandate integrity or confidentiality protection for acknowledgment frames. Once a message is received by a Thread node, the node responds with an ACK frame that includes the sequence number of the received frame. Since the frame is sent in clear text, an adversary can forge an ACK message if it knows the corresponding sequence number. This can be easily done with a 802.15.4 network sniffer. Combined with a jamming attack, an adversary could isolate a Thread device by jamming all the packets sent to it and forging corresponding ACK packets. In addition, missing ACK packets result in packet retransmission up to a maximum number of times, which could potentially be used for a DoS attack.

While the above attacks can be used to degrade the perfor-

mance of a Thread network, they do not exploit design flaws in the Thread protocol, but are inherent to wireless communication. Furthermore, none of the vulnerabilities enables an attacker to violate message integrity or confidentiality.

b) MLE Layer: MLE plays an essential role in Thread security, as it enables to distribute security parameters and to synchronize the IEEE 802.15.4 replay protection between neighboring nodes. MLE uses the same AES-CCM mode as the link layer in order to minimize complexity and cost, but it uses a different key and each Thread device maintains an outgoing MLE frame counter and one incoming MLE frame counter per neighbor for AES-CCM, similar to the link layer frame counters. In addition, since the network Router and the Joiner device do not know each other's MLE frame counters upon joining, a two-way challenge-response protocol is used when the Joiner device attaches to a parent node, so as to provide replay protection.

Despite these, a potential weakness is that MLE Advertisement messages do not support challenge-response protocol, and for Link Request messages MLE only mandates a one way challenge-response [24], which may create the possibility for a replay attack if MLE messages are not protected by link layer security. For this reason Thread recommends that information from insecure MLE messages should not be used for modifying information obtained from secured messages.

To summarize, as in the MAC layer, the adoption of AES-CCM in the Thread MLE layer assures that MLE provides *Confidentiality*, *Authenticity* and *Integrity*. The MLE frame counter and the challenge-response mechanism ensure freshness is verified, thus *Integrity* is preserved. *Authorization* and *Non-repudiation* are also satisfied if network credentials are not compromised.

c) Routing and Forwarding: Thread relies on a routing protocol on top of the mesh network established by MLE for message delivery, and on 6LoWPAN for forwarding at the link layer. The security of the routing protocol and that of forwarding are thus essential. Routing and forwarding security in Thread are provided through link layer and MLE layer security, as the 6LoWPAN headers and distance-vector information are authenticated and encrypted hop-by-hop by every router. This provides protection against message alteration, redirection, black/grey hole, worm hole and link spoofing attacks by outside attackers [33] [34]. Link layer security provides replay protection to MPL as well, despite the 8 bits long sequence numbers used in MPL as specified in RFC 7731. Nonetheless, due to relying on hop-by-hop link layer security, Thread is vulnerable to compromised Routers, which could alter the forwarded 6LoWPAN headers and distance vector protocol information. At the same time, a router compromise would also reveal the network master key, hence compromising the entire network.

d) Key Management: The security of Thread hinges on the security of the network keys used in commissioning, encryption, and authentication. It is important to ensure that the management of the three types of keys used in Thread is secure: a network wide master key, a pre-shared key for the commissioner (PSKc) and another for the device (PSKd).

