# Ghost-in-ZigBee: Energy Depletion Attack on ZigBee-Based Wireless Networks

Xianghui Cao, *Member, IEEE*, Devu Manikantan Shila, *Student Member, IEEE*, Yu Cheng, *Senior Member, IEEE*, Zequ Yang, Yang Zhou, *Student Member, IEEE*, and Jiming Chen, *Senior Member, IEEE*

***Abstract*—ZigBee has been widely recognized as an important enabling technique for Internet of Things (IoT). However, the ZigBee nodes are normally resource-limited, making the network susceptible to a variety of security threats. This paper closely investigates a severe attack on ZigBee networks termed as *ghost*, which leverages the underlying vulnerabilities of the IEEE 802.15.4 security suites to deplete the energy of the nodes. We show that the impact of *ghost* is very large and that it can facilitate a variety of threats including denial of service and replay attacks. We highlight that merely deploying a standard suite of advanced security techniques does not necessarily guarantee improved security, but instead might be leveraged by adversaries to cause severe disruption in the network. We propose several recommendations on how to localize and withstand the *ghost* and other related attacks in ZigBee networks. Extensive simulations are provided to show the impact of the *ghost* and the performance of the proposed recommendations. Moreover, physical experiments also have been conducted and the observations confirm the severity of the impact by the *ghost* attack. We believe that the presented work will aid the researchers to improve the security of ZigBee further.**

***Index Terms*—Countermeasures, energy depletion attack, experiments, security, ZigBee.**

## I. Introduction

THE EMERGING Internet of Things (IoT) will facilitate our life with ubiquitous sensing, distributed computing, self-organized networking, and efficient interaction with the physical world [1]–[4]. Among many enabling protocols for IoT applications, IEEE 802.15.4-based ZigBee has been recently drawing a lot of attention and has become a most popular IoT solution for its expandability, low cost, ease of use, and minimal maintenance. The ZigBee alliance has essentially targeted their efforts on building a global wireless language for myriad of everyday devices such as light switches, thermostats, smart devices, remote controls, as well as more complex sensor devices found abundantly in the health care, commercial building, and industrial automation sectors [2], [5].

Security is a critical concern in many IoT applications [1], [6], [7]. The IEEE 802.15.4 standard addresses the security requirements through a medium access control (MAC) layer package, providing fundamental security services ranging from data confidentiality, data integrity to replay protection [8]. Despite these basic services, a number of security problems and pitfalls, especially pertaining to the initialization vector management, key management, and integrity protection, have been outlined in [9]. More attacks on the physical and MAC sublayers, including jamming, capture, and tampering, exhaustion, collision, and unfairness have been presented in [10]. Besides, nowadays, off-the-shelf attack toolkits like KillerBee [11] are available that can be leveraged even by a novice adversary to explore and exploit the security of ZigBee networks. Using KillerBee and an IEEE 802.15.4 compatible radio interface, an adversary can carry out several attacks ranging from surreptitious eavesdropping to traffic injection with a little or no effort.

Markedly, people need to have a solid understanding of the ZigBee security performance before positioning it as a major player in the market of IoT. In this paper, we investigate a potential flaw related to sending security headers in clear text. These security headers are treated as critical parameters to provide semantic security and replay protection. A key question is: *what might be the consequences if an attacker masquerades as a trusted node by crafting the security headers?* IEEE 802.15.4 provides protection to integrity related attacks by including a cryptographically secure checksum [a.k.a. message integrity code (MIC)] with each message. When an adversary crafts an invalid security header without knowing the key generating the MIC, although the integrity attack fails, the recipient node in fact expends energy for receiving, and processing those bogus messages. In particular, it is shown that the security computing energy can be far from ignorable [12], [13]. If an intelligent adversary sends a number of such crafted messages, a significant amount of energy will be used leading to battery depletion of the victim node.

This paper investigates a severe attack termed as *ghost-in-ZigBee* (aka *ghost*) on commercial ZigBee networks for IoT applications, in which an attacker constructs bogus messages to lure a node to do superfluous security-related computations to intentionally deplete that node's energy. The aftermath of the attack is perilous as it will significantly cut back the lifetime of the victim node and further facilitate an adversary to execute a

Manuscript received November 08, 2015; revised December 22, 2015; accepted December 23, 2015. Date of publication January 08, 2016; date of current version September 08, 2016. This work was supported in part by the National Science Foundation under Grant CNS-1053777, Grant CNS-1117687, and Grant CNS-1320736, and in part by the National Natural Science Foundation of China under Grant U1401253, Grant 61203036, and Grant 61573103.

X. Cao is with the School of Automation, Southeast University, Nanjing 210096, China (e-mail: xhcao@seu.edu.cn).

D. M. Shila is with the United Technologies Research Center, Hartford, CT 06108 USA (e-mail: manikad@utrc.utc.com).

Y. Cheng is with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL 60616 USA (e-mail: cheng@iit.edu).

Z. Yang, Y. Zhou, and J. Chen are with the Department of Control, Zhejiang University, Hangzhou 310027, China (e-mail: yzqcjdtc@gmail.com; zhouyang0807@gmail.com; jmchen@iipc.zju.edu.cn).

Digital Object Identifier 10.1109/JIOT.2016.2516102

variety of after-depletion threats like denial of service (DoS), replay attack, and loss of confidentiality. In the literature, the general class of resource depletion attacks have been studied in different scenarios [14]–[21]. Particularly, there have been studies on crypto-related resource depletion attacks and corresponding countermeasures based on centralized scheduling, key management, or preauthentication [20]–[23]. However, those countermeasures either do not apply in distributed ZigBee networks or cannot defend against *ghost*. For instance, *ghost* can be launched without knowing the keys. Also, the preauthentication scheme is effective only for ZigBee applications where the traffic from normal nodes has certain patterns. Despite these studies, how to execute such attacks over ZigBee has not been systematically studied, and the seriousness of energy depletion attack has not been quantitatively demonstrated before.

For wireless networks, security designers are often concerned about classical security threats such as snooping, replay, and spoofing attacks, and tend to use security protocols to defend against these attacks. In this paper, however, by dissecting the security protocol, we found that by intelligently crafting the packets with larger frame numbers, one could launch *ghost* attacks and consequently replay and confidentiality attacks, without getting caught. We highlight the fact that merely deploying standard advanced security protocols does not necessarily ensure improved security, but instead might be leveraged by attackers to cause severe disruption in the network. We believe that our work in this paper will aid researchers to further improve the security posture of energy-constrained wireless networks.

Our major contributions can be summarized as follows.

1) We investigate a severe attack on ZigBee-based IoT networks, termed as *ghost*, which exploits the underlying vulnerabilities of the IEEE 802.15.4 security suites to cause intentional energy failure of nodes. Besides energy depletion, it can induce DoS attack and other postdepletion attacks.
2) We theoretically analyze the impact of the attack on a victim node's lifetime and develop an analytical model to quantify the impact of the induced DoS attack over a multihop network.
3) We propose a three-phase algorithm to detect and localize the attacker by analyzing the network flow variations. We also discuss several recommendations on how to withstand the *ghost* and its related attacks in ZigBee networks.
4) Extensive computer simulation results demonstrate the impact of the *ghost* and the efficiency of the proposed countermeasures. Moreover, we validate the effectiveness of the *ghost* with physical experiments.

This paper is organized as follows. Section II reviews more related work and Section III overviews IEEE 802.15.4 security architecture. Section IV presents the *ghost* attack and its induced attacks and theoretically analyzes their impacts. Corresponding countermeasures are discussed in Section V. Extensive simulation and physical experimental results are presented in Sections VI and VII, respectively. Finally, Section VIII concludes this paper.

## II. Related Work

The proposed attack belongs to a more general class of energy depletion (or more broadly resource depletion) attacks. However, how to execute such attacks on different protocols or in different application scenarios is quite diverse, and there is not a one-for-all solution for all such attacks. Below, we give a brief review of resource depletion attacks and existing defense strategies in wireless networks.

In physical layer, an adversary can easily launch jamming attacks by sending abundant signals to deny legitimate nodes' access to the channel resource. Meanwhile, jamming attacks can rapidly drain nodes batteries [14]. Basically, jamming attacks are difficult to completely defend, but an effective way for legitimate nodes to mitigate their impact is to sleep and (periodically) wake up to save energy [14]. In upper layers, resource depletion attacks are also possible. For example, in *ad hoc* sensors networks, an adversary can drain the energy of sensors by deliberately sending messages to construct artificial routing paths or introduce loops to the routing process of benign sensors [18]. To mitigate the attack's impact, a modified routing protocol was proposed which forces the packets to always move toward the destination.

Most existing resource depletion attacks are executed at the MAC layer. An adversary can intentionally broadcast requests to others to prevent them from entering sleep mode and deplete their energy faster, so as to cause sleep deprivation attack [15]. For such kind of attack, a number of detection techniques have been proposed, referring to a survey in [16]. In networks such as WLANs, adversaries can manipulate their MAC protocol parameters (e.g., contention window size) or completely disobey the rules to gain unfair share of the channel resource over others [19], [24], [25]. To determine whether a node is misbehaving, many existing methods focus on either throughput analysis [19], [24] in single-hop networks or behavior monitoring by neighbors [25] in both single-hop and *ad hoc* networks. A smart adversary can monitor the channel and send packets to cause collision attack to ongoing transmissions. To detect such an attack, the work in [17] proposes to deploy trusted monitors which apply a sequential detection rule to determine the existence of the attacker based on measurements of collision probabilities.

Adversaries can launch resource depletion attacks by leveraging the security aspects of network protocols. In [21], a "denial-of-sleep" attack was studied, in which an adversary broadcasts unauthenticated packets to drain receivers' packet authentication energy. To prevent such an attack, a G-MAC was proposed in which the broadcasting activities are controlled by a central gateway node so that average nodes will not be suffered from the attack. However, the G-MAC is based on a centralized structure where all the traffic of the nodes is controlled by the gateway, making it difficult to implement in ZigBee networks.

