Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2006, Article ID 93830, Pages 1–12
DOI 10.1155/WCN/2006/93830

MAC Security and Security Overhead Analysis in the IEEE 802.15.4 Wireless Sensor Networks

Yang Xiao,[1] Hsiao-Hwa Chen,[2] Bo Sun,[3] Ruhai Wang,[4] and Sakshi Sethi[5]

[1] *Department of Computer Science, The University of Alabama, Box 870290, Tuscaloosa, AL 35487-0290, USA*
[2] *Institute of Communications Engineering, National Sun Yat-Sen University, Kaohsiung 804, Taiwan*
[3] *Department of Computer Science, Lamar University, Beaumont, TX 77710, USA*
[4] *Department of Electrical Engineering, Lamar University, Beaumont, TX 77710, USA*
[5] *Equifax Inc., 1505 Windward Concourse, Alpharetta, GA, USA*

Received 11 October 2005; Revised 14 May 2006; Accepted 17 May 2006

Sensor networks have many applications. However, with limited resources such as computation capability and memory, they are vulnerable to many kinds of attacks. The IEEE 802.15.4 specification defines medium access control (MAC) layer and physical layer for wireless sensor networks. In this paper, we propose a security overhead analysis for the MAC layer in the IEEE 802.15.4 wireless sensor networks. Furthermore, we survey security mechanisms defined in the specification including security objectives, security suites, security modes, encryption, authentication, and so forth. Then, security vulnerabilities and attacks are identified. Some security enhancements are proposed to improve security and to prevent these attacks such as same-nonce attack, denial-of-service attack, reply-protection attack, ACK attack, and so forth. Our results show that, for example, with 128-bit key length and 100 MIPS, encryption overhead is 10.28 μs per block, and with 100 MIPS and 1500-byte payload, the encryption overhead is as high as 5782.5 μs.

1. INTRODUCTION

Sensor networks have many applications, such as monitoring and surveillance. Each sensor network is built with many sensor nodes, which are small, inexpensive, and battery-powered with limited energy, computation, memory, and communication capacities. The IEEE 802.15.4 specification [1] defines medium access control (MAC) layer and physical (PHY) layer targeted for the low rate wireless personal area networks (LR-WPAN) using short distance applications with low power consumption and low cost communication networks, particularly the short-range applications such as wireless sensor networks, residential/industrial setting networks, and so forth. Applications of IEEE 802.15.4 include light control systems, environmental and agricultural monitoring, consumer electronics, energy management and comfort functions, automatic meter reading systems, industrial applications, and alarm and security systems [2]. Light control systems include power outlets, dimmers, switches, and remote controls; environmental and agricultural monitoring include reading temperature, carbon dioxide level, humidity, and vibration strength; consumer electronics include remote controls, set-top boxes, and PC-peripherals; energy management and comfort functions include thermostats, HVAC (heating, ventilation, air-conditioning), and control of blinds/shades/rollers/windows; automatic meter reading systems may need to monitor electricity, gas, and water; industrial applications include monitoring and control of wireless sensor networks in general; alarm and security systems include smoke detectors, burglary and social alarms, access control, and water leakage systems [2]. The IEEE 802.15.4 specification supports many applications with MAC security requirements. However, if the networks are not secured, confidentiality, privacy, and integrity could be compromised.

Security functionalities providing basic security services and interoperability among all devices are defined in the MAC, and limited by the diverse range of potential applications of the IEEE 802.15.4 specification [1]. The security services include maintaining an access control list (ACL) and using advanced encryption standard (AES) to protect frame transmissions. These security services are optional, and final security policies are defined by the higher layers, which provide key management and device authentication. The IEEE 802.15.4 specification does not include key management and device authentication schemes.

There are some security services that are required in data communication. The data frames should be confidential and protected from being modified by any unauthenticated/unauthorized devices. Any received message is protected from being replayed and the devices should be capable of distinguishing the devices that are willing and authenticated to communicate.

This paper studies the security issues in the IEEE 802.15.4 specification. The paper particularly focuses on the various security suites consisting of the symmetric key encryption algorithm, modes of operations, and the length of message integrity code (MIC) bits. These security suites provide various security services. The symmetric cryptographic algorithm adopts AES. There are three modes of operation: counter mode (CTR mode) that is used for providing the encryption, cipher block chaining message authentication code (CBC-MAC) mode that provides just the authentication, and the CTR+CBC-MAC (CCM) mode that combines both the CTR and the CBC-MAC mode of operations and provides both the authentication and encryption of the message. Finally, the possible MIC (message authentication code) bit lengths can be 32, 64, and 128 bits.

Furthermore, the paper presents a detailed description of the attacks and the vulnerabilities presented in the specification, such as same-nonce attack, denial-of-service attack, ACK attack, reply-protection attack, and so forth. These vulnerabilities allow an intruder to get into the communication channel and access the data. Then, we propose some enhanced security mechanisms to the security services to prevent these attacks such as same-nonce attack, denial-of-service attack, reply-protection attack, ACK attack, and so forth.

Finally, for the most important contribution, we present a MAC security overhead analysis, in which, we obtain some useful observations and conclusions. In particular, we answer the following question: how fast should the processing speed of a device be for decryption under the current IEEE 802.15.4 parameters?

The rest of the paper is organized as follows. Section 2 briefly introduces the IEEE 802.15.4 specification in general including introduction to device types, architecture, and possible network topologies. Sections 3 and 4 survey the security services and modes of operations, respectively. Attacks and vulnerabilities are identified in Section 5. Security enhancements and recommendations are presented in Section 6. A MAC security overhead analysis is provided in Section 7. Finally, we conclude our paper in Section 8.

2. IEEE 802.15.4

The IEEE 802.15.4 specification favors low-cost and low-power LR-WPANs for a wide variety of applications requiring short-range communications. Low power consumption is one of the major design issues in the IEEE 802.15.4 specification to maximize battery life, assuming that the amount of data transmitted is small and transmissions are infrequent [3]. The frame structure is designed with minimal overhead.

This section gives an overview of the IEEE 802.15.4 that includes its basic component-devices, network topology, the PHY layer, and the MAC layer.

2.1. Devices

Personal area network (PAN) coordinator is a coordinator that is the principal controller of a PAN, controls the network, and defines the parameters of the network. An IEEE 802.15.4 network has exactly one PAN coordinator. There are two types of devices described in the specification that communicate together to form different network topologies: *full function device* (FFD) and *reduced function device* (RFD). An FFD is a device capable of operating as a coordinator or device and implementing the complete protocol set. An RFD is a device operating with a minimal implementation of the IEEE 802.15.4 protocol. An RFD can connect to only an FFD whereas an FFD can connect to both FFDs and RFDs. The FFD that acts as a PAN coordinator is the main controller of the network and can initiate a communication, terminate it, and route it around the network. At the physical level, an FFD and an RFD distinguish themselves with capabilities of hardware platform. An RFD can perform a logical role of end devices with extremely simple applications such as a light sensor and a lighting controller, whereas an FFD can take up the role of a coordinator and a router.