- Key generation: PSKc is derived from a user customized commissioning credential, to which a key stretching method is applied in order to protect the original plain text. PSKd is a device unique key set by manufacturers. The Thread master key is generated by the Leader during network creation. The specification requires that a true random number generator (TRNG) must be implemented to maximize randomness and to enhance security. Furthermore, a strong hash function is applied to the master key together with a key sequence number to produce runtime keys for MAC and MLE encryption, which effectively prevents deriving the master key from compromised MAC or MLE keys.
- Key distribution: The network master key is securely provided to the joined devices via a DTLS session after successful commissioning. During this procedure an enhanced key establishment key (KEK) generated during the DTLS handshake is used to deliver the network credentials. The KEK is used exactly once, which endows Thread with high security in key distribution.
- Key usage and storage: PSKc and PSKd are both used as pre-shared short secrets between the DTLS client and server for mutual authentication and for generating an enhanced key for the DTLS session. It shall be noted that the enhanced session key cannot be inferred from a compromised PSKc or PSKd without a record of the corresponding handshake details. The master key is the root of security in Thread communications since it is utilized to derive MAC and MLE layer keys. Thread requires that network security keys should be stored in non-volatile memory in every device within the network [5]. This is reasonable for the master key, but is unnecessary for PSKc when implemented in a BAS, because the PSKc is only used for commissioning an external commissioner with the Border Router, thus the storage of PSKc shall be restricted to the responsible Border Router, to reduce the risk of key compromise. PSKd is the device unique joining credential and it is suggested to be printed on the device label using QR code or serial number, which could be a weakness as it can be easily eavesdropped by an adversary.
- Key replacement: Thread supports the update of network security material. PSKc and the master key can be updated using Thread management commands relying on the CoAP protocol, protected by MAC layer security. An active time stamp is attached to the commissioning dataset to ensure freshness, and extra delay is introduced in addition to ensure every device can update the master key simultaneously without losing connectivity.

In summary, communication in Thread is secure to most attacks against the data or network functions, except for a few DoS attacks that can affect network performance. However, the compromise of the master key is a severe threat to Thread security as the master key is used network-wide. A compromised node can access all network data via eavesdropping data in transit, and could be used for message spoofing or for an elevation of privilege attack. Furthermore, a repudiation attack

is also possible due to the lack of logging and due to the lack of secure identities. These attacks have to be mitigated at the application layer, as we discuss later.

3) Leaving: A legitimate Thread device may decide to leave the network intentionally or be forced to leave the network it is commissioned to. The Thread specification requires that when a Thread Router migrates to another network partition, it has to detach from the original network partition and has to eliminate related network data. However, the Router still keeps the network credentials such as the master key because it is needed to attach to other network partitions.

Surprisingly, however, there is no clear description of how to securely remove a device from the network, and thus credentials could remain in a device that is forced to leave the network (Attack L/1).

4) Firmware: Thread does not have standard profiles to be used by the devices in the application layer, except for Thread management commands based on the CoAP protocol. Thus, firmware implementations of Thread devices may vary between manufacturers. This provides flexibility to customize applications according to different use cases, but also makes Thread devices security dependent on manufacturers' ability to produce good quality code. Secure communication and operation during runtime in the application layer is still achievable because Thread has support for securing CoAP communications, while application layer programs are usually implemented to realize functions for sensors and actuators and do not necessarily need to access to network credentials. An application designer should thus avoid access to security related network data so that a CoAP based application layer can rely on Thread's link layer security.

Although the standard does not mandate it, we argue that CoAP communication should be protected by establishing a DTLS session between peers.Doing so should be feasible, as DTLS support is available in every node due to the commissioning handshake. If using DTLS at the application layer, *Authenticity*, *Integrity* and *Confidentiality* can be ensured whilst repudiation and elevation of privilege are prevented as an attacker cannot eavesdrop or manipulate data in transit.

Parameter configuration and firmware upgrade in Thread are also manufacturer specific. The key is to take advantage of the built-in security infrastructure of Thread. A good solution is to use the Border Router as an upgrade server that is in charge of loading the firmware into specific devices within a Thread network, since it is the only entry to a Thread network. The manufacturer shall ensure that the delivery of the configured parameter or firmware to the Border Router is secure so that the *Authorization* requirement is ensured. Undeniably, firmware security is dependent on hardware and software development practices, but these are outside of Thread's scope. Standards like IEC62443 could be used for assessing the security and maturity levels of hardware and software vendors.

5) Tamper-proofness: Physical access to Thread devices does not allow an adversary to get into the Thread network, as he can neither replay nor forge messages, nor can he obtain the network secret from sniffed traffic, which makes a component replacement or impersonation attack easily detectable. Furthermore, capturing a Thread node has limited effect on the network because of the mesh topology and the ability to self-heal.