Smart adversaries may attack the protocols in a deeper level. For instance, a temporal key integrity protocol (TKIP) MIC attacker on IEEE 802.11 networks decodes the payload 1 byte at a time by using multiple replays and observing the response over the air on MIC failures [26]. TCP SYN attack can send

TABLE I
SECURITY SUITES IN IEEE 802.15.4

| Security level/Id | Security suite | Confidentiality | Integrity |
|---|---|---|---|
| 000 | None | ✗ | ✗ |
| 001 | AES-CBC-MAC-32 | ✗ | ✓ |
| 010 | AES-CBC-MAC-64 | ✗ | ✓ |
| 011 | AES-CBC-MAC-128 | ✗ | ✓ |
| 100 | AES-CTR | ✓ | ✗ |
| 101 | AES-CCM-32 | ✓ | ✓ |
| 110 | AES-CCM-64 | ✓ | ✓ |
| 111 | AES-CCM-128 | ✓ | ✓ |

a chain of SYN requests to a victim system in an attempt to consume enough server resources and launch DoS attack [27]. Adversaries can also broadcast forge packets to force others to perform unnecessary signature verifications [22]. These attacks thwart the legitimate nodes from using the medium and are able to simultaneously consume large amounts of victim nodes' energy in crypto-related operations, referring to [20]. Defending such attacks may need to exploit the security protocols. For example, in [22], a group-key-based and a key-chain-based approaches were proposed which basically filter the forged packets by preventing the attacker from having the keys. Although these approaches are effective to defend against DoS induced by the signature-based attack, energy depletion caused by the attack are still difficult to mitigate. Against a similar attack with which adversaries repeatedly send connection requests to exhaust the energy of implantable medical nodes due to performing a large number of authentications [23], a preauthentication scheme is developed which filters packets that do not match the access pattern of a patient.

## III. SECURITY ARCHITECTURE

In this section, we focus on the security services provided by the IEEE 802.15.4 MAC layer [8]. The standard particularly provides four basic security services for use by the higher layer applications: 1) access control; 2) data integrity; 3) data confidentiality; and 4) sequential freshness for replay protection, each of which is described briefly in the following. The MAC layer is responsible for providing security services on specified incoming and outgoing frames when requested to do so by the higher layers. The higher layer (i.e., the application layer) indicates its choice of the security suite by setting the *security control field* in the *auxiliary security header* of the message, which identifies eight candidate security levels as shown in Table I. The security level configuration can be adjusted message by message.

### A. Security Services

*1) Access Control:* The IEEE 802.15.4 MAC layer protocol prevents unauthorized nodes from participating in the network by maintaining a valid nodes list, commonly known as access control list (ACL). For each incoming message, the receiving node checks the source address against the list of valid addresses in the table. If there is a match, the message is either accepted or forwarded to the next hop, otherwise it is dropped. Although such an access control mechanism can keep out the unauthorized parties from participating in the network, a number of issues emerge such as the spoofing attacks where an adversary masquerade as a valid user through crafting messages (e.g., source address) to bypass the ACL checks.

*2) Data Integrity:* The standard resolves data integrity issue by including a MIC, which is computed by applying a hash function over the message and preshared secret key (aka the symmetric key) [28]. A receiver can validate the integrity by checking whether the received MIC tag can be regenerated using the same hash function, the shared symmetric key, and the received message. If positive, the integrity is considered as maintained, i.e., both the message and the MIC tag were not modified. The standard provides data integrity services through AES-CBC-MAC and AES-CCM with three possible lengths of the MIC tag, i.e., 32, 64, or 128 bits.

*3) Data Confidentiality:* Underpinning the goal of confidentiality are the encryption schemes. Besides data encryption, the semantic security is needed to ensure that the attacker cannot learn even the partial information about the messages that have been encrypted. A common approach to realizing semantic security is to leverage a unique *nonce*, typically a counter or random value, for each invocation of the encryption algorithm. The main purpose of a nonce is to add discrepancy to the encryption process. Since the receiver should rely on the same nonce to decrypt the messages, nonces are typically sent in the same message with the encrypted data in plain text, without keeping it secret. *This work is concerned with the attackers manipulating those nonces sent in plain text.* To provide semantic security, both AES-CTR and AES-CCM use a 13-B nonce, which consists of source address and a counter field.

*4) Sequential Freshness for Replay Protection:* Although data confidentiality and data integrity can prevent the network from a variety of known threats such as eavesdropping and spoofing, these schemes cannot protect the network from replay attacks. With the IEEE 802.15.4 specification, the sender usually assigns a monotonically increasing frame counter to each message and the receiver rejects those messages with smaller sequence numbers than it has already seen. The efficiency of this scheme clearly depends on the amount of time it will take the frame counters to roll over. A 32-bit counter is used in the IEEE 802.15.4 such that an adversary can carry out a replay attack only after $2^{32}$ frames, which is considered cryptographically secure in practice. In addition to replay protection, the frame counters are considered as an important input to the construction of nonces for providing semantic security.

### B. Security Suites

ZigBee uses AES-based [29] security suites to provide fundamental security services like confidentiality, integrity and replay protection.

*1) AES-CTR for Encryption:* Messages in this mode are encrypted and decrypted by XORing with the keystream

produced by the AES encrypting sequential counter block values. Let $O = \{O_1, O_2, \ldots, O_n\}$ denote the output keystream block. To encrypt a payload with AES-CTR, the encryptor partitions the plaintext $P$ into $n$ 128-bit blocks; in case the last block is less than 128 bits, zeros are padded to it. Each $P$ block is then XORed with a block of the keystream $O$ to generate the corresponding ciphertext. To avoid reuse of the same output stream $O$, AES-CTR requires the encryptor to generate a unique keystream per-block per message. The decryption process is similar to encryption process.

*2) AES-CBC-MAC for Authentication:* A message is authenticated by splitting the input $I$ into $n$ 128-bit blocks, with necessary padding. Let $I$ and $O$ denote the input and its corresponding output block, respectively. The CBC-MAC mode is then defined as: $O = C(k)[I]$ and $C(k)[I]$ is the invocation of AES algorithm on the input block using the secret key $k$.

*3) AES-CCM for Encryption and Authentication:* This mode consists of two steps: a) computing the MIC tag using CBC-MAC; and b) encrypting the message concatenated with the MIC tag using CTR mode.

## IV. GHOST-IN-ZIGBEE

Consider a static multihop ZigBee-based IoT network where one or more nodes serve as the coordinators (or gateways) and provide services for the entire ZigBee network. The cryptographic mechanism presented in this work is based on the symmetric-key cryptography and uses keys that are provided by higher layer processes. We assume that the keys are well established and maintained across the network [30]. As stated in the standard, we assume a secure implementation of the cryptographic operations, and secure and authentic storage of the keying material. For ease of analyzing the attacker's impact, we assume that the legitimate nodes work in the nonbeacon-enabled mode by setting both of the parameters macBeaconOrder and macSuperframeOrder to 15 [8]. In this case, they apply the unslotted CSMA/CA protocol for channel access.

In such an environment, we assume the presence of one or multiple *ghost* attackers, equipped with IEEE 802.15.4 compatible radios. As opposed to legitimate nodes, the attackers have no power or memory constraints. We assume a three-phase attack model: 1) *preattack phase* in which the attacker learns about the network by surreptitiously eavesdropping the messages; 2) *attack phase* in which the attacker leverages the learned information to execute the *ghost* attack; and 3) *postattack/depletion phase* in which, once the energy of nodes are depleted, the attacker executes several other attacks such as replay attack or confidentiality attack, to be discussed in Section IV-C.

### A. Preattack Phase

In this phase, the attacker can learn from the overheard messages the information of sender–receiver pairs. For example, since the counter is communicated in plain text, the attacker can learn the sender's address and generate bogus messages with that address. On the other hand, although the proposed attack does not require the knowledge of the keys or the traffic information, the attacker can leverage information learned in the preattack phase to enhance its impact on the network. For example, for a target node in duty-cycling mode, if the attacker learns the cycles by eavesdropping the target node's messages, it can launch attack during the target node's active periods and cause significant impact on that node's energy (to be analyzed in the following).

### B. Attack Phase

The security suites depend on the encryptor to generate a unique keystream per message to provide semantic security. This task is accomplished by leveraging a 16-B unique counter constructed from the fields in the message. The counter consists of a 2-B static flags field, a 13-B nonce field (comprising the sender address, the frame counter, and the security level field), and a 1-B block counter that numbers the 16-B blocks within the message. In addition to semantic security, the frame counters are used by the security suite to enable replay protection as well.

In the IEEE 802.15.4 protocol, those message fields used to construct the 16-B counter need to be communicated in plain text by the sender. Suppose that an adversary injected bogus messages constructed using increasing frame counters in the nonce fields. According to the standard, the recipient node compares the frame counter seen in the incoming packet to the highest value stored in the ACL table. If the incoming packet has a larger counter value than the stored one, the message is accepted for further processing and the new counter is used to update the ACL table; otherwise, the message is rejected. Moreover, although the bogus messages will not pass the integrity check, the recipient node in fact expends a valuable amount of energy accepting and processing those bogus messages. Therefore, if an intelligent attacker sends a number of such crafted bogus messages to lure the receiver to do the superfluous security-related computations, not only the messages from the victim nodes may be rejected but also a significant amount of energy will be spent, leading to battery depletion of the recipient. We therefore term this attack as *ghost-in-ZigBee*.

Notice that an adversary can easily launch *ghost* attack without knowing the network keys or network traffic information; thus, it can escape from being caught by existing key management and preauthentication schemes [22], [23].

*1) Analysis:* We present an analytical model to quantify the impact of a *ghost* attacker on the lifetime of a victim node (denoted as $\mathcal{D}$). We consider that the victim node works in a duty-cycling mode, which is a common working mode for low-cost ZigBee networks such as wireless sensor networks. Duty cycling can be enabled by using the IEEE 802.15.4 beacon-enabled mode, where a superframe with active and inactive portions can be viewed as a duty cycle [8]. Also, duty cycling can be controlled by upper layers in the nonbeacon-enabled mode [31], e.g., the upper layer can issue MLME-RX-ENABLE primitives to enabled the receiver to be active for a certain period of time [8].

