Secure and Energy-Efficient Join-Leave Operations in ZigBee Network

Bong-Whan Kim[1] and Chang-Seop Park[2]
[1] Information Security Lab, Department of Computer Science, Dankook University
Cheonan, Republic of Korea
[e-mail: kbh4365@naver.com]
[2] Information Security Lab, Department of Computer Science, Dankook University
Cheonan, Republic of Korea
[e-mail: csp0@dankook.ac.kr]
*Corresponding author: Chang-Seop Park

Received June 22, 2013; revised August 29, 2013; revised September 26, 2013; accepted October 23, 2013; published November 29, 2013

Abstract

Since security plays an important role in several ZigBee applications, such as Smart Energy and medical sensor applications, ZigBee Specification includes various security mechanisms to protect ZigBee frames and infrastructure. Among them, the Join and Leave operations of ZigBee are investigated in this paper. The current Join-Leave operation is protected by the network key (a kind of group key). We claim it is not adequate to employ the network key for such purpose, and propose a new Join-Leave operation protected by the application link key (a kind of pairwise key), which is based on a more efficient key management scheme than that of ZigBee. Hence, the original Join operation consists of a total of 12 command frames, while the new Join operation consists of only 6 command frames. In particular, the security of the proposed Join-Leave operation is equivalent to or better than that of the original Join-Leave operation. The new Join-Leave operation is extensively analyzed in terms of security and efficiency, and compared with the original Join-Leave operation of ZigBee.

Keywords: ZigBee, Authentication, Key Distribution, Key Exchange, Join, Leave

The present research was conducted by the research fund of Dankook University in 2012

http://dx.doi.org/10.3837/tiis.2013.11.019

1. Introduction

ZigBee is a low cost and low power consumption wireless personal area network standard, which can be used in many different wireless sensor network applications. The latest ZigBee standard, ZigBee Specification [1] defines Network and Application Support layers upon the IEEE 802.15.4 [2]. Since security plays an important role in several ZigBee applications, such as Smart Energy and medical sensor applications, ZigBee Specification includes various security mechanisms to protect ZigBee frames and infrastructure. Key management is an important primitive for building sensor network security. There are two types of key management schemes proposed for sensor network security. One is a distributed key management scheme, where each device interacts directly with neighboring devices, to establish pairwise keys, based on the pre-loaded keying materials. Random key pre-distribution schemes [3, 4] and Transitory master key-based schemes [5, 6, 7] belong to this category. The other is a centralized key management scheme [1, 8]; in this case, a base station plays the role of key server, to establish a pairwise key for any two devices. The key management scheme for ZigBee security is the centralized one. The Trust Center (TC) in ZigBee Coordinator is a kind of authentication and key server for other ZigBee devices. There are three kinds of keys used for ZigBee security: master key, link key, and network key. In particular, the link key can also be classified into TC link key and application link key. The network key, TC link key and application link key of ZigBee correspond to the global key, individual key and pairwise key of LEAP [5], respectively. However, the cluster key of LEAP is not defined in ZigBee. The *Key Establishment* protocol of ZigBee is for establishing a TC link key between TC and a ZigBee device, while the *Key Distribution* protocol is for establishing an application link key between any two devices through TC.

In this paper, we focus on the Join and Leave operations of ZigBee. When a new ZigBee device joins into a secured ZigBee network, two security protocols, *Key Establishment* and *Authentication* protocols, should be performed, to complete the Join operation. A total of 12 command frames are exchanged among TC, the joiner device and the parent router device. On the other hand, when a ZigBee device is to be removed from the network, the Leave operation is performed. Both Join and Leave operations are protected by the network key common to all the devices already joined into the network. Even though the network key plays an important role in securing the route maintenance at the network layer, we claim it is not adequate to employ the network key, at least for securing both Join and Leave operations, since it is a kind of group key. Instead of the network key, an application link key (a kind of pairwise key) established between the joiner device and the router device can be used to secure both the Join and Leave operations. [9] proposed a new Join operation consisting of a total of 9 command frames, using the pairwise key shared between them. However, how to establish the pairwise key between them was not proposed. [10] also proposed using the individual key for the *Authentication* protocol between the joiner device and TC, not between the joiner device and the parent router device, while there was no mention on how it could be integrated with the remaining part of the Join operation.

Contributions.