2.2. Network topology

The RFDs and FFDs combine together to form the two types of network topologies: star topology and peer-to-peer topology, shown in Figure 1. In the star topology, the PAN coordinator acts as the initiation point for the network and other FFDs and RFDs connect to it. Communications are performed between RFDs/FFDs and the PAN coordinator, which is in charge of managing all the star functionality. In the peer-to-peer topology, every FFD can communicate with other FFDs including a PAN coordinator. Peer-to-peer topology allows more complex network formations to be implemented, for example, ad hoc and self-configuring networks. Each PAN coordinator has a unique identifier or the link key through which the other devices can communicate with each other.

2.3. Physical layer

The IEEE 802.15.4 specification supports two PHY options based on direct sequence spread spectrum (DSSS), which allows the use of low-cost digital IC realizations [3]. The PHY adopts the same basic frame structure for low-duty-cycle low-power operation, except that the two PHYs adopt different frequency bands: low-band (868/915 MHz) and high band (2.4 GHz). The low band adopts *binary phase shift key* (BPSK) modulation and operates in the 868 MHz band in Europe offering one channel with a raw data rate of 20 kbps and in the 915 MHz ISM band in North America offering 10 channels with a raw data rate of 40 kbps [1, 3]. The high band adopts *offset quadrature phase shift key* (O-QPSK)

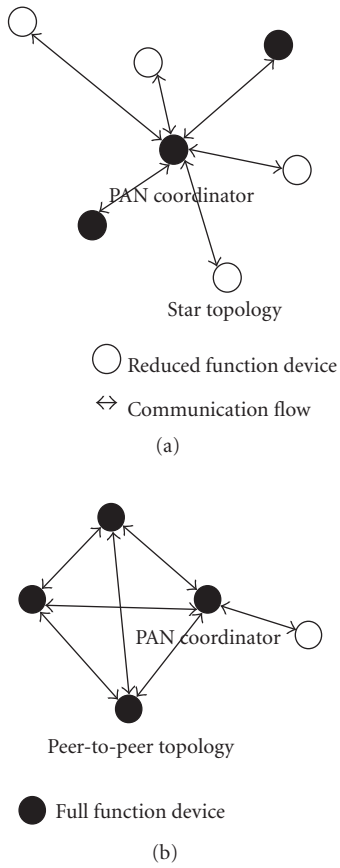


Figure 1: Network topologies.

modulation, operates in 2.4 GHz $\sim$ 2.483 GHz, and is a part of ISM band, which is available almost worldwide, and has 16 channels with channel spacing of 5 MHz with a raw data rate of 250 kbps. The PHY layer uses a common frame structure, containing a 32-bit preamble, a frame length, and a 2 $\sim$ 127 bytes payload field.

2.4. Medium access control

The IEEE 802.15.4 MAC layer is used for a reliable and single hop communication among the devices, providing access to the physical channel for all types of transmissions and appropriate security mechanisms. The MAC uses acknowledged frame delivery, performs frame validation, maintains network synchronization, controls the association/disassociation, manages device security, and schedules the time slots.

The specification allows the optional use of a superframe structure for applications requiring dedicated bandwidth with guaranteed delay. The PAN coordinator defines the format of the superframe, which includes a beacon frame, the contention access period (CAP), and the contention free period (CFP). The total length of the contention access period (CAP) and the contention free period (CFP) is 16 equally sized time slots: The time slots for the CFP are called guaranteed time slots (GTS) and are administered by the PAN coordinator. The CAP adopts the carrier sense multiple access/ collision avoidance (CSMA/CA) mechanism.

3. SECURITY SERVICES

IEEE 802.15.4 devices can operate in either secured mode or nonsecurity mode [1]. Devices operating in the secured mode adopt AES with security services including access control, data encryption, frame integrity, and sequential freshness. Keys are assumed to be provided by higher layer processes, and key management and key establishment are not specified. Access control is to allow a device to select the other devices to communicate with. In IEEE 802.15.4, a device maintains an *access control list* (ACL) from which it expects to receive frames, if the access control service is provided. Data encryption is achieved by using a symmetric cipher, that is, AES, provided on beacon payloads, command payloads, and data payloads. Frame integrity, provided on data frames, beacon frames, and command frames, is to use a message integrity code (MIC) to protect data from being modified without the key as well as to provide assurance that data come from the sender with the key. Sequential freshness is to use an ordered sequence of frames to reject replayed frames by comparing the last known freshness value with the freshness value in a newly arrived frame to update the freshness value or to signal a failed check message. Furthermore, several security suites are defined to achieve different purposes and different levels of security.

In this section, we introduce security objectives, security modes, and security suites in the following sections. In the next section, we will introduce modes of operations.

3.1. Security objectives

There are four objectives of security services: access control, data encryption, frame integrity, and sequential freshness. They are explained as follows.

(i) Access control

It provides a list (ACL) of valid devices from which the device can receive the frames. This mechanism prevents the unauthorized devices to communicate to the network.

(ii) Data encryption

It prevents messages from the unauthorized access via encryption algorithms. Only the devices that share the secret key can decrypt the messages and communicate.

(iii) Frame integrity

This objective is to prevent changes to be made by an invalid intruder and to provide assurance that the messages from the source device have not been manipulated by the invalid intruder.

(iv) Sequential freshness

This objective is to prevent the replayed message from being accepted by the receiver and to ensure that the frame that has

Table 1: Security suites.

Security suite name	Security services			
	Access control	Data encryption	Frame integrity	Sequential freshness
None	—	—	—	—
AES-CTR	X	X	—	X
AES-CCM-128	X	X	X	X
AES-CCM-64	X	X	X	X
AES-CCM-32	X	X	X	X
AES-CBC-MAC-128	X	—	X	—
AES-CBC-MAC-64	X	—	X	—
AES-CBC-MAC-32	X	—	X	—

Figure 2: ACL entry format.

arrived is not a replayed one. This is achieved through means that a receiver checks the recent counter and rejects the frame which has the counter value equal to or less than the previous obtained counter value.

3.2. Security mode

Three security modes are defined in the specification to achieve different security objectives: unsecured mode, ACL mode, and secured mode. Figure 2 shows the format of the ACL entry, and an ACL list includes multiple ACL entries. In Figure 2, the address field is composed of the source and the destination addresses. The last initial vector (IV) and the replay counter are the same except that the last IV is used by the source device when it sends the frame, and the replay counter is used by the destination device to maintain the high water mark to avoid the replay attack. The key is a symmetric key shared between the devices.

Three security modes are defined in the specification to achieve different security objectives: unsecured mode, ACL mode, and secured mode. We explain the three security modes as follows.

(i) Unsecured mode

This mode is for those low cost applications that do not require any security at all. In other words, no security service is provided.

(ii) ACL mode

Each device maintains its ACL. In the ACL mode, limited security services for communications are provided via the ACL. This mode allows the receiving of the frames from only those devices that are present in the device's ACL. If a frame does not come from a device listed in the ACL, the frame will be rejected. However, cryptographic protection is not provided in this mode. In other words, most of fields in the ACL, such as security suite, key, last initial vector (IV), and replay counter, are not used in this mode.