In this section, to simplify the analysis, we assume that the attacker sends bogus messages with crafted security headers to

$\mathcal{D}$ at a high rate (say $\rho$ packets per unit time), while only those transmitted during $\mathcal{D}$'s active period will be effective in depleting its energy. Once $\mathcal{D}$ wakes up, it listens to the channel and prepare for receiving messages. Let the duty cycle of the victim node be $\frac{\tau}{T}$ where $\tau$ is the duration of an active period and $T$ is the length of a cycle. Once entering an inactive/sleep period, the node turns OFF its radio and changes CPU state (i.e., forces CPU into some low-power mode) if no incoming message is being received or decrypted; otherwise, it turns OFF its radio at the end of the reception of the incoming message,[1] and changes CPU state until the message is finished processing. In addition, assume that the attacker messages are of the same size.

For the victim node, denote $T_{\rm rx}$ as the time spent by its radio to receive a bogus message. Denote $P_{\rm rx}, P^a_{\rm cpu}, P^i_{\rm cpu}$, and $P^o_{\rm cpu}$ as the power required by its radio when in receiving mode and by its CPU when in active, idle, and low-power modes, respectively. The energy consumption of a wireless node depends on the amount of energy expended on the following states: 1) transmitting; 2) receiving; 3) computing; 4) idle; and 5) sleeping. Based on these, the energy consumption of $\mathcal{D}$ is divided into communication cost $E_{\rm comm}$, computation cost $E_{\rm comp}$, and passive cost (when it is not involved in communication, say, sleeping) $E_{\rm passive}$.

The computation cost of the node depends on the amount of energy expended for cryptographic operations (including decryption and MIC verification). The computation cost for the decryption and verification of one message can be given by $T_{\rm dec}P^a_{\rm cpu}$, where $T_{\rm dec}$ denotes the time required for decryption and verification of a bogus message, which depends on the specific security suite used. Considering the CPU cost during the packet receiving period, we have

$$E_{\rm comp} = n_p \left(T_{\rm dec}P^a_{\rm cpu} + T_{\rm rx}P^i_{\rm cpu}\right) \tag{1}$$

where $n_p$ is the number of messages to be processed in an active period. $n_p \approx \left\lceil \frac{\tau}{\max\{T_a, \frac{1}{\rho}\}} \right\rceil$, where $T_a \triangleq T_{\rm dec} + T_{\rm rx}$ is the total time to receive and process one bogus message.

As we are concerned about the amount of energy spent by the victim node in receiving the bogus packets, the communication cost will be mainly determined by the receiving cost and is

$$\begin{aligned} E_{\rm comm} &\approx \begin{cases} n_pT_aP_{\rm rx}, & \text{if } n_pT_a \geq \tau \\ \tau P_{\rm rx}, & \text{otherwise} \end{cases} \\ &= \max\{n_pT_a, \tau\}P_{\rm rx}. \end{aligned} \tag{2}$$

The passive cost includes sleep cost and idle cost during the time after processing a current message and before the next message arrives. Hence

$$E_{\rm passive} = \begin{cases} (\tau - n_pT_a)P^i_{\rm cpu} + (T-\tau)P^o_{\rm cpu}, \\ \qquad\qquad \text{if } n_pT_a < \tau \\ (T - n_pT_a)P^o_{\rm cpu}, \quad \text{otherwise.} \end{cases} \tag{3}$$

[1] According to the standard, to turn OFF the transceiver when the active period expires, the MAC layer can issue a PLME-SET-TRX-STATE.request primitive to the PHY layer to change the state of the transceiver. When this primitive arrives at the PHY layer, if the transceiver is in RX_ON state and has already received a valid start-of-frame delimiter (SFD), the transceiver will be turned OFF at the end of the packet reception.

Then, the total energy consumed by the victim node (including receiving and processing the $n_p$ messages) in one cycle is $E_p = E_{\rm comm} + E_{\rm comp} + E_{\rm passive}$.

*a) Number of Messages Leading to Depletion:* Let $E_{\rm res}$ be the amount of available energy of $\mathcal{D}$, $E_{\rm th}$ be the threshold below which $\mathcal{D}$ will fail, and $m$ be the number of bogus messages to deactivate $\mathcal{D}$. Then

$$E_{\rm res} - \frac{m}{n_p}E_p \leq E_{\rm th} \tag{4}$$

i.e., the attacker should send at least $m \geq n_p\frac{E_{\rm res}-E_{\rm th}}{E_p}$ packets to successfully deplete the energy of $\mathcal{D}$.

*b) Lifetime Reduction:* Let $E_0$ be the initial energy of the victim node. If no attacker presents, its lifetime is $L_0 = \frac{E_0}{E_{p,0}}$ where $E_{p,0} = \tau(P_{\rm rx} + P^i_{\rm cpu})$ if we neglect the sleep cost. If $n_pT_a \geq \tau$, based on the above analysis, we have $E_p = n_p(T_{\rm dec}P^a_{\rm cpu} + T_{\rm rx}P^i_{\rm cpu}) + n_pT_aP_{\rm rx}$. Thus, the ratio of $L_0$ over the lifetime under *ghost* attack becomes

$$\begin{aligned} \frac{L_0}{L} &= n_pT_a\frac{\frac{T_{\rm dec}}{T_a}P^a_{\rm cpu} + \frac{T_{\rm rx}}{T_a}P^i_{\rm cpu} + P_{\rm rx}}{\tau(P_{\rm rx} + P^i_{\rm cpu})} \\ &= \frac{n_pT_a}{\tau}\left(1 + \frac{T_{\rm dec}}{T_a}\frac{P^a_{\rm cpu} - P^i_{\rm cpu}}{P_{\rm rx} + P^i_{\rm cpu}}\right) \qquad (5) \\ &> \frac{n_pT_a}{\tau}. \qquad (6) \end{aligned}$$

That is $\frac{L}{L_0} < \frac{\tau}{n_pT_a}$. Similarly, if $n_pT_a < \tau$

$$\frac{L_0}{L} = 1 + \frac{n_pT_{\rm dec}}{\tau}\frac{P^a_{\rm cpu} - P^i_{\rm cpu}}{P_{\rm rx} + P^i_{\rm cpu}}. \tag{7}$$

Above two equations show that the lifetime of the victim node is reduced to some extent, and the amount of reduction can be significantly large if $n_pT_a$ is much larger than $\tau$.

### C. Other Attacks Due to Ghost

*1) Denial of Service (DoS):* DoS can be easily executed in three ways.

*a) DoS due to high-computational load:* The attacker sends a number of bogus messages to quickly deplete the energy of the victim node and thereby suspend the availability of the services. It turns out that if the network has some traffic abnormality detection schemes in place, sending such numerous messages in short period of time can be easily caught. To escape from the detection, for instance, *ghost* attacker(s) can send messages either at different times or at different addresses to a subset of victim nodes in its range.

*b) DoS due to MAC misbehavior:* Due to the channel sensing and contention-based access nature of the IEEE 802.15.4 CSMA/CA protocol, if a *ghost* attacker continuously sends the traffic to the victim node, all nodes within the interference region will be deprived of channel access and services. Moreover, each node has to spend a significant amount of time sensing and waiting to get access to channel, which again will lead to energy depletion.

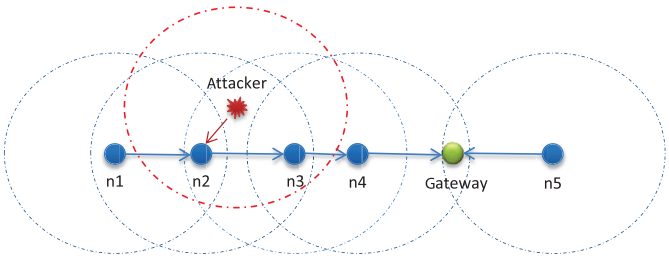



Fig. 1. Simple multihop network formed by five ZigBee nodes and a gateway, where nodes $n2$ and $n3$ are in the interference range of a *ghost* attacker.

*c) DoS with a postdepletion replay attack:* Such an attack is to be discussed in Section IV-C2.

*Analysis*: To illustrate the effect of DoS attack on the network throughput performance, we conduct mathematical analysis based on a simple multihop network as shown in Fig. 1. All the nodes apply the unslotted CSMA/CA protocol for channel access. Assume each node randomly generates packets (of the same length $L$) at the same rate $\lambda$ packets/slot, where a slot is the backoff slot in IEEE 802.15.4 standard. For ease of analysis, duty cycling is not implemented, and we assume that the minimum and maximum backoff exponents are the same as $macBE$. The nodes send packets to the gateway through multihop paths. A node will switch to receiving mode only when its MAC buffer is empty. The gateway node keeps in receiving mode and listening packets from others. Such a scenario, known as the random unsaturated traffic case, has been widely considered in evaluating the performance of the CSMA/CA protocol in single-hop network environments [32]. In this paper, we extend it to a simple multihop scenario. We assume that the *ghost* attacker sends bogus packets at a constant rate $p_{att}$ independent of the channel state.