A main function of the Join operation is to establish a TC link key (individual key) between TC and the joiner device, and to securely transport the network key to the joiner device. Then, the network key (global key) is used for mutual authentication between the joiner device and its parent router device. In order to complete the Join operation of ZigBee, 12 command

frames should be exchanged among the ZigBee devices, which is a source of a lot of energy consumption. First, a new Join operation consisting of only 6 command frames is proposed, while the security of the proposed Join operation is equivalent to or better than that of the original Join operation. Hence, the energy consumption of ZigBee devices during the Join operation can be greatly reduced. Second, an application link key (pairwise key) between the joiner device and its router device can also be derived, as a result of the new Join operation. In order to derive the application link key in ZigBee, another 3-way *Key Distribution* protocol should be separately executed between them, via TC. Third, the Leave operation can be protected using the application link key, instead of the network key, so that the security of the Leave operation can be more improved, in terms that the effect of compromising ZigBee devices can be localized.

Organization.

In Section 2, ZigBee security architecture is introduced, together with the Join and Leave operations of ZigBee. After pointing out the weaknesses of the Join and Leave operations of ZigBee, a new security mechanism for the Join and Leave operations is proposed in Section 3. This is extensively analyzed, and compared with that of ZigBee in terms of security and efficiency in Section 4. Finally, performance analysis is given in Section 5.

2. ZigBee Security Architecture

Fig. 1 presents a typical ZigBee network topology, consisting of three types of devices: ZigBee Coordinator (Z_{TC}), ZigBee Router and ZigBee End Devices, where ZigBee Routers and End Devices are denoted as Z_X (X = A, B, C, …). A ZigBee device **(ZigBee Router or ZigBee End Device)** can join the network through the parent router device. Z_A becomes a parent of Z_B, when Z_B joins the network through Z_A. The Trust Center (TC) in the ZigBee Coordinator is a key component in ZigBee security. It generates various cryptographic keys and distributes them to ZigBee devices, and updates them on a regular basis, or on request from ZigBee devices. Furthermore, the TC maintains a list of ZigBee devices (device table) currently joined into the ZigBee network. From now on, the TC is also denoted as Z_{TC}, since it is a part of Z_{TC}.

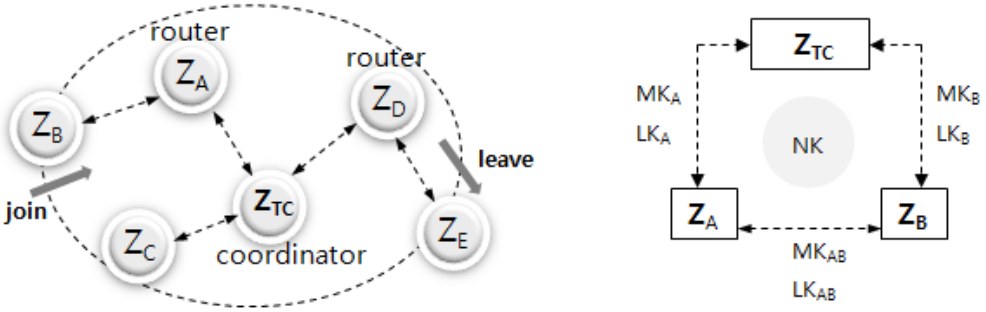


Fig. 1. ZigBee Network Topology and Types of ZigBee Keys [1]

In the following, $[m]K$ is a symmetric encryption of m using a secret key K. $kdf(.)$ and $h(.)$ are a key derivation function and a one-way hash function, respectively. $MIC(K)$ is the message integrity code computed over all preceding fields of a message, using the secret key K.

X and X^* denote Z_X's extended 64-bit MAC (Medium Access Control) address and 16-bit network address, respectively, and FC_X denotes a frame counter managed by Z_X to guarantee frame freshness. Each protocol frame has several inherent fields, described in [1, 2]. However, since most of them are not related to security, for the sake of simplicity they are excluded from the explanation. Instead, only the security-related fields are shown in each frame. The notations used in this paper are shown in **Table 1**.

Table 1. Table of Notations

Notation	Description
Z_X	ZigBee Devices (X = TC, A, B, C, …)
X, X^*	Z_X's extended 64-bit MAC address and 16-bit network address
$kdf(.)$	Key derivation function [20]
$MIC(K)$	Message Integrity Code computed over all preceding fields of a message using a secret key K [1, Appendix]
$[m]K$	Symmetric encryption of m using a secret key K [1, Appendix]
$h(.)$	one-way hash function [1, Appendix]
FC_X	Frame Counter managed by Z_X
NK	Network key
$NKSeq$	Network key Sequence number
MK_X, LK_X	TC master key and TC link key shared between Z_{TC} and a ZigBee device Z_X, respectively
MK_{XY}, LK_{XY}	Application master key and Application link key shared between any two ZigBee devices, Z_X and Z_Y, respectively
R_X, TS_X	Random number and Timestamp generated by X, respectively
$HMAC$	HMAC function [14]