(iii) Secured mode

The secured mode provides all the security services according to the defined security suite. It provides the confidentiality of the frame along with the message integrity, access control, and sequential freshness. It uses all the fields in the ACL entry format. The secured mode is implemented by the security suite listed in the ACL entry explained in the next section.

3.3. Security suite

Several security suites are defined in the IEEE 802.15.4 specification and listed in Table 1, where a security suite includes security mechanisms defined for MAC frames, including the symmetric algorithm, mode, and integrity code bit length. If the security mode is enabled, the security suite is used, and the MAC checks the ACL entry for the suite and provides the security services accordingly.

Table 1 shows the entire possible security suites. There are three parts in a security suite name. The first part indicates the symmetric cryptographic algorithm, that is, AES. The second part is the mode of operation in which the security suite works. The last part indicates the message integrity code bit length. The contents in Table 1 indicate whether a security suite is applied to the security objectives, including access control, data encryption, frame integration, and sequential freshness.

The AES encryption algorithm is used in this specification and is defined in Federal Information Processing Standard (FIPS) Publication 197 [4] used by US Government organizations to protect sensitive and unclassified information. NIST selected Rijndael as the AES algorithm in 2001, where Rijndael was developed and submitted by two cryptographers, Joan Daemen and Vincent Rijmen. The AES is an official US Government standard since May 26, 2002, with features such as better security, performance, efficiency, ease of implementation, and flexibility. Rijndael has good performance in both hardware and software with low memory requirements, and it is also against power and timing attacks.

The AES is to replace data encryption standard (DES) [5], but NIST anticipates that Triple DES will remain an approved algorithm for US Government use for the foreseeable future. The AES specifies three key sizes: 128, 192, and 256 bits. In comparison, the DES keys are 56 bits long, which means there are approximately 7.2×1016 possible DES keys. Thus, there are on the order of 1021 times more AES 128-bit keys than DES 56-bit keys. In the IEEE 802.15.4 specification, the AES adopts 128 bit block size and 128 bits of key length.

Nonsecurity mode in Table 1 does not provide any security services at all. The counter mode (CTR) generates a key stream using a block cipher with a given key and nonce, and performs an exclusive OR (XOR) of the key stream with the plaintext and integrity code, where a nonce can be a time stamp, a counter, or a special marker. The AES-CTR means that the CTR uses AES as the block cipher, and provides access control, data encryption, and optional sequential freshness.

The cipher block chaining with message authentication code (CBC-MAC) generates an integrity code using a block cipher in the CBC mode, and computes message authentication code based on the message that includes the length of the authenticated data. The integrity code is computed at the receiver side and compared with the received integrity code.

The CTR combines the CBC-MAC to become the CTR-CBC-MAC (CCM) providing both encryption and authentication mechanisms. The CCM generates an integrity code followed by the encryption of plaintext data and the integrity code such that the encrypted data and the encrypted integrity code are produced. The authentication operation uses a nonce concatenated with padded authentication data and padded plaintext data, if present, to generate an integrity code via a block cipher in the CBC mode. The integrity code is recomputed by the receiver and compared with the received integrity code for verification. A key stream is generated for encryption of a block cipher in CTR with a given key and nonce such that the key stream is XORed with the integrity code and plaintext, if present. At the receiver end, the same key stream is generated for decryption such that the XOR of the key stream with the ciphertext obtains the plaintext and integrity code.

The AES-CCM provides the encryption and data integrity and comes with three MIC bit options: 32, 64, and 128 bits. The AES-CBC-MAC provides the data integrity but not the data encryption. It also comes with three options: 32, 64, and 128 bits. Security suites work in the secured mode, that is, security enabled bit is 1. The MAC checks the ACL entries and the corresponding security suite from the security suite field to use.

The MAC PIB (PAN information base) includes the security information, and the formats depend on the selected security suite. The AES is the block cipher for the one among the CTR encryption, the CCM encryption and authentication, and the CBC-MAC authentication. A frame counter is included in the payload and incremented each time a secure frame is transmitted. The frame counter does not roll over to ensure freshness. The key sequence counter can be used if the frame counter is exhausted.

The AES-CCM is for both encryption and authentication, the AES-CBC-MAC is for authentication only, and the AES-CTR is for encryption only.

4. MODES OF OPERATIONS

We explain these modes of the operation adopted in IEEE 802.15.4 in detail.

4.1. CTR mode

In the CTR mode, counters are encrypted with a block cipher to produce a sequence of output blocks that are XORed with the plaintext to produce the ciphertext. All counters must be different in all of the encrypted messages that are encrypted under the given key. Forward cipher (CIPHk) is applied to input block known as counters to produce output blocks (O) which are then XORed with the plaintext (P) to produce the encrypted data or ciphertext (C). Let the counters be $T_1, T_2, \ldots, T_n$. Therefore, the CTR encryption and decryption are $\{O_j = \mathrm{CIPH}_k(T_j)$ for $j = 1, 2, \ldots, n$; $C_j = P_j \oplus O_j$ for $j = 1, 2, \ldots, n-1$; $C_n = P_n \oplus \mathrm{MSB}_u(O_n)\}$ and $\{Oj = \mathrm{CIPH}_k(T_j)$ for $j = 1, 2, \ldots, n$; $P_j = C_j \oplus O_j$ for $j = 1, 2, \ldots, n-1 : P_n = C_n \oplus \mathrm{MSB}_u(O_n)\}$, where $C = C_1 \| C_2 \| C_3 \| \cdots \| C_n$; $P = P_1 \| P_2 \| P_3 \| \cdots \| P_n$; $O = O_1 \| O_2 \| O_3 \| \cdots \| O_n$.

The last block may be a partial block of u bits, most significant bit (MSB) u bits of the last output block are used and the remaining b–u bits of the last output block are discarded. In both the CTR encryption and the CTR decryption, the forward cipher functions can be performed in parallel. The CTR mode of operation is shown in Figure 3.

4.2. CBC-MAC mode

The CBC-MAC mode uses a block cipher with a key K to encrypt input vectors of the block size to output vectors of the block size. Let D and O denote any input vector and its output block: $O = E_K(D)$. Let $D = D_1 \| D_2 \| \cdots \| D_n$ and $O = O_1 \| O_2 \| \cdots \| O_n$. The CBC-MAC mode is defined as $O_1 = E_K(D_1)$, $O_2 = E_K(D_2 \oplus O_1)$, $O_3 = E_K(D_3 \oplus O_2), \ldots, O_n = E_K(D_n \oplus O_{n-1})$.

The final block is appended with zeros if it is a partial block. The MIC is the leftmost M bits of O_n, where $32 < M = 8\,\mathrm{h} < 128$ and h is integer. The CBC-MAC is not for the encryption, but for authentication of data, that is, to check the integrity of the data message by finding out whether the MIC computed by the sender matches that computed by the receiver or not. The CBC-MAC mode of operation is shown in Figure 4.