For each node, define $\tau_i$, $\alpha_i$, and $\rho_i$ as the probabilities that this node conducts a clear channel assessment (CCA), the channel is idle when it conducts a CCA, and that there is at least one packet in its MAC buffer, respectively, in a slot. Conditioned on that its buffer is nonempty, a node will transmit a packet with probability $p_i$ and the packet will be successfully received (without collision) by its next-hop node with probability $p_i^s$. We extend the model in [33] for single-hop networks to the multihop network shown in Fig. 1. Since the modeling approach is the same for each node, below we shall focus on node $n2$. Since the backoff exponent is constant, according to [33]

$$\tau_2 = \frac{1}{\bar{b} + 1 + L\alpha_2} \tag{8}$$

where $\bar{b} \triangleq \frac{1}{2}(2^{\mathrm{macBE}} - 1)$ is the average backoff length. When this node conducts a CCA, the channel is idle only when none of its neighbors (including the attacker) is transmitting (i.e., none of them starts transmitting in either of the previous $L$ slots). Hence

$$\alpha_2 = 1 - L\left[1 - (1 - p_{\mathrm{att}})(1 - \rho_1 p_1)(1 - \rho_3 p_3)\right]. \tag{9}$$

By definition, $p_2 = \tau_2\alpha_2$ and $\rho_2 = \min\left\{\frac{\lambda + \rho_1 p_1 p_1^s}{p_2}, 1\right\}$, where $\rho_1 p_1 p_1^s$ is the rate of packets being received from $n1$. When $n2$ transmits a packet, the packet will be successfully received by $n3$ only if: 1) neither $n1$ nor the attacker starts to transmit packet simultaneously as $n2$; 2) $n4$ does not transmit packet in any slot during the transmission by $n2$; and 3) $n3$ is in receiving mode. Therefore

$$p_2^s = (1 - \rho_1 p_1)(1 - p_{\mathrm{att}})(1 - L\rho_4 p_4)(1 - \rho_3). \tag{10}$$

Based on the same approach, the performance of the other nodes can be modeled. Then, the throughput of the nodes can be calculated as $S_1 = \rho_1 p_1 p_1^s p_2^s p_3^s p_4^s$, $S_2 = \frac{\lambda}{\lambda + \rho_1 p_1 p_1^s}\rho_2 p_2 p_2^s p_3^s p_4^s$, $S_3 = \frac{\lambda}{\lambda + \rho_2 p_2 p_2^s}\rho_3 p_3 p_3^s p_4^s$, $S_4 = \frac{\lambda}{\lambda + \rho_3 p_3 p_3^s}\rho_4 p_4 p_4^s$, and $S_5 = \rho_5 p_5 p_5^s$.

We name the scenario shown in Fig. 1 as case 1. For comparison, another case (case 2) is considered in which the attacker's interference range only covers $n3$. Numerical results are shown in Fig. 1, where $L = 3, \mathrm{macBE} = 3, \lambda = 0.02$. In both cases, due to increase in attack intensity, nodes that are within the interference range of the *ghost* have fewer opportunities to send out packets, and their throughput drops quickly. Since packets from $n1$ transverse the interfered area, its throughput drops even quicker than that of the nodes within the area. On the other hand, due to less interference from $n3$, the throughput of $n4$ and $n5$ whose packets do not cross the interfered area may gain some improvement. From Fig. 2(d), we observe that, even $n2$ is a relay of $n1$, they experience similar throughput variation when $n3$, a common relay of them is under attack. This observation is used in developing our attacker localization method in Section V-A.

In case of duty cycling, similar results as in Fig. 2 can be obtained. In particular, if knowing the active time of its neighbors, the attacker can completely destroy their throughput by using a high attack rate only during their active periods, regardless of the MAC protocol used by the neighbors. This is another reason why we choose the random unsaturated traffic case to reveal the DoS impact of the attacker on the CSMA/CA protocol performance.

*2) Postdepletion Attacks:* What happens when the node encounters energy depletion? If no specific controls are taken, the node will emerge with a cleared ACL table on reboot and consequently, all the nonces or the frame counters will be reset to the initial value 0. Upon such resetting, we see the plausibility of two attacks:

*a) Replay attack due to frame counter reuse:* Replay attack can cause severe problems. For example, medical nodes such as fitbits, pacemakers, and insulin pumps that replay old messages can lead to erroneous outputs and dangerous events. Below, we show that a *ghost* can launch replay attack. Consider a *ghost* attacker who captures a certain number of legitimate packets in the beginning (preattack phase) and starts depleting the energy of the nodes through bogus messages. When the counters are reset to 0 upon restart, the attacker can start replaying those messages. The replay attack can lead to serious results in two aspects. 1) The attack can have the receiver consume obsolete messages, leading to unexpected operations. 2) The replay attack can result in the DoS. Specifically, if the attacker replays messages with frame counters larger than the current legitimate value maintained in the ACL table, the ACL table will then be updated by such replayed messages.

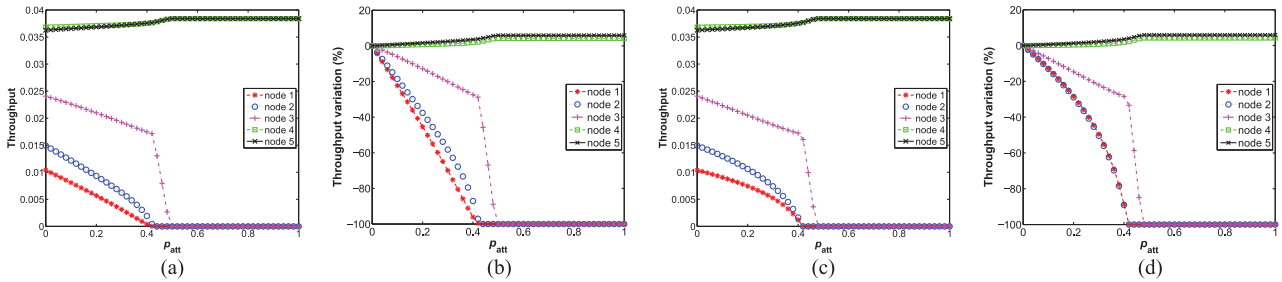



Fig. 2. Network performance under various attack rate. $p_{\mathrm{att}} = 0$ means the scenario without *ghost*. (a) Throughput: case 1. (b) Throughput variation: case 1. (c) Throughput: case 2. (d) Throughput variation: case 2.

Consequently, in future, when legitimate nodes send messages with the actual frame counter, the message will be rejected as it carries a counter value less than what is currently stored in the ACL of the victim node (which was updated by the replayed messages).

We would like to emphasize that the *ghost* attack significantly facilitates the chance of launching a replay attack. From the IEEE 802.15.4 specification, it follows that by leveraging a 4-B counter, one can execute a replay attack only after $2^{32}$ frames; nonetheless, in this work, we demonstrate that by executing the *ghost* attack, an adversary can use a number of messages which are far less than $2^{32}$ to deplete the energy of a victim node. Upon the restart of this victim node, the attacker can launch the relay attack.

*b) Loss of confidentiality due to nonce reuse:* According to the standard, messages destined for the next hop nodes are encrypted by XORing the plaintext message $P$ with the unique keystream $O$ constructed from the unique nonces, static flags, and the preshared key. Take the AES-CTR scheme for instance. The encryptor partitions the plaintext $P$ into 16-B blocks and then XOR along with $O$ to generate the corresponding ciphertext $C = P \oplus O$. If the nonces are reused, the keystream will repeat for the subsequent messages. Thus, for a different plain text $P'$, the output of the encryptor will be $C' = P' \oplus O$. Assume that the attacker has captured the ciphertexts $C$ and $C'$, then a simple XORing of the two ciphertexts will lead to $C' \oplus C = (P' \oplus O) + (P \oplus O) = P' \oplus P$; such a simple operation can trivially be broken using statistical analysis and the subsequent plaintext messages can be deciphered without the knowledge of preshared key, if the content of any one message can be guessed.

## V. Countermeasures

### A. Attacker Localization

The first step to defend against *ghost* is to know its existence and location. *Ghost* can be known by the victim node if it continuously decodes bogus messages, or may be detected by its neighbors which analyze the impact of the induced DoS attack. Such phenomena have been utilized to develop neighbor monitoring based techniques to detect energy depletion attacks or protocol noncompliance attacks [22], [25]. However, there are two reasons motivated us to develop another localization method without relying on neighbor monitoring. First, since an intensive attack can significantly block the channel access of its neighbors or cause them busy in processing the bogus messages, we cannot rely on them to timely report the existence of the attacker. Second, the impact of the induced DoS attack is not confined to the attacker's neighborhood. As demonstrated in both Section IV-C1 and Fig. 6(a), some nodes may even get benefited in terms of throughput due to *ghost*. In this case, a node may be mistakenly deemed misbehaving by its neighbors. In this work, we let the cluster head nodes (or CHs for short, which are assumed trustable) to carry out the attacker localization. This is practically reasonable because many IoT applications distribute some central nodes serving as data fusion centers. In particular, the cluster-tree-based topologies are also supported by the IEEE 802.15.4 standard. We assume that the interference range of the attacker is limited; otherwise, if the entire cluster is under the attack's interference, localizing the attacker becomes very difficult, if not impossible. The basic idea of localization is to analyze the throughput variation in each node within a cluster and use a weighted centroid algorithm to localize the attacker.

*1) Phase 1:* Identify suspected victim nodes. The CH applies a moving analysis window to record the throughput of each node[2] in the corresponding cluster, and calculates the percentage of throughput variation (denoted as $\Delta S_i$ for node $i$) in real time. Suspected nodes are identified by checking the variation along each path. The observation mentioned in Section IV-B1 is utilized to exclude nodes outside of the attacker's interference area but their packet routes traverse that area. The exclusion operation is controlled by a threshold $\delta'$ as in Algorithm 1. Denote each path as $\mathcal{P}_l = (l_1, l_2, \ldots, CH)$ with $l_1$ as the source node.

*2) Phase 2:* Group suspected victim nodes. We consider a general case that neither the number nor interference ranges of the *ghosts* is known to the CH. Two suspected nodes can be considered under the interference of the same attacker and included in the same group if there is a circle that covers both of them but does not cover any other nonsuspected nodes. Otherwise, they are considered to be attacked by different attackers and included in different groups. Since an attacker may want to span its attacking range over a large number of nodes, small clusters will not be considered for attacker localization.

*3) Phase 3:* Calculate the *ghost* location. For each group, suppose there are $s$ suspected nodes and the throughput

[2]For dense deployment, to save computation cost, the CH can select a few paths with the nodes uniformly scattering over the area for analysis.

**Algorithm 1.** Identify suspected victim nodes

$\mathcal{A} \leftarrow \emptyset$: set of suspected victim nodes
**for** *all paths* $\mathcal{P}$ **do**
  **if** $\Delta S_{l_1} > -\delta$ **then**
    continue;
  **for** $i \leftarrow 1, \ldots, m-1$ **do**
    **if** $\Delta S_{l_i} < -\delta$ **the**
      **if** $i+1 = m$ or $|\Delta S_{l_i} - \Delta S_{l_{i+1}}| > \delta'$ **then**
        add $l_i$ into $\mathcal{A}$
        **if** $|\Delta S_{l_i} - \Delta S_{l_{i-1}}| < \delta'$ **then**
          add $l_{i-1}$ into $\mathcal{A}$

variance is $\{\Delta S_i\}$. Then, the attacker's location can be estimated by the following weighted sum method

$$\mathbb{L}_{\text{ghost}} = \sum_{i=1}^{s} \frac{\Delta S_i}{\sum_{k=1}^{s} \Delta S_k} \mathbb{L}_i \tag{11}$$

where $\mathbb{L}_i$ is the location of each suspected node.