2.1 Types of ZigBee Keys

Three types of keys are employed for ZigBee security, as in **Fig. 1**: master key, link key, and network key. The network key (NK) is a kind of group key used for protecting ZigBee frames at the network layer, while the link key derived from the master key is for protecting ZigBee frames at the application layer. When deriving the link key from the master key, the 4-way *Key Establishment* protocol is performed between two ZigBee devices. There are two kinds of master and link keys: TC master key (MK_A) and TC link key (LK_A) are shared between Z_{TC} and a ZigBee device Z_A, and the application master key (MK_{AB}) and application link key (LK_{AB}) are shared for end-to-end security between any two ZigBee devices, Z_A and Z_B. When requested by Z_A or Z_B, the application master or link key (MK_{AB} or LK_{AB}) is distributed to both Z_A and Z_B by Z_{TC}, through the *Key Distribution* protocol. In this case, Z_{TC} plays the role of a key server for Z_A and Z_B. The keying material can be pre-installed before deployment, or transported by Z_{TC} after deployment. The confidentiality and integrity of ZigBee frames are provided by the CCM* cryptographic algorithm, which is a minor variant of CCM [13], based on 128-bit AES.

2.2 Joining a Secured ZigBee Network

There are two options to install the keying materials into the ZigBee devices [1]. The one is pre-installation during ZigBee commissioning time, and the other is key transport after deployment. Since the key transport is performed in the air, the secure installation of the keying materials cannot be guaranteed. Therefore, throughout this paper, it is assumed that a set of ZigBee devices authorized to join the network are predefined, which means that each

authorized ZigBee device's MAC address and its TC master key are pre-installed in the TC's device table. **Fig. 2** shows a message sequence chart ensuing from when a joiner device (Z_B) communicates with a router device (Z_A), to join a secured ZigBee network. The main point of the Join operation is for Z_B to obtain the network key from Z_{TC}, and to perform a successful authentication protocol with Z_A based on the network key. It is assumed that the TC link key LK_A has already been established between Z_{TC} and Z_A. When receiving a periodic Beacon {A} command broadcasted by the router device Z_A, Z_B starts the Join operation by sending an *Association-Request* {B} command to Z_A. If an entry is available in Z_A's neighbor table, Z_A allocates Z_B's network address B^*, and makes an entry for Z_B consisting of B, B^*, and its state "*joined and unauthenticated*" in its neighbor table. Then, an *Association Response* {B^*, *Status*} command is sent to Z_B, where *Status* is "*association successful*". No security mechanism is employed at this phase, since there is no pre-established security association between them. Then, Z_A reports the joiner device's address (B, B^*) to Z_{TC} through the *Update-Device* command, which is protected by LK_A.

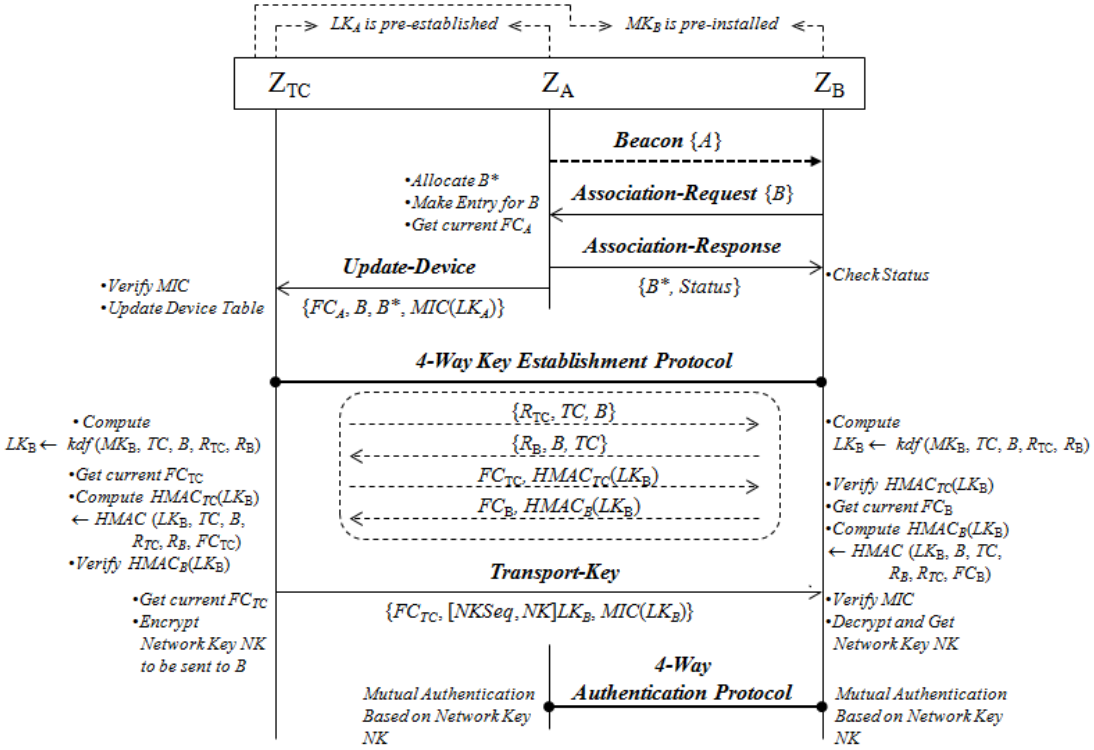
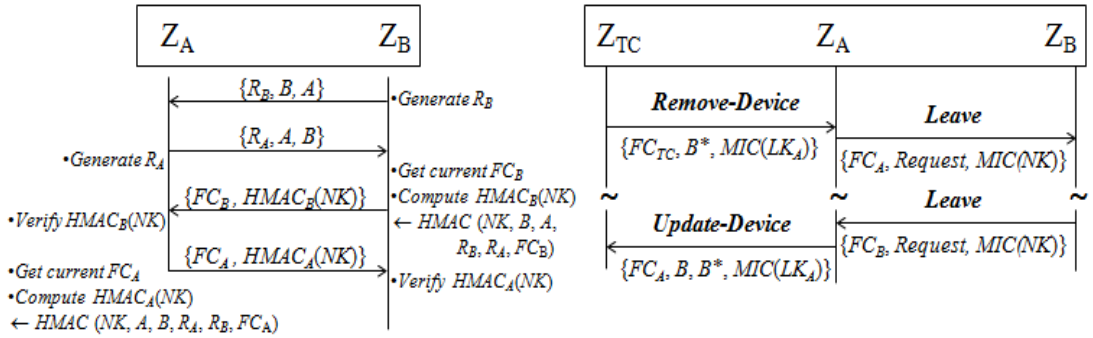