4.3. The CCM mode

The CTR-CBC-MAC (CCM) [1] is an authentication-and-encryption block cipher mode, using a block cipher with 128 bit block size, for example, AES in IEEE 802.15.4. There are three inputs for the CCM mode: *data* payload to be both encrypted and authenticated, the *associated data* (e.g., header, etc.) to be authenticated but not encrypted, and the

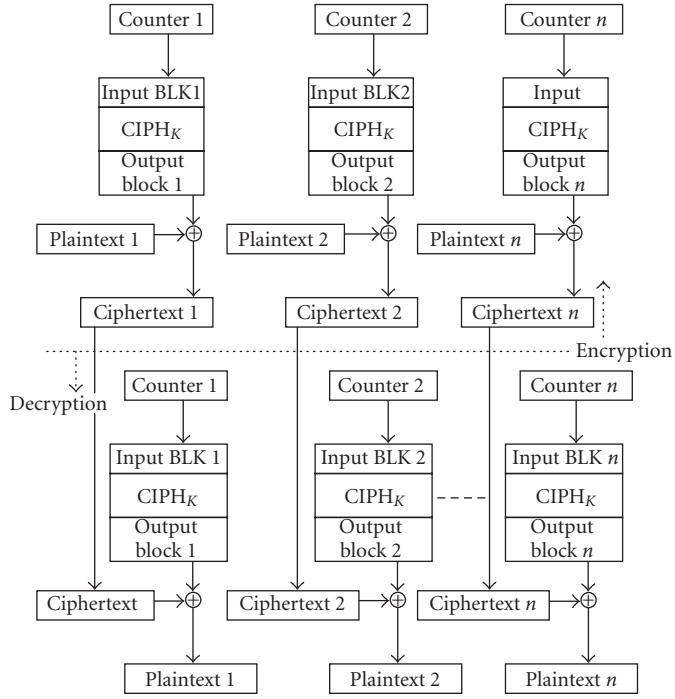


Figure 3: The CTR mode.

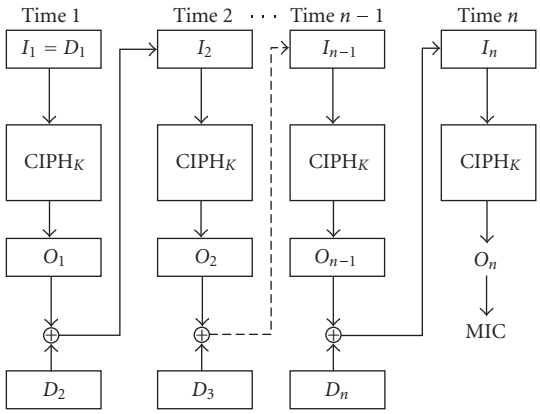


Figure 4: The CBC-MAC mode.

nonce to be assigned to the payload and the associated data [6]. The CCM provides both the authentication and encryption and uses the techniques of the CTR for encryption and the CBC-MAC for authentication. The CCM is composed of two methods: *generation-encryption* that requires the generation of the MIC first and then the encryption, and *decryption-verification* that requires first the decryption of the ciphertext and then the verification of the MIC.

A sender needs an input of $\{K, N, m, a\}$, where K is the AES encryption key, N is a nonce of $15 - L$ octets, m is the message consisting of a string of $l(m)$ octets where $0 \leq l(m) < 2^{8L}$ to be encoded in a field of L octets, and a is additional authenticated data consisting of a string of $l(a)$ octets where $0 \leq l(a) < 2^{64}$. Additional data a are authenticated, but not encrypted, and are not included in the output of this mode. Furthermore, they can be used to authenticate plaintext headers that affect the interpretation of the message.

For authentication, the authentication field T is computed using the CBC-MAC. Let $B_0, B_1, \ldots, B_n$ denote a sequence of blocks for the CBC-MAC. We have $B_0 = \{F, N, l(m)\}$, where F is one octet in length, N is the nonce with $15 - L$ octets in length, and $l(m)$ has L octets in length. We have $F = \{\text{Reserved-bit, Adata-bit,}(M-2)/2, L-1\}$, where *Reserved-bit* and *Adata-bit* are both one bit in length, and $(M - 2)/2$ and $L - 1$ are both 3 bits in length. The *Adata-bit* is 0 if $l(a) = 0$ and 1 if $l(a) > 0$. The CCM mode has two parameters to select: M, the size of the authentication field, and L, the size of the length field. M (3 bits) can be 4, 6, 8, 10, 12, 14, and 16 octets, has an encoding field of $(M - 2)/2$, and involves a trade-off between message expansion and the probability that an attacker can undetectably modify a message. L (3 bits) can be 2 to 8 octets, has an encoding field of $L - 1$, and requires a trade-off between the maximum message size and the size of the nonce based on applications.

If $l(a) > 0$, that is, Adata-bit =1, one or more blocks of authentication data are added including $l(a)$ and a encoded in a reversible manner. If $0 < l(a) < 2^{16} - 2^8$, the length field is encoded as 2 octets. If $2^{16} - 2^8 \leq l(a) < 2^{32}$, the length field is encoded as 6 octets consisting of the octets $0 \times$ ff, $0 \times$ fe, and 4 octets encoding $l(a)$. If $2^{32} \leq l(a) < 2^{64}$, the length field is encoded as 10 octets consisting of the octets $0 \times$ ff, $0 \times$ ff, and 8 octets encoding $l(a)$.

The blocks encoding a are formed by concatenating the string that encodes $l(a)$ with a and splitting the result into 16 octet blocks, padding the last block with zeros if necessary. These blocks are appended to the first block B_0. After the (optional) additional authentication blocks have been added, the message blocks are added. The message blocks are formed by splitting the message m into 16 octet blocks, padding the last block with zeros if necessary. If m is an empty string, no blocks are added. The result is a sequence of blocks $B_0, B_1, \ldots, B_n$. The CBC-MAC is now computed by $X_1 = E_K(B_0)$; $X_{i+1} = E_K(X_i \oplus B_i)$ for $i = 1, \ldots, n$; T = first-M-octets(X_{n+1}).

The CTR mode is used for encryption, and key stream blocks are defined as follows. $Si = E_K(Ai)$ for $i = 0, 1, 2, \ldots$, where $Ai = \{F, N, \text{Counter } i\}$, and $F = \{\text{Reserved-bits (2 bits)}, 0\ (3\text{ bits}), L-1\ (3\text{ bits})\}$, N is the nonce with $15 - L$ octets in length, and *Counter* has L octets in length. The message is encrypted by XORing the octets of message $m \oplus S$, where $S = l(m)$ octets of $S1 \| S2 \| S3, \ldots$, and note that $S0$ is not used to encrypt the message. The authentication value is obtained as follows: $U = T \oplus$ first-M-octets$(S0)$. The ciphertext is $m \oplus S \| U$.