Based on analyzing the throughput variations of the nodes, the proposed localization method is not affected if attackers can manipulate their energy consumption. Moreover, if the attackers do not have the secrete keys, they cannot manipulate the throughput of legitimate nodes by generating fake messages with the addresses of the legitimate nodes, i.e., it is difficult for the attackers to disturb the localization method by pretending to be the legitimate nodes.

The proposed localization method focuses on the case that there is at most one attacker in the network. In multiattacker scenarios, the proposed localization method is still valid if the attackers's influence ranges interleave each other. Otherwise, if two attackers are close, it is possible that they are identified as a single attacker by the proposed method. In this case, it is a challenging issue to identify the number of collocated or geographically close attackers and distinguish their locations, solely based on the throughput information available to the monitoring node. A possible solution to this can be as follows. By the proposed method, such attackers may be treated as a single virtual attacker and its location can be estimated. Then, some mobile actuators can be sent to that location and perform more precise intrusion detection [34]. We leave this problem in our future work.

*4) Discussions on Other Countermeasures:* To mitigate the impact of *ghost*, each normal node can maintain a list of misbehaving nodes. If a victim node observes a certain number of messages with bogus security headers, it will add the sender node to the blacklist and inform the network or the operator about the attack. One plausible issue with this approach is the *badmouthing* attack [35] in which an attacker sends bogus messages with fake addresses to lure a node to blacklist all its neighbors, leading to a temporary disruption or DoS. The badmouthing attacker is hard to be directly identified; however, the victim node can apply the challenge-response scheme (as discussed below) to avoid blacklisting its normally behaving neighbors. A side benefit of such a challenge-response scheme is that those normal neighbors can then know that a certain node is badmouthing and report the existence of the *ghost*. Moreover, our localization algorithm can still find the location of the attacker and inform others to detour their packets around the neighborhood of the attacker.

Another approach is to add a second layer of *challenge-response scheme* in which the node, after observing a certain number of bogus messages from a specific address or when the energy is restored and the communications are re-initiated, will challenge the attacker with a random number. To correctly respond, the attacker needs to know the secret key which is available only to the legitimate parties and is securely stored in the node. On restoration of energy, we also require, as part of the challenge-response scheme, to establish new keys so that they do not reuse the same nonce twice with the same key. Another solution is to add a timestamp to the protocol and after a reboot; each node is required to update its timestamp by communicating with the controller. There are also works that suggest storing the counter values in nonvolatile or flash memory so that even if the energy is lost, the state of the node can be restored. Nevertheless, storing and retrieving values from flash memory is slow and energy inefficient, specifically for energy-constrained nodes [9].

## VI. Simulations

In this section, we present extensive simulation results to demonstrate the impact of the proposed *ghost* attack. We develop our simulation codes within the NS-3 environment [36]. We consider a ZigBee network with a single coordinator serving as the gateway, $n$ legitimate nodes, and a *ghost* attacker which injects messages with bogus security headers to drain victim nodes' energy. Each ZigBee node is equipped with an 8-MHz (a typical clock frequency of a ZigBee node's MPU) processor. All the nodes apply the nonbeacon mode of the IEEE 802.15.4 CSMA/CA protocol for channel access. We adopt the following energy consumption model [37]: the current drained by each node for its CPU being in active, idle, and power-saving states is 8.0, 3.2 mA, and 110 $\mu$A, respectively; the current drained by each node's transceiver for receiving and transmitting is 7.0 and 8.5 mA, respectively; and the transmission power of each node is 0 dB. We also adopt an accurate Li-Ion battery model (which is provided in NS-3 based on the models proposed in [38]) for each node with the initial capacity set as 2.45 Ah to simulate the energy drain activities. The computation time is estimated in the following way: we first measure the computation time of an encryption/decryption process which is carried out in the simulating PC of frequency 1.85 GHz. Then, this computation time is mapped to that of the 8 MHz processor. The data rate is the default value—250 kb/s.

### A. Energy Expenditure for Security

We first compare the energy expenditures in encryption and decryption for the security suites listed in Table I. To avoid the impact from other factors such as interference and routing, we simulate a simple scenario: a *ghost* attacker sends packets to a victim node. The victim node runs in a duty-cycling mode with fixed duty cycle 1%. Specifically, the victim node stays in active

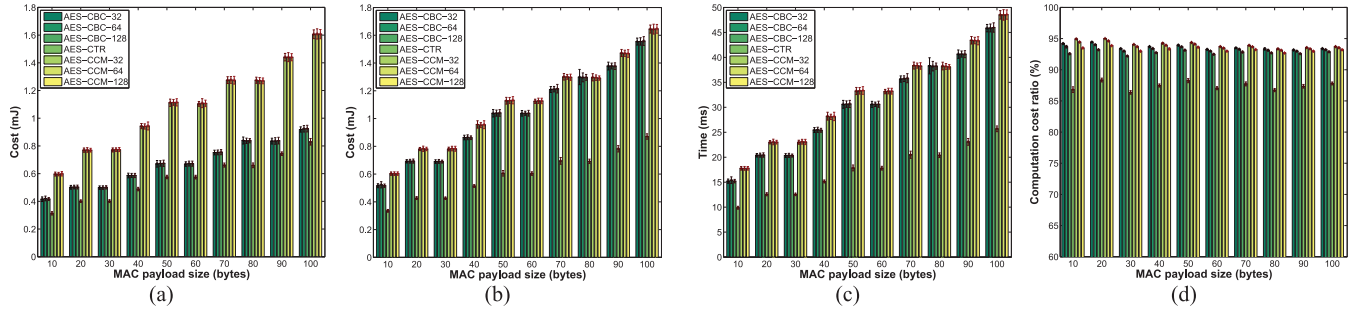



Fig. 3. Energy and time costs. (a) and (b) Average energy consumption for encrypting a packet and decrypting a secured packet, respectively. (c) Average time (ms) for decrypting and verifying a secured packet. (d) Percentage of CPU computational cost in receiving (including decrypting) a secured packet.

state for channel listening for 1 ms in every 0.1 s. If there are packets arriving during the active period, the node will process the packets and then switch to sleep if there is still leftover time within the current cycle. If there is no packet received, the node will directly sleep upon expiration of the active period. Such a mechanism would ensure that the victim node could function properly over 1 year. To demonstrate the energy-depletion attack, the attacker sends crafted messages to this victim node every 0.1 s within its active period.

Fig. 3 reports the average computational energy cost and time for different security suites, where the averages are taken over 25 000 bogus packets sent by the attacker. Generally, both encryption and decryption energy costs increase as the payload size increases. Moreover, Fig. 3(a) and (b) demonstrates that the ascending order of the security suites in terms of the average per-packet energy cost (for both encryption and decryption) is as AES-CTR, AES-CBC-MAC, and AES-CCM. This order is reasonable because more data bytes are involved in the XORing operations when the protocol changes from AES-CTR to AES-CCM.

As both the encryption and decryption processes pad additional 0's to the secured payload to partition input data into blocks of 16 B, the amount of energy expenditure may remain the same for some payload sizes, which is shown in Fig. 3(b). For instance, with AES-CCM-32, a secured packet with MAC payload of 20 B will be padded by 10 B of 0's so that the resulting string, together with a 2-B length indicator, has length divisible by 16 B. However, a MAC payload data of 30 B does not need extra 0's. As a result, the two secured packets with different MAC payload sizes consume almost the same computational energy. It is also interesting to observe that, for some MAC payload lengths (e.g., 80 B in the figure), the average energy costs for decrypting a secured packet under the AES-CCM and AES-CBC-MAC security suites are quite close. In fact, after padding with necessary 0's, the total numbers of bytes (including secured data and authentication fields) input to the XORing operations are very close to each other in these two security suites.

The processing time by the victim node's CPU for decrypting and verifying a secured packet is much longer than that by its transceiver for receiving it. For instance, as shown in Fig. 3(c), with AES-CCM-128, the computation time for decrypting and verifying a packet with 60-B payload is about 35 ms, while the receiving time is only about 3 ms. As shown in Fig. 3(d), over 88% of the victim node's energy is spent by its CPU for decrypting crafted packets, which clearly demonstrates the significance of the *ghost* attack. We can also observe that AES-CCM-32 introduces the highest CPU cost ratio. Note that the computation cost ratios in all situations slightly decrease when the MAC payload size increases. The reason is that the payload size does not significantly impact the computational energy cost, while it is linearly related to the transceiver's energy consumption in receiving.

### B. Node Lifetime Under the Ghost Attack

Fig. 4 presents the lifetime of the victim node in different scenarios. The attack is according to the configuration in Section VI-A with a MAC payload of 60 B. The results in Fig. 4 clearly show that the *ghost* attack significantly shortens the lifetime from over 1 year to days. Specifically, the lifetime reduces to around 10.2%, 6.5%, and 6.1% of the baseline value (i.e., without *ghost* attack) under AES-CTR, all the AES-CBC-MAC and all the AES-CCM security suites, respectively. On the other hand, based on the analysis presented in Section IV-B1, we can easily calculate the lifetime radio $\frac{L}{L_0}$ by using the computing time shown in Fig. 3(c). The calculated results are 10.9%, 6.8%, and 6.5% for AES-CTR, all the AES-CBC-MAC and all the AES-CCM security suites, respectively, which well match the simulation results.