Fig. 2. Join Operation in ZigBee [1]

Z_B's MAC address B and the TC master key MK_B have already been pre-installed in the TC's device table. Since Z_B has joined and the *Key Establishment* protocol is subsequently performed, to establish the TC link key LK_B between Z_{TC} and Z_B, the device table of Z_{TC} is updated as follows: (B, MK_B, B^*, LK_B, *parent*), where *parent* is the network address of Z_A.

This is a 4-way handshake protocol, through which two random numbers R_{TC} and R_B generated by Z_{TC} and Z_B, respectively, are exchanged, and $LK_B = kdf\,(MK_B, TC, B, R_{TC}, R_B)$ is computed. The successful verifications on $HMAC_{TC}(LK_B)$ and $HMAC_B(LK_B)$ guarantee both the entity authentication and key confirmation on LK_B between them, where *HMAC* is a

HMAC function [14]. Z_{TC} then transports a current network key *NK* with its sequence number *NKSeq* to Z_B, through the *Transport-Key* command. The network key is protected by LK_B. Whenever *NK* is updated, *NKSeq* is incremented by one.



(a) 4-way Authentication Protocol (b) Leave Operations

Fig. 3. Authentication Protocol and Leave Operation in ZigBee [1]

Based on *NK*, another 4-way *Authentication* protocol between Z_A and Z_B is performed, as in **Fig. 3 (a)**. It is a kind of challenge-response mutual authentication protocol using two random numbers R_A and R_B. If it is successful, Z_B's state in Z_A's neighbor table is changed to "*joined and authenticated*", and Z_B is authorized to send and receive data or command frames. When the joiner device Z_B directly joins the network through Z_{TC}, the *Update-Device* command is not needed and the 4-way *Authentication Protocol* is performed directly between Z_{TC} and Z_B.

2.3 Leaving a Secured ZigBee Network

Z_{TC} can remove a ZigBee device from the network, as in **Fig. 3 (b)**. For example, if Z_B fails to authenticate properly during the 4-way *Authentication* Protocol, Z_{TC} requests the router device (Z_A) to remove its child device (Z_B). When receiving a *Remove-Device* command from Z_{TC}, the router device (Z_A) sends a *Leave* command, to notify the ZigBee device (Z_B) of its removal from the network. On the other hand, a ZigBee device (Z_B) can remove itself from the network, by sending a *Leave* command to its parent router (Z_A). The ZigBee device (Z_B) can select the parent router (Z_A) since Z_B stores its parent router information in the neighbor table during the association phase. Z_{TC} will also be informed of the device that leaves the network, through the *Update-Device* command. In either case, Z_{TC} shall delete the device from its device table. If Z_{TC} and the router share a TC link key LK_A, then the *Remove-Device* command between the two will be secured with the TC link key LK_A. The *Leave* command is protected by the network key known to all ZigBee devices.