For decryption, the receiver needs the encryption key K, the nonce N, the additional authenticated data a, and the encrypted and authenticated message c. First, the key stream is generated to recover the message m and the value T. The message and additional authentication data are then used to recompute the CBC-MAC value and check T. If the T value is not correct, the receiver will not reveal any information except for the fact that T is incorrect. In particular, the receiver will not reveal the decrypted message, the value T, or any other information.

5. ATTACKS AND VULNERABILITIES

In the IEEE 802.15.4 specification, there are some security vulnerabilities that have been described in [7–9]. In this section, we present a more detailed description of several kinds of attacks and vulnerabilities.

5.1. Same-nonce attack

Same-nonce attack [8] is defined as follows. There is a chance that in a sender's ACL entry table, there are entries with the same key and the same nonce. If such a thing happens, a security attack is possible. Note that the nonce is also used as the frame counter. Assume that there are two plaintexts ($P1, P2$) and two ciphertexts ($C1$ and $C2$) using the same key (K) and the same nonce (N). Also assume that an adversary can obtain $C1$ and $C2$, but cannot obtain $P1$ or $P2$. Then the adversary can obtain $P1 \oplus P2 = C1 \oplus C2$ since the counters are the same and the keys are the same although the adversary does not know the key. The adversary may obtain much useful information from $P1 \oplus P2$. The same nonce occurs in many situations such as power failure, sleep mode, and so forth. Same keys happen in many situations too such as using broadcasting key, grouping key, and so forth.

5.2. Replay-protection attack

In the IEEE 802.15.4 specification, the replayed message is prevented by the replay protection mechanism, that is, sequential freshness. This is achieved by which a receiver checks the recent counter and rejects the frame which has the counter value equal to or less than the previous obtained counter. However, this replay protection mechanism is subject to another attack, called replay-protection attack, which is one kind of denial-of-service attacks. It is very easy to launch replay-protection attacks as follows. An adversary can send many frames containing different large frame counters to a receiver who performs replay protection and raises the replay counter up as the largest frame counter in the receiver so far. Then, when a normal station sends a frame with a reasonable size of frame counter that is smaller than the replay counter maintained at the receiver, the frame will be discarded for the replay-protection purpose. In other words, the service is denied.

5.3. ACK attack

There is no integrity protection provided on ACK frames. When a sender sends a frame, it can request an ACK frame from the receiver by setting the bit flags in the outgoing data frame.

The eavesdropper can forge the ACK frame by using the unencrypted sequence number from the data frame. If an adversary does not want a particular frame to be received by the receiver, it can send interference to the receiver at the same time when the sender is sending the data frame. This leads to the rejection of the frame. The adversary can then send a forged ACK frame fooling the sender that the receiver successfully received the frame. Therefore, a sender cannot be sure if the received frame is coming from the receiver or another node even if the receiver received the ACK frame.

6. SECURITY ENHANCEMENTS

In this section, we propose some security enhancements to prevent the attacks described above.

6.1. Separating nonce from frame counter

We believe that the current approach that nonce serves as both IV and the frame counter is a bad design and causes some vulnerability.

We propose to separate nonce from the frame counter so that two fields, nonce and frame counter, are both used. The drawback is that an additional field is added, but security is much enhanced.

6.2. Randomly generating nonces

Since a nonce is separated from the frame counter, the nonce can be generated using a random generation algorithm instead of increasing the counter/nonce one by one each time.

6.3. Using time stamp as the frame counter

We notice that same-nonce attack, replay-protection, and denial-of-service are all related to the frame counter.

The frame counter is used for sequential freshness, that is, the replayed message is prevented by the replay protection mechanism. Furthermore, the frame counter potentially causes problems when nodes are in sleep mode or the power of nodes is temporarily failed, and so forth.

We propose to use timestamp as the sequential freshness. The sequential freshness is achieved by which a receiver checks the recent timestamp obtained from the sender and rejects the frame which has the timestamp equal to or less than the previous obtained timestamp. Furthermore, there is not relay counter to be raised up. The drawback of this approach is that the field length is larger. Since the IEEE 802.15.4 specification defines beacon frames which help clock synchronization, using timestamp can prevent replay-protection attack as follows. Whenever the sender receives a frame with a timestamp, it compares this timestamp with the current time. If the current time is much smaller than the timestamp, the sender believes that this is an attack, and rejects the frame. Therefore, the recorded timestamp has never been raised up to a value so that replay-protection attack or denial of service attack cannot be launched.

Furthermore, when a sensor just wakes up or obtains power supply after a power failure, it contacts the coordinator, synchronizes the clock with beacon frames received, and raises all the time stamps up to the current time.

In such a way, both replay-protection attack and denial-of service attack can be prevented.

6.4. Using MIC for ACK

For ACK frame, we propose to append MIC at the end of ACK frame, where MIC is obtained by the authentication algorithm AES-CBC-MAC. The authenticated field is the whole ACK frame.

6.5. Dynamically dividing nonce spaces

For the broadcasting key and group keys, it may have multiple same key entries in the ACL table. In order to prevent the same-nonce attack, nonce space is divided into multiple groups so that different entries with the same key will use different space of nonce values (also chosen randomly). This feature plus timestamp can prevent same-nonce attack and other attacks.

6.6. Eliminating the key sequence counter

In practice, the key sequence counter is always zero and of no use. It generates one byte overhead in each security-enabled frame. In order to increase the air efficiency, reduce the size of the ACL table, and simplify the processing in the CCM mode, it is recommended to eliminate key sequence counter.

6.7. Tracking frame counter for each device

To present the replay-protection attack, each station keeps track of the frame counter for each device sending to it. However, this scheme may not be very robust when there is a failure such as a power failure or restart. Furthermore, it is a little awkward to maintain the consistency.

6.8. CCM* mode

In [10], the author introduces a CCM* mode, in which a counter determined from the frame counter of the source device is used to provide frame freshness and to prevent the replay-protection attack. For each node to which a device sends or receives secured frames, an ACL entry is created in the MAC PIB, containing the implicit or explicit address of the entity and the associated corresponding security material including an AES key, a frame counter for outgoing frames, and an external frame counter for incoming frames [10]. If it is explicit, it contains a key identifier. The AES symmetric key is 16 octets to secure incoming and outgoing frames; the frame counter for outgoing frames is used by a device when originating a frame; and the external frame counter for incoming frames is used by a device to verify freshness of incoming frames [10]. This counter is increased each time when a secure frame is transmitted, but it will not roll over to ensure that the CCM* nonce is unique and to ensure freshness or to detect duplicates.

The IEEE 802.15.4 security suite includes three components, the AES-CCM is for encryption and authentication, the AES-CBC-MAC is for authentication only, and the AES-CTR is for encryption only. There are several problems as follows [10]: these three separate components require a larger implementation (counted in gates or code) than the unified CCM* implementation; switching between these modes compromises security unless separate keys are kept, but it requires additional storage; and the CBC-MAC does not provide freshness and is subject to replay attacks. Therefore, when replacing security suite, the AES-CCM with the AES-CCM*, backward compatibility needs to be considered such as approaches of negotiating security as well as falling back to "no security."