### C. DoS Attack

The following simulations are conducted on a network with $n = 38$ nodes randomly deployed inside a $100 \times 100$ $\mathrm{m}^2$ area, as shown in Fig. 5. The gateway node is located at the center of this area. All the nodes (including the attacker) are assumed to have the same communication range (30 m) and the same interference range (40 m) , respectively. All the legitimate nodes periodically report data (1 packet/s) to the gateway through multihop routes, where we adopt the shortest path routing. The attacker node select node 1 as the victim node and sends bogus packets to it, where the interpacket sending intervals obey Poisson distribution with 20 ms as the mean. Here, we interpret the attack as a DoS attack, according to the discussion in Section IV-C1. Simulation results with AES-CTR are as follows.

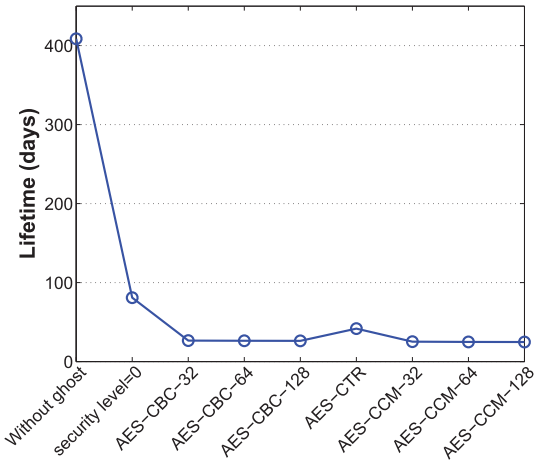



Fig. 4. Amount of time (in days) needed to deplete the energy, where the attacker rate is 10 packets/s.

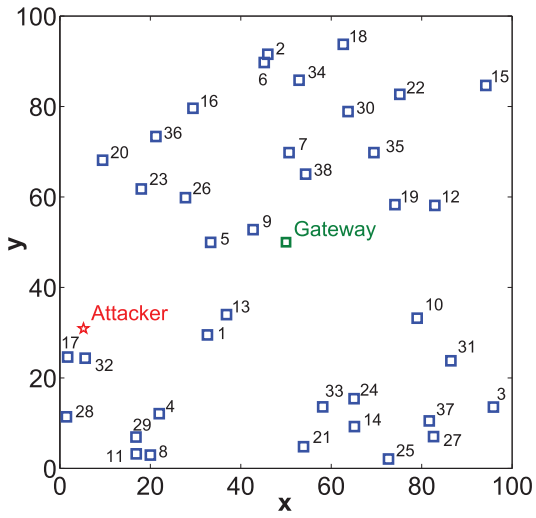



Fig. 5. Network topology.

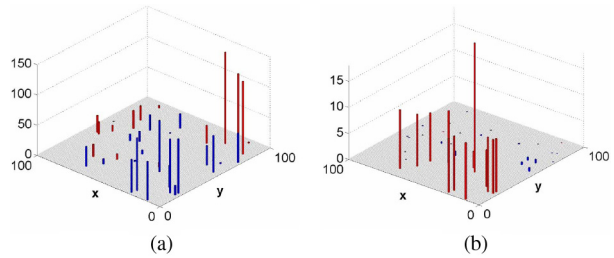



Fig. 6. Impact of DoS attack. (a) Throughput variation (%). (b) Energy drain speed variation (%).

Fig. 6(a) plots the throughput variation (i.e., the percentage of throughput that is increased/decreased due to the attack) for each node. Note that the red and blue bars mean positive and negative variations, respectively. Based on this figure, the DoS attack by the attacker will change the distribution of the network traffics, and the effects are two-fold. Since the attacker causes node 1 to spend most time on receiving and decrypting crafted packets, the spare capacity that node 1 can provide to others for relaying and also transmitting packets of its own significantly

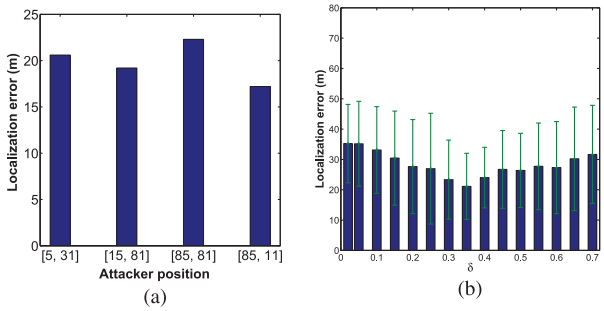



Fig. 7. Localization error. (a) Different attacker positions. (b) Different topologies.

drops. Therefore, nodes, such as node 28, that needs node 1 to relay packets experiences a much higher packet loss rate in the presence of the attack. Although other nodes that receive a crafted packet will directly discard it due to unmatched destination address, there is still bandwidth and energy waste due to channel sensing, packet receiving, and collisions due to the hidden terminal impact [39]. Thus, the throughput of the attacker's neighbors (e.g., node 8, 17, and 32) reduces greatly as shown in Fig. 6(a). Therefore, the attacker can perform DoS attack on all its neighbors by just targeting at one of them. On the other hand, the presence of the attacker may be beneficial to some other nodes, e.g., node 2, 10, 27, 31, and 37 as shown in Fig. 6(a), which locate far away. For those nodes, because the traffics of some hidden-terminal nodes locating within the attacker's interference range are partly suppressed by the attacker, their successful packet deliver ratios increase in turn. Therefore, Fig. 6(a) demonstrate that the impact of the *ghost* attack can be propagated in a multihop network such that it can change the throughput of a larger amount of nodes more than that of its neighbors.

Fig. 6(b) plots the 3-D view of the variation in energy drain speed. We can see that, under the DoS attack, the energy drain speed is accelerated for the attacker's direct neighbors due to receiving and processing the large amount of bogus messages sent from the attacker. For those non-neighboring nodes, their energy drain speeds do not change too much. Especially for those nodes which gain some benefit in throughput (e.g., node 2, 10, 27, 31, and 37), they spent more energy for transmitting data on the one hand, but save more energy for channel sensing and conducting backoff for each packet on the other hand. As a result, their average energy consumptions are roughly balanced. Therefore, Fig. 6(b) demonstrates that the impact of the *ghost* attack in terms of energy drain speed is mostly confined within its neighborhood.

### D. *Effectiveness of Countermeasures*

The performance of the proposed attacker localization method is demonstrated in Fig. 7(a) where the attacker is placed at four different positions in the network shown in Fig. 5. It shows that the localization error is around 20 m, which is smaller than the communication range of each legitimate node. We also generate more random topologies (the deployment area is 100 m $\times$ 100 m and the number of nodes is 80) to evaluate

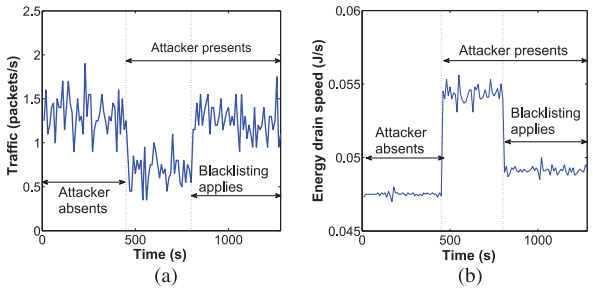



Fig. 8. Effect of blacklisting on the victim node's: (a) traffic and (b) energy drain speed.

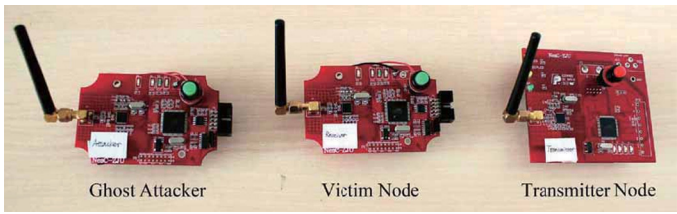



Fig. 9. ZigBee nodes in experiments.

the localization error. In each topology, we randomly select one node to act as the attacker. As shown in Fig. 7(b), the average localization error falls in $(20, 35)$ m, which is in the same scale as the communication range of each node. In addition, the threshold value $\delta$ (see Algorithm 1) also affects the localization performance. A small $\delta$ will include a good number of nodes (including those far away from the attacker because the impact in terms of throughput variation will be propagated over the network) in the suspected victim node set, which will increase the error. On the other hand, a large $\delta$ will exclude some neighbors of the attacker from the suspected set, and hence degrade the localization performance.

As shown in Fig. 8(a), the intrusion of the attacker significantly reduces the victim node's traffic, which is then recovered by blacklisting the attacker so that all future crafted packets are directly discarded by the victim node at the MAC layer. In terms of the energy drain speed, similar trends can be observed in Fig. 8(b). However, the drain speed cannot be fully recovered because the victim node's physical layer energy expenditure for receiving crafted packets is inevitable. In Section V, we also proposed the challenge-response scheme to address both DoS and replay attacks. The traces of the victim nodes' traffic and energy drain rates exhibit similar patterns as shown in Fig. 8(a) and (b), and hence are omitted due to the page limit.

## VII. Experiments

To further validate the effect of *ghost*, we conduct physical experiments based on the ZigBee nodes shown in Fig. 9. Each node has an ATmega128L processor operating with 8-MHz frequency, 128-kB In-System programmable Flash, and a CC2420 RF transceiver compliant with the 2.4-GHz IEEE 802.15.4 standard. We develop C programs to control the embedded security suites of IEEE 802.15.4.

TABLE II
Node Lifetime in Single-Hop Scenario

| | | | Battery | | | Average | |
|---|---|---|---|---|---|---|---|
| | | | #1 (h) | #2 (h) | #3 (h) | Lifetime (h) | Percent (%) |
| Experimental | Attack absent | | 1.43 | 1.5 | 1.67 | 1.53 | 100 |
| | Attack presents | 011 | 0.93 | 0.8 | 1.08 | 0.93 | 60.9 |
| | | 100 | 1.03 | 0.95 | 1.18 | 1.05 | 68.5 |
| | | 111 | 0.87 | 0.83 | 1.0 | 0.9 | 58.7 |
| Mapped | Attack absent | | 64.3 | 71.4 | 74.3 | 70 | 100 |
| | Attack presents | 011 | 4.72 | 3.58 | 5.48 | 4.58 | 6.6 |
| | | 100 | 5.55 | 4.58 | 6.32 | 5.48 | 7.8 |
| | | 111 | 4.2 | 3.78 | 4.85 | 4.28 | 6.1 |

### A. Single-Hop Scenario

A simple single-hop network consisting of an attack node and a pair of transmitter and receiver/victim nodes is deployed in the experiments. The transmitter and victim nodes, separated by a distance of 0.5 m, operate in a low duty-cycling mode: they wake up and stay active for roughly 5 ms every 50 ms. Due to clock drift and the consequent problem of asynchronous communication rendezvous in duty-cycling networks, it is possible that the receiver does not receive anything from the transmitter in one active period. The *ghost* attack node is placed close to both the transmitter and the victim nodes. It broadcasts 25 bogus packets (of size 60 B) per second.