The only concern is that a possible probe of a Thread device may disclose the network master key and the PSKc, which are stored in the non-volatile memory of every Thread node and are used for device re-join in case of temporary power outage (Attack T/1). Such a probe can violate *Confidentiality* and *Integrity*, fetching data at rest and in use, and monitoring data in transit. Based on the probed network secret, other types of attacks can also be performed. Thus, finding a trade-off between security and performance requires new solutions to be developed.

TABLE I
ALGORITHMS USED IN THREAD DTLS PROCEDURE

Algorithm	Operation times	Usage
ECC modular multiplication	38 total, 19 each side	Public key generation
SHA-256	16 total, 8 each side	Signature generation
AES-128	Many	Message encryption

TABLE II
JOINING HANDSHAKE TIME OF THREAD REED DEVICE

Node Number	Min. Time	Max. Time	Avg. Time
1	9.83 s	9.91 s	9.87 s
2	95.18 s	176.86 s	124.12 s
3	185.01 s	469.64 s	299.38 s

V. VALIDATION AND COUNTERMEASURES

In this section, we validate our analysis of Thread with experimental results and we propose enhancements that can improve Thread's security.

A. Handshake Flooding (Attack C/2)

We first present a handshake flooding attack against Thread commissioning based on experiments with a NXP FRDM-KW24D512 board (Attack C/2). We start with choosing one node as Thread Leader to create a Thread network, and we then let different amounts of Router Eligible End Devices (REED) to join the network simultaneously. With the help of a USB-KW24D sniffer and Wireshark software, we capture the packets, monitor the whole commissioning procedure and measure the handshake time. The results shown in Table II show that multiple nodes trying to initialize handshake with the Joiner Router increases the joining time of valid nodes significantly, degrading the performance and affecting the availability of the Joiner Router's service. Even though all the devices possess a correct joining credential, handshake flooding still causes many errors resulting in terminated handshakes, and indicates that a DoS attack is achievable through handshake flooding.

The handshake flooding attack is caused by the high complexity of the J-PAKE protocol compared to other PAKE protocols [26]. Table I shows the operations performed during the Thread DTLS handshake, indicating that elliptic curve cryptography (ECC) and SHA-256 algorithms are executed many times during the handshake, causing significant computational load. Unfortunately, most hardware platforms feature only

TABLE III
ATTACK CLASSIFICATION AND ATTACK IMPACT ON THREAD. SYMBOL " + " MEANS RESISTANCE (NO EFFECT), " − " MEANS VULNERABILITY (DEGRADATION)

	Network Attacks				Device Attacks		Effect on Thread	
Attacks	Spoofing	Tampering	Disclosure	DoS	Software	Physical	Performance	Security
Beacon Request Flooding [35]				√			−	+
Handshake Flooding [28]				√			−	+
Network Sniffing [20]			√				+	+
Man-In-Middle Attack [20]		√					+	+
Offline Dictionary Attack [26]	√						+	+
Online Dictionary Attack [26]	√						+	+
Radio Jamming [32]				√			−	+
Symbol Flipping [32]		√					+	+
Signal Overshadowing [32]		√					+	+
Link layer jamming [31]				√			−	+
Node-specific flooding [31]				√			−	+
Back-off Manipulation [31]				√			−	+
CCA Manipulation [31]				√			−	+
Same Nonce Attack [30]			√				+	+
Replay Attack [31]	√						+	+
Replay-protection Attack [31]				√			+	+
ACK Spoofing [31]	√						−	+
ACK Dropping [31]				√			−	+
GTS-related Attacks [31]	√			√			+	+
PANID Conflict Attack [31]				√			+	+
Bootstrapping Attack [31]				√			+	+
Steganography Attack [31]		√					+	+
Routing Hop Manipulation [33]		√					+	+
Routing Redirection [33]		√					+	+
Black/grey Hole Attack [34]				√			+	+
Worm Hole Attack [34]				√			+	+
Link Spoofing Attack [34]	√						+	+
Configuration Mechanism Abuse [20]					√		+	+
Component Replacement [20]						√	+	+
Microprobing [20]						√	−	−