7. SECURITY OVERHEAD ANALYSIS

In this section, we provide a security overhead analysis for the MAC of IEEE 802.15.4.

7.1. AES overhead analysis

Let $4B$, $4K$, and R denote the block size (in bytes), the key length (in bytes), and the number of rounds of Rijndael, respectively. Let T_{and}, T_{or}, and T_{shift} denote the numbers of processing cycles required for performing basic operations of a byte-wise AND, a byte-wise OR, and a byte-wise SHIFT, respectively.

Encryption includes an initial stage, R rounds, and a final stage. From [11], we can obtain the following calculations.

(i) To implement the initial stage, operations of $8B$ byte-wise ANDs and $4B$ byte-wise ORs are needed.
(ii) To implement one round, operations of $46B$ byte-wise ANDs, $31B + 12$ byte-wise ORs, and $64B+ 96$ binary SHIFTs are needed.
(iii) To implement the final stage, operations of $8B$ byte-wise ANDs, $7B$ byte-wise ORs, and $3B$ byte-wise SHIFTs are needed.

Therefore, the total number of processing cycles for encrypting a block is given as follows:

$$\begin{aligned} T_E = {} & (8BT_{\text{and}}+4BT_{\text{or}}) \\ & + [46BT_{\text{and}}+(31B+12)T_{\text{or}} + (64B+96)T_{\text{shift}}](R-1) \\ & + (8BT_{\text{and}} + 7BT_{\text{or}} + 3BT_{\text{shift}}). \end{aligned} \tag{1}$$

The difference calculation between encryption and decryption is that in decryption, InvMixColumns uses different number of processing cycles from MixColumns in encryption [11]. From [11], we have the following.

(i) To implement MixColumns, operations of $19B$ byte-wise ANDs, $8B$ byte-wise ORs, and $64B$ SHIFTs are needed.
(ii) To implement InvMixColumns, operations of $134B$ byte-wise ANDs, $99B$ byte-wise ORs, and $32B$ SHIFTs are needed.

Therefore, the total number of processing cycles for decrypting a block is given as follows:

$$\begin{aligned} T_D = {} & (8BT_{\text{and}} + 4BT_{\text{or}}) + (8BT_{\text{and}} + 7BT_{\text{or}} + 3BT_{\text{shift}}) \\ & + [161BT_{\text{and}} + (122B + 12)T_{\text{or}} \\ & + (32B + 96)T_{\text{shift}}](R-1). \end{aligned} \tag{2}$$

7.2. Security MAC overhead analysis

In a long run, time is divided into cycles called superframes. A superframe includes a beacon frame, a contention access period (CAP), a contention free period (CFP), and an inactive portion.

The MAC layer needs a finite amount of time to process a frame received so that a transmitted frame will be followed by an interframe space or spacing (IFS) period, which depends on the size of the frame. If acknowledgment is used, the IFS will follow the acknowledgment (ACK) frame. Frames smaller than *aMaxSIFSFrameSize* in length will be followed by an SIFS period of a duration of at least *aMinSIFSPeriod* symbols [1]; otherwise will be followed by an LIFS of a duration of at least *aMinLIFSPeriod* symbols [1].

Let L denote the payload size in a frame in bytes, and then the numbers of processing cycles of encrypting the frame and decrypting the frame are given as follows, respectively,

$$\begin{aligned} O_E &= \left\lceil \frac{8\times L}{4B\times 8} \right\rceil T_E = \left\lceil \frac{L}{4B} \right\rceil T_E, \\ O_D &= \left\lceil \frac{8\times L}{4B\times 8} \right\rceil T_D = \left\lceil \frac{L}{4B} \right\rceil T_D. \end{aligned} \tag{3}$$

Let T_p denote the processor speed in a device. Let T_{IFS}, T_{LIFS}, and T_{SIFS} denote the time intervals for IFS, LIFS, and SIFS, respectively. Let R_T denote transmission rate. Let L_o and L_{ACK} denote the lengths of MAC overhead (header and trailer) and ACK, respectively. Let D_{A_L}, D_{A_S}, D_{U_L}, and D_{U_S} denote the delays of an acknowledged long frame transmission, an acknowledged short frame transmission, an unacknowledged long frame transmission, and an unacknowledged short frame transmission, respectively, in a successful transmission. We have

$$\begin{aligned} D_{A\_L} &= \frac{O_E}{T_p} + \frac{8L_o + 8L}{R_T} + T_{\text{IFS}} + \frac{8L_{\text{ACK}}}{R_T} + T_{\text{LIFS}}, \\ D_{A\_S} &= \frac{O_E}{T_p} + \frac{8L_o + 8L}{R_T} + T_{\text{IFS}} + \frac{8L_{\text{ACK}}}{R_T} + T_{\text{SIFS}}, \\ D_{U\_L} &= \frac{O_E}{T_p} + \frac{8L_o + 8L}{R_T} + T_{\text{LIFS}}, \\ D_{U\_S} &= \frac{O_E}{T_p} + \frac{8L_o + 8L}{R_T} + T_{\text{SIFS}}. \end{aligned} \tag{4}$$

In the above equations, we assume that T_D/T_p, the time of decrypting the last block is a part of T_{IFS}, T_{LIFS}, or T_{SIFS}. In particular, we have

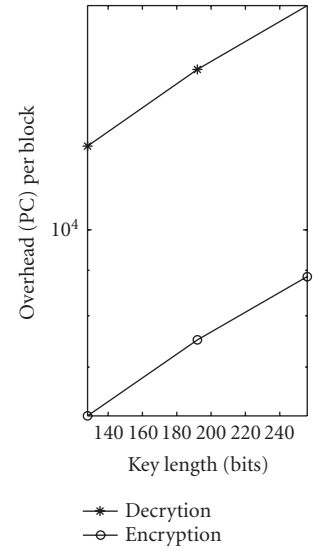
$$\frac{T_D}{T_p} < \min(T_{\text{IFS}}, T_{\text{LIFS}}, T_{\text{SIFS}}). \tag{5}$$

7.3. Results of overhead analysis

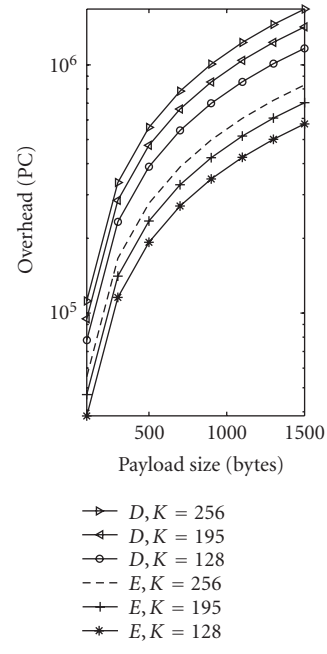
In our results, we assume that $T_{\text{and}} = T_{\text{or}} = T_{\text{shift}} = 1$ holds. AES adopts 128-bit block so that $B = 4$ hold; the AES adopts key lengths of 128, 192, or 256 bits, and therefore, we have $R = 10$, $R = 12$, and $R = 14$, respectively. We let $T_{\text{SIFS}} = T_{\text{IFS}} = 12\,\mu$s and $T_{\text{LIFS}} = 40\,\mu$s. Date rates can be 20 kb/s, 40 kb/s, and 250 kb/s. Since L_o ranges from 5 bytes to 25 bytes, we let $L_o = L_{\text{ACK}} = 25$ bytes. In the following figures, we adopt the following legends: PC = the number of processing cycles, E = encryption, D = decryption, K = key length in bits, A = acknowledged, U = unacknowledged, and MIPS = millions instructions per second.