We measure the lifetime of the victim node powered by a 70-mAh Li-Ion rechargeable battery. We conduct three groups of independent experiments with three batteries. In order to have the same initial energy, the batteries are charged for a same period of time after they have been completely depleted. We assume that the *ghost* has unlimited resource as powered by a dc power supply. The transmitter node's power supply is a normal AA battery which has much higher initial energy than the victim node.

The experiment results are shown in Table II which clearly justifies the effectiveness of *ghost* attack since the node lifetime is obviously shortened under attack. Particularly, with the AES-CCM-128 security suite, the node lifetime is significantly reduced by 39.5%, 44.4%, and 40.0% with respect to the three batteries used, respectively. Moreover, the percentages of lifetime reduction also climb as the security level increases.

For ease of implementation, the network settings in the experiments are different from our simulations in Section VI-B, for which reason the results in Table II are different from those shown in Fig. 4. Besides using different battery models and different initial battery power, there are three other major differences between experiments and simulations. 1) A transmitter is introduced to inject packets to the victim node in the experiments. 2) The victim node's duty cycle is 10%, larger than that in the simulations (which is 1%). 3) In our simulations, once the victim node enters sleep period, it turns OFF its radio and switches its CPU to power-saving mode. However, in our experiments, the victim node only switches off its radio (while keeping its CPU idle) during its sleeping period.

TABLE III
NODE WORKING CURRENT

| Working mode | Average current (mA) |
|---|---|
| Idle | 11 |
| Pure data processing | 18 |
| Receiving and processing unsecured packets | 32 |
| Receiving and processing secured packets | 40 |

To fairly compare the simulation and experimental results, we map the experiment settings to the simulation simulation settings and obtain the mapped results shown in Table II. The mapping is based on the working current. By utilizing an ETCR6000 ac/dc Clamp Leaker with the sampling rate of two times per second, the working current under different modes of the victim node is measured and the mean values are shown in Table III. In the idle state, i.e., MCU is in idle state and the radio is turned OFF, node's energy is consumed by the basic node operations including the glowing of LED light. When we maintain the MCU executing instructions (e.g., keeping on adding) incessantly, the energy consumption is raised by 7 mA. Furthermore, when we turn ON the radio, the node's working current is remarkably raised. Specifically, if the security suite is implemented into the procedure of transmission, the current reaches around 40 mA; otherwise it reaches around 32 mA.

Based on the measured node working current shown in Table III, the experiment results are mapped to those shown in Table II by removing the transmitter, using 1% as the victim node's duty cycle and switching its CPU to power-saving mode once it sleeps. We can observe that, under *ghost* attack, the node lifetime is remarkably reduced to 6%–8% of the lifetime when the attacker is not present. The percentages of lifetime reduction are very close to those observed in the simulations shown in Fig. 4. Therefore, both of our experiments confirm the significance of the *ghost* attack and validate our methodology of simulations.

### B. Multihop Scenario

A simple multihop network as shown in Fig. 1 is constructed with the ZigBee nodes. All the legitimate nodes turn OFF the radio for 5 ms in every 50 ms and transmit the packets to the sink node through the fixed path shown in the figure. The *ghost* attacker uses the same strategy as in the above single-node scenario to attack Node 2. Each node uses the AES-CCM-128 as its security suite. The experimental results of four representative nodes (Node 1, 2, 3, and 5) are shown in Table IV. We can observe that Node 2, the target of the *ghost*, suffers from the fastest speed of energy draining. Its lifetime is shortened as much as 31.6%, which basically approaches the performance of the victim node in our single-hop scenario (as compared to the experimental results in Table II). Since Node 1 and 3 also receive the bogus packets, their lifetime is decreased by 15.8% and 16.2%, respectively. Since the traffic of Nodes 1, 2, and 3 are suppressed by the *ghost*, the throughput of Node 5 increases a little bit, which causes 9.5% reduction of its lifetime. This trend of Node 5 is also demonstrated in Fig. 2 based on our previous analytical model.

TABLE IV
NODE LIFETIME IN THE MULTIHOP SCENARIO

| | | Battery set | | | Average | |
|---|---|---|---|---|---|---|
| | | #1 (min) | #2 (min) | #3 (min) | Lifetime (min) | Percent (%) |
| Attack absent | $N_1$ | 85 | 80 | 81 | 82 | 100 |
| | $N_2$ | 81 | 78 | 79 | 79 | 100 |
| | $N_3$ | 78 | 71 | 72 | 74 | 100 |
| | $N_5$ | 88 | 84 | 80 | 84 | 100 |
| Attack presents | $N_1$ | 74 | 64 | 68 | 69 | 84.1 |
| | $N_2$ | 53 | 52 | 59 | 54 | 68.4 |
| | $N_3$ | 60 | 62 | 64 | 62 | 83.8 |
| | $N_5$ | 77 | 73 | 77 | 76 | 90.5 |

## VIII. CONCLUSION

In this paper, we investigate a severe attack in ZigBee-based IoT networks in which the attacker transmits a number of bogus messages to lure the receiving victim node to do the superfluous security-related computations, leading to battery depletion. Our simulation results demonstrate that by launching this attack, an attacker could easily reduce the lifetime of ZigBee nodes from years to days. We also demonstrate that the *ghost* attack can further trigger severe DoS and postdepletion attacks. We propose several recommendations on how to withstand the *ghost* and other related attacks in ZigBee networks. Extensive simulations are provided to show the impact of the *ghost* attack and the performance of the proposed recommendations. To further validate the effectiveness of *ghost* attack, physical experiments were conducted on ZigBee nodes and interestingly, our results show that the lifetime of nodes are significantly impacted with this attack. We believe that the presented work will aid the researchers to improve the security of ZigBee networks further.

## REFERENCES

[1] J. A. Stankovic, "Research directions for the Internet of Things," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 3–9, Feb. 2014.
[2] N. Lu, N. Cheng, N. Zhang, X. Shen, and J. W. Mark, "Connected vehicles: Solutions and challenges," *IEEE Internet Things J.*, vol. 1, no. 4, pp. 289–299, Aug. 2014.
[3] Y. Liu, C. Yuen, X. Cao, N. Ul Hassan, and J. Chen, "Design of a scalable hybrid MAC protocol for heterogeneous M2M networks," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 99–111, Feb. 2014.
[4] Y. Zhang, S. He, and J. Chen, "Data gathering optimization by dynamic sensing and routing in rechargeable sensor networks," *IEEE/ACM Trans. Netw.*, Jun. 2015, to be published.
[5] K. Chebrolu and A. Dhekne, "Esense: Energy sensing-based cross-technology communication," *IEEE Trans. Mobile Comput.*, vol. 12, no. 11, pp. 2303–2316, Nov. 2013.
[6] K. Zhang, X. Liang, R. Lu, and X. S. Shen, "Sybil attacks and their defenses in the Internet of Things," *IEEE Internet Things J.*, vol. 1, no. 5, pp. 372–383, Apr. 2014.
[7] H. Zhang, P. Cheng, L. Shi, and J. Chen, "Optimal DoS attack scheduling in wireless networked control system," *IEEE Trans. Control Syst. Technol.*, Aug. 2015, to be published.
[8] IEEE Comput. Soc., *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*, IEEE Standard 802.15.4, 2006.
[9] N. Sastry and D. Wagner, "Security considerations for IEEE 802.15.4 networks," in *Proc. ACM Workshop Wireless Security*, 2004, pp. 32–42.