hardware acceleration for AES to support message encryption, but not for ECC and SHA-256. An effective countermeasure for handshake flooding attacks would be to speed up ECC and SHA-256 computation through hardware acceleration. To our knowledge, the recently launched CRYPTO module can accelerate complex cryptographic functions including ECC, SHA and CCM [36].

Furthermore, by defining a proper relay rate for the Joiner Router, one could also limit the impact of intentional handshake flooding to an acceptable level. Finally, since the handshake is only needed for joining a new device, one could ignore the beacon request and handshake request messages to disable the handshake function, which would also help to avoid this attack.

B. Jamming and Flooding Attacks (Attacks R/1 and R/2)

Table III suggests that jamming and packet flooding could affect the Thread network's performance. Combined with radio jamming, ACK spoofing and ACK dropping attacks can also threaten Thread and reduce the availability of specific devices in a BAS. Back-off and CCA manipulation attacks are obscure but can result in considerable collisions in the network. To counteract jamming and MAC manipulation attacks we recommend to introduce an intrusion detection and prevention system (IDPS) for wireless networks into Thread based BAS.

As an example, in [37] Paria *et al.* provided a solution by monitoring network features such as datagram traffic rate (TR), received signal strength (RSS), packet error rate (PER) and node availability (NA), and comparing them with the normal behavior of the network according to its specification. Intuitively, radio jamming will inevitably increase the PER and decrease NA, once these abnormal features are detected by the IDPS, the system could migrate to another channel using the existing update mechanism in Thread. Such a reactive mitigation scheme would not work for wide band jamming, but that is easier to detect in general. In the case of MAC manipulation illegitimate nodes must have an abnormally high TR and affect PER and the NA parameters. Besides, checking frame authentication information is also helpful for attack detection, since malicious nodes do not have the network keys, and thus the datagrams are unreadable to the IDPS. The countermeasures to MAC manipulation include changing channels and localizing malicious nodes based on RSS values, as mentioned in [38].

C. Secure Leaving (Attack L/1)

For devices that are forced to leave the network we propose a mechanism that relies on CoAP based Thread management commands. The mechanism is based on a *network leave request* command defined and ordered by the Thread commissioner. On receiving this request, a Thread device has to erase all the network credentials from its non-volatile memory and has to reset itself to un-commissioned state, which effectively prevents an adversary to acquire network security material from from the device. Besides, for the case that a leaving device does not appropriately erase the network credentials

TABLE IV
OVERVIEW OF THREAD VULNERABILITIES

Vulnerability	Phase of Interaction	Violated Requirement	Criticality	Effect on Thread Network	Mitigating Solution	Complexity of solution	Suggestion for Future Standard
Microprobing	Tamper-proofness	Integrity Confidentiality	*****	Leakage of network credential	Resistive RAM for key storage	Low	Use temporarily generated PSKc; and/or limit the storage of PSKc to necessary nodes; enable DTLS session between paired devices.
Insecure Leaving	Leaving	Confidentiality	****	Potential leakage of network credential	CoAP based command forcing key erasure	Low	Explicitly specify secure leaving procedure, ensure removal of network credentials.
Beacon Request Flooding	Commissioning	Availability	***	Degrade network performance, increase power consumption	Limit beacon response rate	Low	Set proper beacon response rate; disable commissioning after installation.
Handshake Flooding	Commissioning	Availability	***	Degrade network performance, increase power consumption	Hardware accelerator for cryptographic functions	Low	Set proper relay rate; disable commissioning after installation.
Radio Jamming	Communication	Availability	**	Degrade network performance	Intrusion detection and prevention system (IDPS) for network monitoring	High	N/A
Link layer jamming	Communication	Availability	**	Degrade network performance	IDPS for network monitoring	High	N/A
Node-specific flooding	Communication	Availability	**	Degrade network performance, increase power consumption	IDPS for network monitoring	High	N/A
ACK Spoofing	Communication	Authenticity	**	Degrade network performance	IDPS for network monitoring	High	N/A
ACK Dropping	Communication	Availability	**	Degrade network performance	IDPS for network monitoring	High	N/A
Back-off Manipulation	Communication	Availability	*	Degrade network performance	IDPS for network monitoring	High	N/A
CCA Manipulation	Communication	Availability	*	Degrade network performance	IDPS for network monitoring	High	N/A