Figure 5(a) shows overhead (PC) per block over key length. As illustrated in the figure, PC increases as the key length increases and decryption has a much larger PC than encryption does. The increase of PC over the key length appears to be linear.

Figure 5(b) shows overhead (PC) over payload size. As illustrated in the figure, PC increases as the payload size increases and decryption has a much larger PC than encryption does. The increase of PC over the payload size appears to be linear.



(a)



(b)

Figure 5: (a) Overhead (PC) per block over key length. (b) Overhead (PC) over payload.

Figure 6 shows overhead (μs) per block over MIPS. As illustrated in the figure, the overhead decreases as MIPS increases. For encryption, 128-bit key length, and 100 MIPS, the overhead is 10.28 μs, which is not trivial.

Figure 7 shows overhead (μs) over MIPS and payload size when encryption and $K = 128$ are adopted. As illustrated in the figure, the overhead increases as either MIPS decreases or the payload increases. With 100 MIPS and 1500-byte payload, the overhead is 5782.5 μs, which is very large.

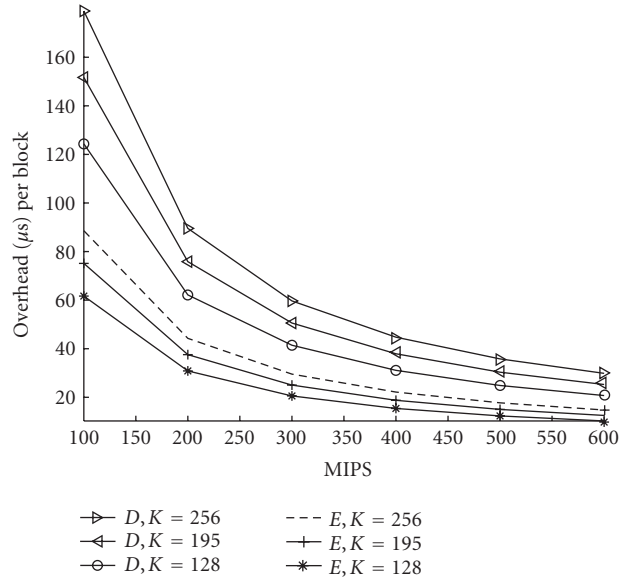


Figure 6: Overhead (μs).

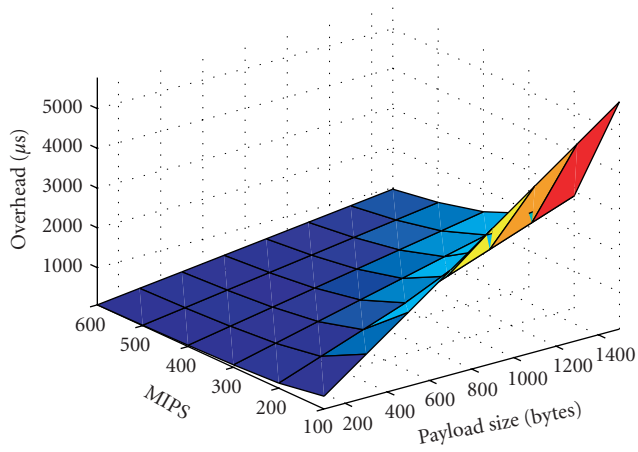


Figure 7: Overhead (μs) per block over MIPS and payload size when encryption and $K = 128$ are adopted.

Figure 8 shows delay (s) over payload size with encryption and $K = 128$. As illustrated in the figure, the overhead increases as the payload size increases. Figure 8(a) shows that with the transmission rate 20 k/s, different acknowledged schemes and long/short frame schemes have little difference in delay. Figure 8(b) shows that with acknowledged long-frame scheme, a higher transmission rate decreases delay a lot. Figure 8 shows that encryption/decryption delay does not contribute too much to the delay of transmitting a frame. In order to satisfy (5), we have $T_D/T_p < \min(T_{\text{IFS}}, T_{\text{LIFS}}, T_{\text{SIFS}}) = 12\,\mu$s under the current chosen parameters. We would like to answer the following question: how fast should the processing speed of the device be so that the above condition can be satisfied? Figure 9 shows overhead (μs) per block as well as $\min(T_{\text{IFS}}, T_{\text{LIFS}}, T_{\text{SIFS}}) = 12\,\mu$s

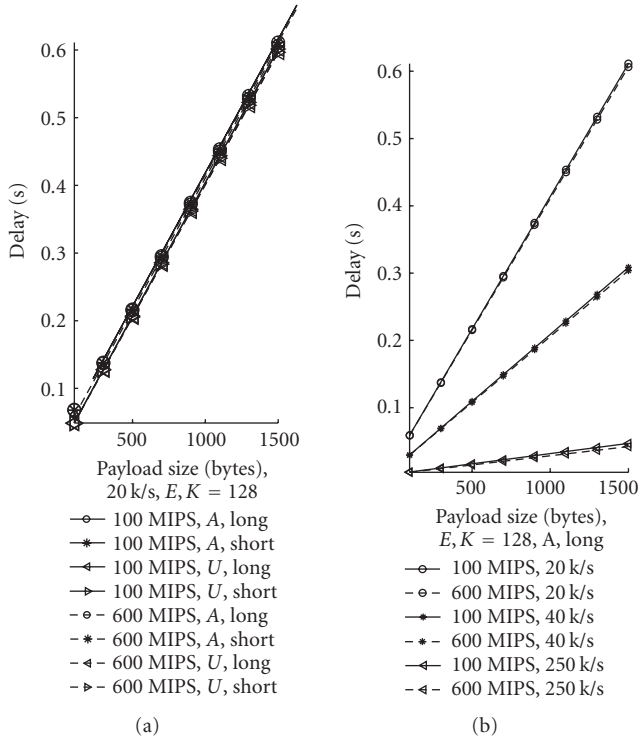


Figure 8: Delays (s) over payload size.