[10] J. Zheng, M. J. Lee, and M. Anshel, "Toward secure low rate wireless personal area networks," *IEEE Trans. Mobile Comput.*, vol. 5, no. 10, pp. 1361–1373, Oct. 2006.
[11] J. Wright, "Killerbee: Practical ZigBee exploitation framework," in *11th ToorCon Conf.*, San Diego, CA, USA, 2009
[12] Y. Xiao, H. Chen, B. Sun, R. Wang, and S. Sethi, "MAC security and security overhead analysis in the IEEE 802.15.4 wireless sensor networks," *EURASIP J. Wireless Commun.*, vol. 2006, pp. 1–12, 2006, Art. ID 93830.
[13] M. Doomun, K. Soyjaudah, and D. Bundhoo, "Energy consumption and computational analysis of Rijndael-AES," in *Proc. IEEE/IFIP Int. Conf. Central Asia Internet*, 2007, pp. 1–6.
[14] D. R. Raymond, R. C. Marchany, M. I. Brownfield, and S. F. Midkiff, "Effects of denial-of-sleep attacks on wireless sensor network MAC protocols," *IEEE Trans. Veh. Technol.*, vol. 58, no. 1, pp. 367–380, Mar. 2009.
[15] M. Pirretti, S. Zhu, N. Vijaykrishnan, P. McDaniel, M. Kandemir, and R. Brooks, "The sleep deprivation attack in sensor networks: Analysis and methods of defense," *Int. J. Distrib. Sens. Netw.*, vol. 2, no. 3, pp. 267–287, 2006.
[16] T. Bhattasali, R. Chaki, and S. Sanyal, "Sleep deprivation attack detection in wireless sensor network," *Int. J. Comput. Appl.*, vol. 40, no. 15, pp. 19–25, 2012.
[17] M. Li, I. Koutsopoulos, and R. Poovendran, "Optimal jamming attack strategies and network defense policies in wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 9, no. 8, pp. 1119–1133, Aug. 2010.
[18] E. Y. Vasserman and N. Hopper, "Vampire attacks: Draining life from wireless ad hoc sensor networks," *IEEE Trans. Mobile Comput.*, vol. 12, no. 2, pp. 318–332, Feb. 2013.
[19] J. Tang, Y. Cheng, and W. Zhuang, "Real-time misbehavior detection in IEEE 802.11-based wireless networks: An analytical approach," *IEEE Trans. Mobile Comput.*, vol. 13, no. 1, pp. 146–158, Jan. 2014.
[20] A. Mpitziopoulos, D. Gavalas, C. Konstantopoulos, and G. Pantziou, "A survey on jamming attacks and countermeasures in WSNs," *IEEE Commun. Surv. Tuts.*, vol. 11, no. 4, pp. 42–56, Fourth Quart. 2009.
[21] M. Brownfield, Y. Gupta, and N. Davis, "Wireless sensor network denial of sleep attack," in *Proc. Annual IEEE SMC Inf. Assur. Workshop*, 2005, pp. 356–364.
[22] Q. Dong, D. Liu, and P. Ning, "Providing DoS resistance for signature-based broadcast authentication in sensor networks," *ACM Trans. Embedded Comput. Syst.*, vol. 12, no. 3, p. 73, 2013.
[23] X. Hei, X. Du, J. Wu, and F. Hu, "Defending resource depletion attacks on implantable medical devices," in *Proc. IEEE GLOBECOM*, 2010, pp. 1–5.
[24] X. Cao, L. Liu, W. Shen, A. Laha, J. Tang, and Y. Cheng, "Real-time misbehavior detection and mitigation in cyber-physical systems over WLANs," *IEEE Trans. Ind. Informat.*, Nov. 2015, to be published.
[25] M. Li, S. Salinas, P. Li, J. Sun, and X. Huang, "MAC-layer selfish misbehavior in IEEE 802.11 ad hoc networks: Detection and defense," *IEEE Trans. Mobile Comput.*, vol. 14, no. 6, pp. 1203–1217, Jun. 2015.
[26] A. H. Lashkari, M. M. S. Danesh, and B. Samadi, "A survey on wireless security protocols (WEP, WPA and WPA2/802.11i)," in *Proc. IEEE Int. Conf. Comput. Sci. Inf. Technol.*, 2009, pp. 48–52.
[27] H. Wang, D. Zhang, and K. G. Shin, "Detecting SYN flooding attacks," in *Proc. IEEE INFOCOM*, 2002, vol. 3, pp. 1530–1539.
[28] J. Zheng and M. Lee, "Will IEEE 802.15.4 make ubiquitous networking a reality? A discussion on a potential low power, low bit rate standard," *IEEE Commun. Mag.*, vol. 42, no. 6, pp. 140–146, Jun. 2004.
[29] J. Daemen and V. Rijmen, *The Design of Rijndael: AES—The Advanced Encryption Standard*. New York, NY, USA: Springer, 2002.
[30] W. Shen, W. Hong, X. Cao, B. Yin, D. M. Shila, and Y. Cheng, "Secure key establishment for device-to-device communications," in *Proc. IEEE GLOBECOM*, 2014, pp. 336–340.
[31] P. Park, S. C. Ergen, C. Fischione, and A. Sangiovanni-Vincentelli, "Duty-cycle optimization for IEEE 802.15.4 wireless sensor networks," *ACM Trans. Sens. Netw.*, vol. 10, no. 1, p. 12, 2013.
[32] X. Cao, J. Chen, Y. Cheng, X. Shen, and Y. Sun, "An analytical MAC model for IEEE 802.15.4 enabled wireless networks with periodic traffic," *IEEE Trans. Wireless Commun.*, vol. 14, no. 10, pp. 5261–5273, Oct. 2015.
[33] X. Ling, Y. Cheng, J. W. Mark, and X. Shen, "A renewal theory based analytical model for the contention access period of IEEE 802.15.4 MAC," *IEEE Trans. Wireless Commun.*, vol. 7, no. 6, pp. 2340–2349, Jun. 2008.
[34] I. Butun, S. D. Morgera, and R. Sankar, "A survey of intrusion detection systems in wireless sensor networks," *IEEE Commun. Surv. Tuts.*, vol. 16, no. 1, pp. 266–282, First Quart. 2014.
[35] D. Shila, Y. Cheng, and T. Anjali, "Mitigating selective forwarding attacks with a channel-aware approach in WMNs," *IEEE Trans. Wireless Commun.*, vol. 9, no. 5, pp. 1661–1675, May 2010.
[36] NS-3: A Discrete-Event Network Simulator, [Online]. Available: http://www.nsnam.org
[37] V. Shnayder, M. Hempstead, B. Chen, G. Allen, and M. Welsh, "Simulating the power consumption of large-scale sensor network applications," in *Proc. ACM SenSys*, 2004, pp. 188–200.
[38] O. Tremblay, L. Dessaint, and A. Dekkiche, "A generic battery model for the dynamic simulation of hybrid electric vehicles," in *Proc. IEEE Veh. Power Propul. Conf.*, 2007, pp. 284–289.
[39] A. Tsertou and D. Laurenson, "Revisiting the hidden terminal problem in a CSMA/CA wireless network," *IEEE Trans. Mobile Comput.*, vol. 7, no. 7, pp. 817–831, Jul. 2008.

**Xianghui Cao** (S'08–M'11) received the B.S. and Ph.D. degrees in control science and engineering from Zhejiang University, Hangzhou, China, in 2006 and 2011, respectively.

From July 2012 to July 2015, he was a Senior Research Associate with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL, USA. Currently, he is an Associate Professor with the School of Automation, Southeast University, Nanjing, China. His research interests include wireless sensor and actuator networks, wireless network performance analysis, and network security.

Dr. Cao served as a Publicity Co-chair for ACM MobiHoc 2015, Symposium Co-chair for IEEE/CIC ICCC 2015, and TPC Member for a number of conferences. He serves as an Associate Editor for the *KSII Transactions on Internet and Information Systems*, *Security and Communication Networks* and the *International Journal of Ad Hoc and Ubiquitous Computing*, and a Guest Editor for the *Asian Journal of Control*. He was a recipient of the Best Paper Runner-Up Award from ACM MobiHoc 2014.

**Devu Manikantan Shila** (S'07) received the M.S. and Ph.D. degrees in computer engineering from the Illinois Institute of Technology, Chicago, IL, USA, in 2007 and 2011, respectively.

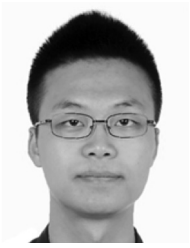
She is currently with the United Technologies Research Center, Hartford, CT, USA, as a Staff Research Scientist and a Principal Investigator in the cyber security domain. She mainly conducts research in the area of analyzing, designing, and developing secure methodologies for embedded systems. She has authored heavily along the lines of interference and secure-aware algorithm design and analysis. She has security expertise in a variety of topics including reverse engineering, modeling risks and vulnerabilities at the physical, system, network and application level, and design and theoretical analysis of secure protocols and algorithms. She has research and development expertise in both the design and analysis of secure and resilient algorithms and protocols for wireless and embedded systems.

**Yu Cheng** (S'01–M'04–SM'09) received the B.E. and M.E. degrees in electronic engineering from Tsinghua University, Beijing, China, in 1995 and 1998, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2003.

From September 2004 to July 2006, he was a Postdoctoral Research Fellow with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada. Since 2006, he has been with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL, USA, where he is currently an Associate Professor. His research interests include next-generation Internet architectures and management, wireless network performance analysis, network security, and wireless/wireline interworking.

Dr. Cheng served as a Co-Chair for the Wireless Networking Symposium of IEEE ICC 2009, a Co-Chair for the Communications QoS, Reliability, and Modeling Symposium of IEEE GLOBECOM 2011, a Co-Chair for the Signal Processing for Communications Symposium of IEEE ICC 2012, a Co-Chair for the Ad Hoc and Sensor Networking Symposium of IEEE GLOBECOM 2013, and a Technical Program Committee (TPC) Co-Chair for WASA 2011, ICNC 2015, and IEEE/CIC ICCC 2015. He is a founding Vice Chair of the IEEE

ComSoc Technical Subcommittee on Green Communications and Computing. He is an Associate Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY and the New Books and Multimedia Column Editor for IEEE NETWORK. He was the recipient of the Best Paper Award from the conferences QShine 2007 and IEEE ICC 2011, and the Best Paper Runner-Up Award from ACM MobiHoc 2014. He was the recipient of the National Science Foundation (NSF) CAREER Award in 2011 and the IIT Sigma Xi Research Award in the junior faculty division in 2013.



**Zequ Yang** received the B.Eng. degree in automation from Zhejiang University, Hangzhou, China, in 2012, and is currently working toward the Ph.D. degree at Zhejing University.

He is a member of the Group of Networked Sensing and Control (IIPC-NeSC), State Key Laboratory of Industrial Control Technology. His research interests include security issues in wireless sensor networks, privacy protection in smart grid, control, and optimization in cyber-physical system.

**Yang Zhou** (S'14) received the B.S. degree in communication engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2013, and is currently working toward the M.S. degree at Zhejiang University, Hangzhou, China.

He is currently with the Department of Control Science and Technology, Zhejiang University. His research interests include mobile computing and wireless rechargeable sensor network.

**Jiming Chen** (M'08–SM'11) received the B.Sc. and Ph.D. degrees in control science and engineering from Zhejiang University, Hangzhou, China, in 2000 and 2005, respectively.

He was a Visiting Researcher with INRIA, Rocquencourt, France, in 2006, the National University of Singapore, Singapore, in 2007, and the University of Waterloo, Waterloo, ON, Canada, from 2008 to 2010. He is currently a Full Professor with the Department of Control Science and Engineering, the Coordinator of the Group of Networked Sensing and Control, State Key Laboratory of Industrial Control Technology, and the Vice Director of the Institute of Industrial Process Control, Zhejiang University. His research interests include sensor networks and networked control.

Dr. Chen serves as an Associate Editor for several international journals including the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEM, IEEE NETWORK, IEEE TRANSACTIONS ON CONTROL OF NETWORK SYSTEMS, etc. He was a Guest Editor for the IEEE TRANSACTIONS ON AUTOMATIC CONTROL.