upon leaving, the request should be followed by the update of the network security material using the readily available key replacement management command. By combining these mechanisms Thread would be able to ensure secure leaving.

D. Key Compromise (Attack T/1)

Thread's key storage is a weak point to a BAS. Storing the PSKc and master key in every device need to be carefully secured due to the risk of key compromise by probing the device. Considering the rare usage of PSKc, one solution could be to take advantage of the random number generator inside the Border Router for generating a temporal PSKc with the format of a QR code or serial number. Customers can easily scan the PSKc and start to commission the external commissioner. After the commissioning procedure, the temporal PSKc can safely be destroyed immediately, and need not be stored. For the master key storage physical protection on the chip is crucial to the security of the BAS. Techniques such as read prevention and copy prevention have been developed and adopted by many chip manufacturers and should be applied to Thread devices' non-volatile memory. In [39] Xie *et al.* implemented a logic resistive random access memory (RRAM) chip for physically secure key storage, which was shown to resist deprocessing, microscopy observation, side-channel attacks, malicious writing and data interception attacks and could be an option to secure Thread master key. Another concern is PSKd, as it is usually exposed on devices' labels without any protection, thus necessary physical access control needs to be ensured. An installer should avoid eavesdropping during the commissioning of devices, in addition it should be verified that the joined device is the one that is being commissioned. A more secure way is to mark the device's location and keep its label with sensitive information in a secure place with strict physical access control.

As an adversary with the compromised network master key can easily synchronize network data from other Routers so as to monitor traffic, impersonate nodes, tamper routing data and paths and even commission new malicious devices, we suggest paired sensor and actuator devices establish a DTLS session to limit the kind of attacks made possible by a compromised node. All of the aforementioned vulnerabilities to Thread and possible mitigations are listed in Table IV.

VI. CONCLUSION AND FUTURE WORK

In this paper, we argue that improved security is a key driver for adopting native-IP BAS protocols. We proposed a security assessment taxonomy and applied it to Thread, which is an emerging native IP-based BAS communication protocol. Our assessment shows that Thread is more secure than traditional non-IP based BAS solutions, and is only

vulnerable to a few novel attack techniques. As an extension to the Thread protocol, we proposed several enhancements that could strengthen Thread's security and improve network performance and reliability. It is subject of future work to implement and experimentally validate these solutions. We believe that a security assessment using the proposed taxonomy should be performed for recently proposed security extensions of existing protocols, such as KNX, BACnet, and LONworks.

VII. Acknowledgment

This research was partly funded by Vinnova (Swedish Innovation Agency) and Norrköping Fund for Research and Development in Sweden, the authors are very grateful to their support. Dán was partly supported by MSB through the Cerces project.

References

[1] "Commercial Building Automation," in *From Machine-To-Machine to the Internet of Things*, J. Hller, , V. Tsiatsis, , C. Mulligan, , S. Karnouskos, , S. Avesand, , and D. Boyle, Eds. Oxford: Academic Press, 2014, pp. 269 – 279.

[2] T. Radichel, "Case study: Critical controls that could have prevented Target breach," SANS Institute InfoSec Reading Room, 2014.

[3] C. Gomez and J. Paradells, "Wireless home automation networks: A survey of architectures and technologies," *IEEE Communications Magazine*, vol. 48, no. 6, pp. 92–101, June 2010.