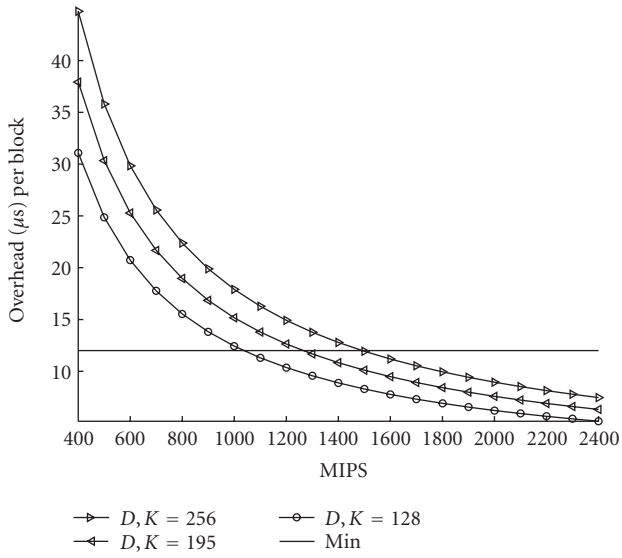


Figure 9: Overhead (μs) per block over MIPS.

over MIPS. We observe that the device should be at least more than 1000 MIPS, which is very fast for a wireless device. Furthermore, the condition of (5) is just a rough bound and the processing unit should also have another overhead. Therefore, the minimum 1000 MIPS device is a very conservative condition already.

8. CONCLUSION

In this paper, we have provided a survey of security services provided in the IEEE 802.15.4 wireless sensor networks. Security vulnerabilities and attacks have been identified. Some security enhancements have been proposed to prevent same-nonce attack, denial-of-service attack, reply-protection attack, ACK attack, and so forth. The proposed enhancements include separating the nonce from the frame counter, randomly generating nonces, using a timestamp as the frame counter, using MIC for ACK, dynamically dividing nonce spaces, eliminating the key sequence counter, tracking frame counter for each device, as well as the CCM* mode.

Furthermore, we have provided a security overhead analysis. The following observations have been made.

(i) Processing cycles per block increases as the key length increases, the payload size increases, or MIPS decreases; decryption has a much larger processing cycles than encryption does; the increase of processing cycles over the key length and the payload size appears to be linear.

(ii) For encryption, 128-bit key length, and 100 MIPS, the overhead is 10.28 μs; with 100 MIPS and 1500-byte payload, the overhead is 5782.5 μs.

(iii) Different acknowledgement scheme and long/short frame schemes have little difference in delay; encryption/decryption delay does not contribute too much to the delay of transmitting a frame.

(iv) We answered the following question: how fast should the processing speed of the device be for decryption under the current IEEE 802.15.4 parameters? We observe that the device should be at least more than 1000 MIPS, which is a very conservative condition already. Therefore, either the parameters such as T_{IFS} and T_{SIFS} of IEEE 802.15.4 should be increased or powerful devices (1000 + MIPS) should be used.

ACKNOWLEDGMENT

This research was supported in part by the Texas Advanced Research Program under Grant 003581-0006-2006.

REFERENCES

[1] IEEE 802.15.4, "Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)," May 2003.

[2] Zigbee Alliance, www.zigbee.org.

[3] I. Howitt and J. A. Gutierrez, "IEEE 802.15.4 low rate—Wireless personal area network coexistence issues," in *Proceedings of IEEE Wireless Communications and Networking (WCNC '03)*, vol. 3, pp. 1481–1486, New Orleans, La, USA, March 2003.

[4] FIPS Publication 197, "Advanced Encryption Standard," U.S. DoC/NIST, 2001.

[5] FIPS Publication 46-3, "Data Encryption Standard (DES)," U.S. DoC/NIST, October 1999.

[6] FIPS Publication 800-38C, "Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality," N U.S. DoC/NIST, May 2004.

[7] R. Struik, "Security Resolutions 802.15.4," Doc. #: IEEE 802.15-04-0540-08. 2004.

[8] N. Sastry and D. Wagner, "Security considerations for IEEE 802.15.4 networks," in *Proceedings of the ACM Workshop on Wireless Security (WiSe '04)*, pp. 32–42, Philadelphia, Pa, USA, October 2004.

[9] Y. Xiao, S. Sethi, H.-H. Chen, and B. Sun, "Security services and enhancements in the IEEE 802.15.4 wireless sensor networks," in *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM '05)*, vol. 3, St. Louis, Mo, USA, November-December 2005.

[10] R. Struik, "Formal Specification of the CCM* Mode of Operation," Doc. #: IEEE 15-04-0537-00-004b.

[11] F. Granelli and G. Boato, "A novel methodology for analysis of the computational complexity of block ciphers: Rijndael, Camellia and Shacal-2 compared," in *Proceedings of 3rd Conference one Security and Network Architectures (SAR '04)*, La Londe, France, June 2004.

Yang Xiao worked at Micro Linear as a MAC Architect involved in the IEEE 802.11 standard enhancement work before he joined the University of Memphis in 2002. He joined University of Alabama in 2006. He was a Voting Member of the IEEE 802.11 Working Group from 2001 to 2004, and is an IEEE Senior Member. He currently serves as Editor-in-Chief for International Journal of Security and Networks and for International Journal of Sensor Networks. He serves as an Associate Editor or on the editorial board for five refereed journals. He serves as a Panelist for NSF, and a Member of Canada Foundation for Innovation (CFI)'s Telecommunications expert committee. His research areas include wireless networks, mobile computing, and network security.

Hsiao-Hwa Chen is currently a Full Professor in National Sun Yat-Sen University, Taiwan. He has authored or coauthored over 160 technical papers in major international journals and conferences, and five books and three book chapters in the areas of communications. He served as symposium Cochair of major international conferences, including IEEE VTC, IEEE ICC, IEEE Globecom, IEEE WCNC, and so forth. He served or is serving as an Editorial Board Member or/and Guest Editor of IEEE Communications Magazine, IEEE JSAC, IEEE Wireless Communication Magazine, IEEE Networks Magazine, IEEE Transactions on Wireless Communications, IEEE Vehicular Technology Magazine, Wireless Communications and Mobile Computing (WCMC) Journal, International Journal of Communication Systems, and so forth. He is a Guest Professor of Zhejiang University, Shanghai Jiao Tung University, China.

Bo Sun received his Ph.D. degree in computer science from Texas A&M University, College Station, USA, in 2004. He is now an Assistant Professor in the Department of Computer Science at Lamar University, USA. His research interests include the security issues (intrusion detection in particular) of wireless ad hoc networks, wireless sensor networks, cellular mobile networks, and other communications systems.

Ruhai Wang received a Ph.D. degree in electrical engineering from New Mexico State University, USA, in 2001. He currently serves as an Assistant Professor in Department of Electrical Engineering at Lamar University, Texas. His research interests include computer networks and communication systems with emphasis on wireless communications, wireless and space Internet, network protocols and security, and performance analysis.

Sakshi Sethi is a Solution Integrator working for credit scoring organization Equifax. She was born in India and has acquired her Bachelor's degree in computer science from Manipal Institute of Technology. She received her M.S. degree in computer science from the University of Memphis and worked as a Research Assistant in Computer Science Department with Professor Yang Xiao. Her research interests include computer networks. Her work focuses on analysis of security services and security overhead in wireless networks.