3. An Enhanced Security Mechanism for ZigBee

A purpose of the *Key Establishment* protocol in **Fig. 2** is to establish the TC link key LK_B between Z_{TC} and Z_B based on the pre-shared TC master key MK_B. The TC link key is used to protect the network key to be delivered to Z_B. Based on the network key, mutual authentication is performed between Z_A and Z_B through the *Authentication* protocol, which is a final step of the Join operation. In order for a joiner device to join a secured ZigBee network, a total of 12 command frames should be exchanged among Z_{TC}, Z_A, and Z_B. In this Section, a new Join operation consisting of only 6 command frames is proposed, which also allows an application

link key LK_{AB} to be established between Z_A and Z_B. So, the application link key can be employed to secure the *Authentication* protocol and the Leave operation, instead of the network key.

3.1 Assumptions and Design Principles

First, as in **Fig. 2**, it is assumed there is a pre-established TC link key LK_A between Z_{TC} and Z_A, and Z_B's TC master key MK_B is pre-installed in Z_{TC}'s device table during ZigBee commissioning time. Second, the 4-byte frame counter FC_X is used to guarantee the freshness of the command frames in ZigBee. Due to the security problems of the frame counter of ZigBee pointed out in [11], the 8-byte timestamp TS_X (X = *TC*, *A*, *B*) is instead employed in the proposed Join and Leave operations. Nonetheless, strict time synchronization between two ZigBee devices is not required for security, since the timestamp is more like a sequence number in the proposed Join and Leave operations. Third, the timestamps are also used to derive the TC link key LK_B and application link keys LK_{AB}, unlike in **Fig. 2**, where random numbers are used to derive them. Fourth, an application link key LK_{AB} can also be established between Z_A and Z_B, as well as the TC link key LK_B, without the *Key Establishment* protocol. Namely, the *Key Establishment* protocol and the *Transport-Key* command are omitted in the proposed Join operation, while a newly-defined *Update-Result* command is employed. The application link key is used to protect the network key, and to perform both an *Authentication* protocol and the Leave operations. In particular, Z_A sends the network key to Z_B, when the *Authentication* protocol is performed successfully.

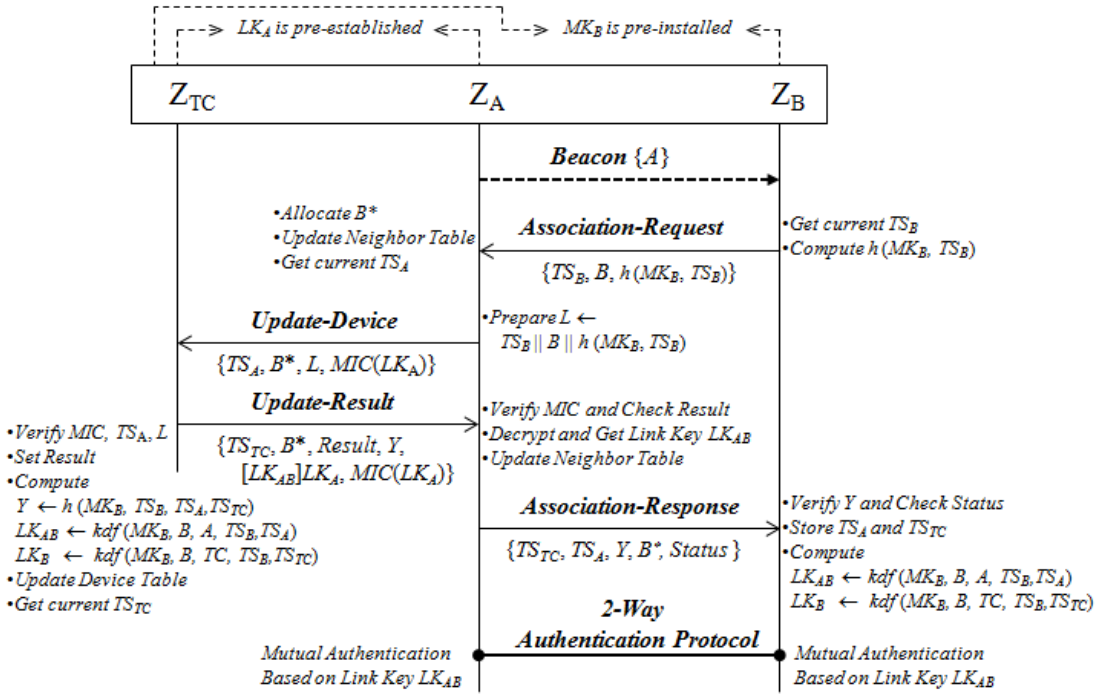


Fig. 4. A New Join Operation in ZigBee

3.2 Proposed Join Operation

Fig. 4 shows a message sequence chart for a new Join operation proposed for ZigBee. A main difference between **Fig. 2** and **Fig. 4** is that both the *Key Establishment* protocol and the

Transport-Key command are replaced by a new *Update-Result* command.