[4] F. J. Bellido-Outeirino, J. M. Flores-Arias, F. Domingo-Perez, A. G. de Castro, and A. Moreno-Munoz, "Building lighting automation through the integration of dali with wireless sensor networks," *IEEE Transactions on Consumer Electronics*, vol. 58, no. 1, pp. 47–52, February 2012.

[5] "Thread specification 1.1.1," Specification, Thread Group, February 2017.

[6] T. Mundt and P. Wickboldt, "Security in building automation systems - a first analysis," in *Proc. of Int. Conf. On Cyber Security And Protection Of Digital Services*, June 2016, pp. 1–8.

[7] A. Brauchli and D. Li, "A solution based analysis of attack vectors on smart home systems," in *Proc. of Int. Conf. on Cyber Security of Smart Cities, Industrial Control System and Communications (SSIC)*, Aug 2015, pp. 1–6.

[8] M. A. B. Karnain and Z. B. Zakaria, "A review on zigbee security enhancement in smart home environment," in *Proc. of Int. Conf. on Information Science and Security (ICISS)*, Dec 2015, pp. 1–4.

[9] L. Coppolino, V. DAlessandro, S. DAntonio, L. Levy, and L. Romano, "My smart home is under attack," in *IEEE 18th International Conference on Computational Science and Engineering (CSE)*, Oct 2015, pp. 145–151.

[10] X. Cao, D. M. Shila, Y. Cheng, Z. Yang, Y. Zhou, and J. Chen, "Ghost-in-zigbee: Energy depletion attack on zigbee-based wireless networks," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 816–829, Oct 2016.

[11] T. Goodspeed, "Extracting keys from second generation zigbee chips," presented at the Black Hat USA Conference, 2009.

[12] Y. S. Shiu, S. Y. Chang, H. C. Wu, S. C. H. Huang, and H. H. Chen, "Physical layer security in wireless networks: a tutorial," *IEEE Wireless Communications*, vol. 18, no. 2, pp. 66–74, April 2011.

[13] C. Alcaraz and J. Lopez, "A security analysis for wireless sensor mesh networks in highly critical systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 40, no. 4, pp. 419–428, July 2010.

[14] S. Marksteiner, V. J. E. Jimenez, H. Valiant, and H. Zeiner, "An overview of wireless iot protocol security in the smart home domain," in *2017 Internet of Things Business Models, Users, and Networks*, Nov 2017, pp. 1–8.

[15] Rajkumar, V. B. A, G. Rajaraman, and D. H. G. Chandrakanth, "Security attacks and its countermeasures in wireless sensor networks," *International Journal of Engineering Research and Applications*, vol. 4, pp. 05–15, Oct 2014.

[16] I. Tomi and J. A. McCann, "A survey of potential security issues in existing wireless sensor network protocols," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1910–1923, Dec 2017.

[17] J. Granjal, E. Monteiro, and J. S. Silva, "Security for the internet of things: A survey of existing protocols and open research issues," *IEEE Communications Surveys Tutorials*, vol. 17, no. 3, pp. 1294–1312, 2015.

[18] J. M. Batalla, A. Vasilakos, and M. Gajewski, "Secure smart homes: Opportunities and challenges," *ACM Comput. Surv.*, vol. 50, no. 5, pp. 75:1–75:32, Sep. 2017.

[19] A. Mosenia and N. K. Jha, "A comprehensive study of security of internet-of-things," *IEEE Transactions on Emerging Topics in Computing*, vol. 5, no. 4, pp. 586–602, Oct 2017.

[20] W. Granzer, F. Praus, and W. Kastner, "Security in building automation systems," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 11, pp. 3622–3630, Nov 2010.

[21] W. Granzer and W. Kastner, "Security analysis of open building automation systems," in *Proc. of SAFECOMP*, 2010, pp. 303–316.