(Step 1) A joiner device Z_B computes $h(MK_B, TS_B)$ based on its current timestamp TS_B, and sends an *Association Request* $\{TS_B, B, h(MK_B, TS_B)\}$ command to Z_A, where $h(MK_B, TS_B)$ is used to authenticate Z_B by Z_{TC}. If entry is available in Z_A's neighbor table, Z_A allocates Z_B's network address B^*, and makes an entry for Z_B in its neighbor table, as follows:

Update Z_B's entry in Neighbor Table

- (***B***, *network_addr, stored_TS$_B$, Application_link_key*, *state*)
- *network_addr* ← B^*;
- *stored_TS*$_B$ ← *TS*$_B$;
- *state* ← "*joined and unauthenticated*";

(Step 2) A router device Z_A sends to Z_{TC} an *Update-Device* $\{TS_A, B^*, L, MIC(LK_A)\}$ command, where $L = TS_B \| B \| h(MK_B, TS_B)$. The freshness of the command is guaranteed by both the timestamp TS_A generated by Z_A, and $MIC(LK_A)$.

When receiving the *Update-Device* command, Z_{TC} first verifies if it is authentic command sent from Z_A. If the verification of $MIC(LK_A)$ fails or TS_A is not fresh, then Z_{TC} discards the command frame, and stops processing. Second, by checking L, Z_{TC} verifies if B in L is authorized to join. If Z_B's entry is not found in the device table, or the received $h(MK_B, TS_B)$ in L is not valid, Z_{TC} notifies Z_A that the *Update-Device* command cannot be processed, by replying with *Update-Result* $\{ TS_{TC}, B^*,$ "*Update Unsuccessful*", _, _, $MIC(LK_A)$ $\}$ command, where "_" denotes an empty field, namely Y and LK_{AB} are not computed. When all of the above tests are passed, the device table is updated, as follows:

Update Z_B's entry in Device Table

(***B***, ***MK*$_B$**, *network_addr, stored_TS$_B$, TC_link_key*, *parent*)

- *network_addr* ← B^*;
- *stored_TS*$_B$ ← *TS*$_B$;
- *TC_link_key* ← *LK*$_B$;
- *parent* ← A^*;

(Step 3) Z_{TC} computes Y, LK_{AB}, and LK_B as in **Fig. 4**, where $Y = h\ (MK_B, TS_B, TS_A, TS_{TC})$, $LK_{AB} = kdf\ (MK_B, B, A, TS_B, TS_A)$, and $LK_B = kdf\ (MK_B, B, TC, TS_B, TS_{TC})$. Then, an *Update-Result* $\{TS_{TC}, B^*, Result, Y, [LK_{AB}]LK_A, MIC(LK_A)\}$command is sent to Z_A, where *Result* = "*Update Successful*".

When receiving the *Update-Result* command with valid $MIC(LK_A)$, Z_A first checks if *Result* is "*Update Successful*". If not, Z_A deletes Z_B's entry in the neighbor table, and stops processing. Otherwise, the encrypted LK_{AB} is decrypted from the command, and its neighbor table is updated, as follows:

(***B***, ***B****, ***TS$_B$***, *Application_link_key* = ***LK*$_{AB}$**, "***joined and unauthenticated***")

(Step 4) Z_A sends to Z_B an *Association-Response* $\{TS_{TC}, TS_A, Y, B^*, Status\}$ command, where *Status* = "*Successful Association*". When receiving it, Z_B first verifies if Y is valid, since TS_{TC} and TS_A are used to compute the link keys. If the verification is successful, Z_B computes and shares the application link key LK_{AB} and the TC link key LK_B with Z_A and Z_{TC}, respectively. Both TS_A and TS_{TC} are also stored. Now, Z_B performs a 2-way *Authentication* protocol. If it is done successfully, Z_B's state in Z_A's neighbor table is changed from "*joined and unauthenticated*" to "***joined and authenticated***", and Z_B is authorized to send and receive data or command frames.

When the joiner device Z_B directly joins the network through Z_{TC}, the *Update-Device* and *Update-Resul*t commands are not needed and the 2-way *Authentication Protocol* is performed directly between Z_{TC} and Z_B.

3.3 A 2-Way Authentication Protocol

The *Authentication* protocol in **Fig. 2** is a 4-way handshake protocol for mutual authentication between Z_A and Z_B based on the network key *NK*. However, the proposed *Authentication* protocol shown in **Fig. 4** is a 2-way handshake protocol based on the application link key LK_{AB} already shared between Z_A and Z_B.

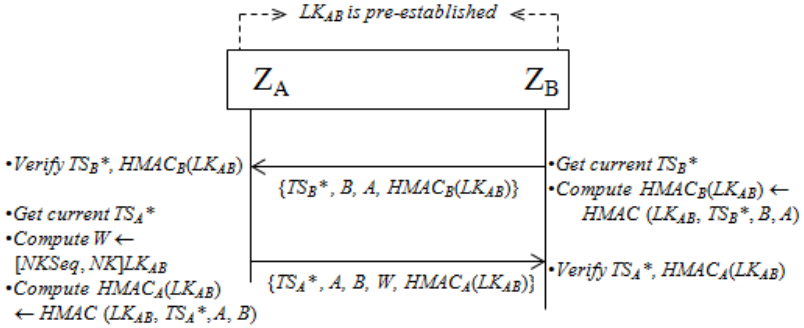


Fig. 5. A Proposed 2-Way Authentication Protocol

Z_B computes $HMAC_B(LK_{AB}) = HMAC\ (LK_{AB}, TS_B^*, B, A)$ based on the current timestamp TS_B^*, and sends to Z_A a command frame $\{TS_B^*, B, A, HMAC_B(LK_{AB})\}$. When receiving it, Z_A checks if the stored TS_B is less than the received TS_B^*, and if the computed *HMAC* is the same as the received *HMAC*. If both verifications are successful, Z_A computes $W = [NKSeq, NK]LK_{AB}$ and $HMAC_A(LK_{AB}) = HMAC\ (LK_{AB}, TS_A^*, A, B)$, and sends to Z_B a command frame $\{TS_B^*, A, B, W, HMAC_A(LK_{AB})\}$. Then Z_B performs the verification on the timestamp and $HMAC_A(LK_{AB})$, and decrypts W and gets the network key *NK*.

3.4 Proposed Leave Operation

During the Leave operation of ZigBee in **Fig.3 (b)**, the *Leave* command is protected by the network key known to all ZigBee devices. However, if the network key is compromised, an adversary can send a bogus *Leave* command to any ZigBee device in the network for the purpose of removing a victim device from the network. On the other hand, as a result of successful completion of the proposed Join operation in **Fig. 4**, the application link key LK_{AB} is shared between Z_A and Z_B, and it can be employed to secure the *Leave* command, instead of the network key. Therefore, the newly proposed Leave operation is the same as that in **Fig.3 (b)** except the *Leave* command as follows:

$$Z_{TC} \rightarrow Z_A : Remove\text{-}Device\ \{FC_{TC}, B^*, MIC(LK_A)\}$$

$Z_A \rightarrow Z_B$: *Leave* $\{FC_A, Request, MIC(LK_{AB})\}$

$Z_A \leftarrow Z_B$: *Leave* $\{FC_B, Request, MIC(LK_{AB})\}$
$Z_{TC} \leftarrow Z_A$: *Remove-Device* $\{FC_A, B, B^*, MIC(LK_A)\}$

The network key is a kind of group key, while the application link key is a kind of pair-wise key. The security of the proposed Leave operation will be more detailed in Section 4.3.

4. Security Analysis and Comparisons

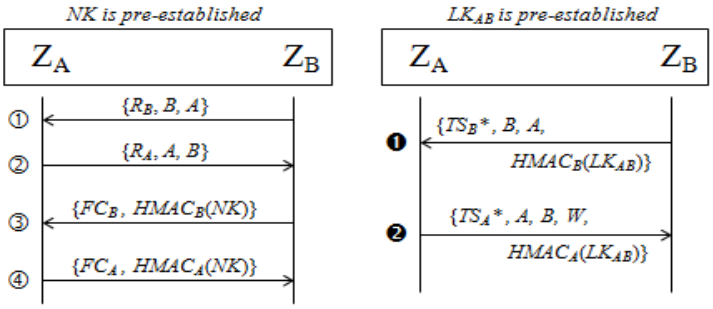
In this Section, the ZigBee Join-Leave and the proposed Join-Leave operations are compared and analyzed, in terms of security and efficiency.

4.1 Security Assumptions and Threat Model

The level of security provided by the ZigBee security architecture depends on the safekeeping of the symmetric keys, and on the protection mechanisms employed. So, trust in the ZigBee Join and Leave operations ultimately reduces to trust in the secure installation, processing and storage of keying material. The ZigBee specification assumes that the keying material, such as TC master key, is securely installed at each ZigBee device during the ZigBee commissioning time before deployment. However, due to the low-cost nature of *ad hoc* network devices, one cannot generally assume the availability of tamper-resistant hardware, which means that physical access to a device may yield access to secret keying material and other privileged information [1]. Therefore, it is assumed that an adversary is able to eavesdrop and manipulate the ZigBee commands exchanged, and access the keying materials in ZigBee devices (Z_A and Z_B in **Fig. 2** and **Fig. 4**) by physical capture. One exception is the ZigBee Coordinator (Z_{TC}), with the device table maintaining TC master keys of authorized ZigBee devices. So, it is assumed that the ZigBee Coordinator is physically protected, so that physical capture and compromise by an adversary are not feasible. Finally, the internal functioning of the devices in the network cannot be arbitrary controlled by an adversary.