[22] W. Granzer, W. Kastner, G. Neugschwandtner, and F. Praus, "Security in networked building automation systems," in *IEEE International Workshop on Factory Communication Systems*, 2006, pp. 283–292.

[23] K. Islam, W. Shen, and X. Wang, "Security and privacy considerations for wireless sensor networks in smart home environments," in *Proc. of Int. Conf. on Computer Supported Cooperative Work in Design (CSCWD)*, May 2012, pp. 626–633.

[24] R. Kelsey, "Mesh link establishment," IETF Draft, 2014.

[25] F. Hao and P. Zieliński, *A 2-Round Anonymous Veto Protocol*. Springer Berlin Heidelberg, 2009, pp. 202–211.

[26] M. Abdalla, F. Benhamouda, and P. MacKenzie, "Security of the j-pake password-authenticated key exchange protocol," in *Proc. of IEEE Symposium on Security and Privacy*, May 2015, pp. 571–587.

[27] "Thread commissioning white paper," White Paper, Thread Group, July 2015.

[28] M. Tiloca, C. Gehrmann, and L. Seitz, "On improving resistance to denial of service and key provisioning scalability of the DTLS handshake," *International Journal of Information Security*, vol. 16, no. 2, pp. 173–193, Apr 2017.

[29] *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPANs)*, IEEE Std. 802.15.4, 2006.

[30] N. Sastry and D. Wagner, "Security considerations for ieee 802.15.4 networks," in *Proc. of ACM Workshop on Wireless Security*, 2004, pp. 32–42.

[31] Y. M. Amin and A. T. Abdel-Hamid, "Classification and analysis of ieee 802.15.4 mac layer attacks," in *Proc. of Int. Conf. on Innovations in Information Technology (IIT)*, Nov 2015, pp. 74–79.

[32] ——, "Classification and analysis of ieee 802.15.4 phy layer attacks," in *Proc. of Int. Conf. on Selected Topics in Mobile Wireless Networking (MoWNeT)*, April 2016, pp. 1–8.

[33] "Secure Routing Protocols for Mobile AdHoc Wireless Networks," in *Advanced Wired and Wireless Networks*, 1st ed., B. J. W. Tadeusz A. Wysocki, Arek Dadej, Ed. Springer, 2006, pp. 57 – 80.

[34] B. Kannhavong, H. Nakayama, Y. Nemoto, N. Kato, and A. Jamalipour, "A survey of routing attacks in mobile ad hoc networks," *IEEE Wireless Communications*, vol. 14, no. 5, pp. 85–91, Oct. 2007.

[35] D. Ratnayake, H. Kazemian, and S. Yusuf, "Identification of probe request attacks in WLANs using neural networks," *Neural Computing & Applications*, vol. 25, no. 1, pp. 1 – 14, 2014.

[36] *AN0955: CRYPTO*, Silicon Labs, 2016. [Online]. Available: https://www.silabs.com/documents/public/application-notes/AN0955.pdf

[37] P. Jokar and V. Leung, "Intrusion detection and prevention for zigbee-based home area networks in smart grids," *IEEE Transactions on Smart Grid*, vol. 9, no. 3, pp. 1800–1811, May 2018.

[38] M. Maheshwari, S. Ananthanarayanan, A. Banerjee, N. Patwari, and S. K. Kasera, "Detecting malicious nodes in rss-based localization," in *Proc. of IEEE Int. Symp. on World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, June 2011, pp. 1–6.

[39] Y. Xie, X. Xue, J. Yang, Y. Lin, Q. Zou, R. Huang, and J. Wu, "A logic resistive memory chip for embedded key storage with physical security," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 63, no. 4, pp. 336–340, April 2016.

Yu Liu received B.Eng. degree in electronics science and technology from Harbin Institute of Technology (HIT), Weihai, China, in 2014, M.Sc. degree in computer science from University of Trento, Trento, Italy, in 2016, M.Sc. degree in innovation in information and communication technology from Technical University of Berlin, Berlin, Germany, in 2017. He is currently working toward the Ph.D. degree with the department of science and technology in Linköping University. He was a thesis student with ABB Corporate Research Center, Västerås, Sweden from April to November 2016. His research interests include cloud based Internet of Things (IoT) solution, embedded systems and wireless sensor networks.

Zhibo Pang (M13, SM15), received MBA from University of Turku, Turku, Finland in 2012, and PhD from the Royal Institute of Technology (KTH), Stockholm, Sweden in 2013. He is currently a Principal Scientist on Wireless Communications at ABB Corporate Research, Västerås, Sweden. He is also serving as Adjunct Professor or similar roles at Royal Institute of Technology (KTH), Tsinghua University, University of Sydney, Zhejiang University, and Beijing University of Posts and Telecommunications. Before joined ABB, he was co-founder and CTO of startups such as Ambigua Medito AB. He is a Senior Member of IEEE and Co-Chair of the Technical Committee on Industrial Informatics. He is taking editorial roles of IEEE Proceedings, IEEE Transactions on Industrial Informatics, IEEE Journal of Biomedical and Health Informatics, and IEEE Reviews in Biomedical Engineering, IEEE Access, Journal of Management Analytics (Taylor & Francis), and Journal of Industrial Information Integration (Elsevier). He was the General Chair of IEEE ES2017. He was awarded the 2016 Inventor of the Year Award by ABB Corporate Research Sweden. His current research interests include the Industry4.0, health engineering, real-time cyber physical systems, Internet-of-Things, wireless control network, industrial communication, real time embedded system, high accuracy localization and navigation, enterprise systems, automation and robotics, multicore system-on-chip and network-on-chip. He also works on the business-technology joint research such as strategy, business model, value chain, and entrepreneurship and intrepreneurship.

György Dán (M07, SM17) is Professor of Teletraffic Systems at KTH Royal Institute of Technology, Stockholm, Sweden. He received the M.Sc. in computer engineering from the Budapest University of Technology and Economics, Hungary in 1999, the M.Sc. in business administration from the Corvinus University of Budapest, Hungary in 2003, and the Ph.D. in Telecommunications from KTH in 2006. He worked as a consultant in the field of access networks, streaming media and videoconferencing 1999-2001. He was a visiting researcher at the Swedish Institute of Computer Science in 2008, a Fulbright research scholar at University of Illinois at Urbana-Champaign in 2012-2013, and an invited professor at EPFL in 2014-2015. He has been an area editor of Computer Communications since 2014. His research interests include the design and analysis of content management and computing systems, game theoretical models of networked systems, and cyber-physical system security and resilience.

Dapeng Lan is a PhD Research Fellow in the Department of Informantics in University of Oslo, Norway on Fog Computing. He received B.Eng. degree in Microelectronics from Sun Yat-sen University (SYSU) Guangzhou, China, in 2014, M.Sc. degree in ICT Innovation from KTH Royal Institute of Technology, Stockholm, Sweden, in 2016, M.Sc. degree in innovation in information and communication technology from Technical University of Berlin, Berlin, Germany, in 2017. In 2017, he worked in a company called InnoEnergy about smart building energy management in Sweden. He was a thesis student with ABB Corporate Research Center, Västerås, Sweden from January to September 2016. His research interests include Fog computing, Internet of Things, and Distributed systems.

Prof. Shaofang Gong received his B.Sc. degree in microelectronics from Fudan University, China in 1982, and Licentiate of Engineering and Ph.D. degrees from Linköping University, Sweden in 1988 and 1990, respectively. Between 1991 and 1999, he was a senior researcher with the research institute RISE Acreo. From 2000 to 2001, he was the chief technology officer at a spin-off company from the research institute. In the meantime, he had an adjunct professorship at Linköping University. Since 2002, he has been chair professor of Communication Electronics at Linköping University. His main research interest has been communication electronics including radio frequency and microwave system design, high speed data transmissions and wireless sensor networks towards Internet of Things.