4.2 Nonces and Replay Attacks

The freshness of the command frames in the Join and Leave operation of ZigBee (**Fig. 2**) is guaranteed by the 4-byte frame counter. For this purpose, each device maintains two kinds of frame counter: outgoing frame counter (*OutgoingFrameCounter*), and a set of incoming frame counters (*IncomingFrameCounter*). When a frame is sent, the frame counter field of the frame is set to *OutgoingFrameCounter,* and it is increased by one. When a frame is received, *IncomingFrameCounter* corresponding to the sender address is compared with the frame counter value in the received frame. On the other hand, in the proposed Join and Leave operation (**Fig. 4**), 8-byte timestamps are employed, instead of the frame counters. Nonetheless, strict time synchronization between two devices is not required for security, since the timestamp is more like a sequence number in the proposed Join and Leave operations. Therefore, replay attacks against both schemes (**Fig. 2** and **Fig. 4**) are not feasible without the secret keying materials. Besides the security problems of the frame counter in ZigBee pointed out in [11], there is another advantageous reason to employ timestamps for the *Authentication* protocol. The *Authentication* protocol of ZigBee is a 4-way handshake protocol for mutual authentication, based on the network key. Two random numbers R_A and R_B are employed to compute and verify the HMAC value.



(a) Authentication protocol of ZigBee [1] (b) Proposed Authentication protocol

Fig. 6. Comparison of 4-Way and 2-Way Authentication Protocols

In **Fig. 6 (a)**, Z_A and Z_B do not have each other's *IncomingFrameCounter* in advance, since they exchange frames for the first time. In **Fig. 2**, since security is not enabled on both *Association Request* and *Response* commands, the frame counters are not included. Even though the command frames ③ and ④ include the frame counter, it is not for frame freshness, but for creating each other's frame counter for the first time. Therefore, frame freshness is guaranteed by two random numbers. That is why the *Authentication* protocol in the Join operation of ZigBee consists of 4 command frames. On the other hand, in **Fig. 6 (b)**, instead of random numbers, the timestamps, TS_A* and TS_B*, are used to provide frame freshness, since Z_A and Z_B store the previous timestamps, TS_A and TS_B, received from each other through the *Association Request* and *Response* commands.

4.3 Forgery Attacks for Unauthorized Leave

Since the network key is known to all the devices joined into the ZigBee network, we claim it is not adequate to use it, at least for securing the Leave operation. In this section, three types of attack scenarios for the Leave operation of ZigBee are investigated.

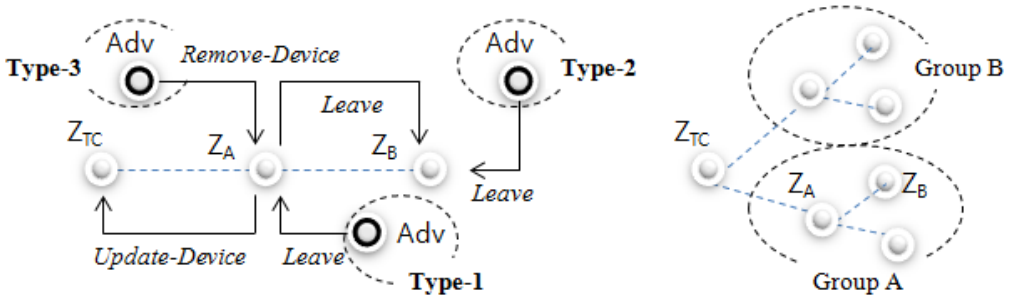


Fig. 7. Attack Scenarios for the Leave Operation of ZigBee

First, suppose an adversary (Adv) disguising Z_B knows the network key, and sends a bogus *Leave* command to Z_A (Type-1 Attack). If this occurs, the entry for Z_B is deleted from the neighbor table of Z_A, which means it is detached from the ZigBee network, so that frames destined for Z_B are discarded by Z_A. Second, if Adv disguising Z_A sends a bogus *Leave* frame to Z_B (Type-2 Attack), Z_B think Z_A is no longer its router, and tries to find another neighboring router. Third, suppose Adv knows the TC link key LK_A of the router device Z_A (Type-3 Attack), and forges and sends the *Remove-Device* $\{FC_{TC}, B^*, MIC(LK_A)\}$ command to Z_A. Then, Z_B is also removed from the network, when receiving the *Leave* command from Z_A. For

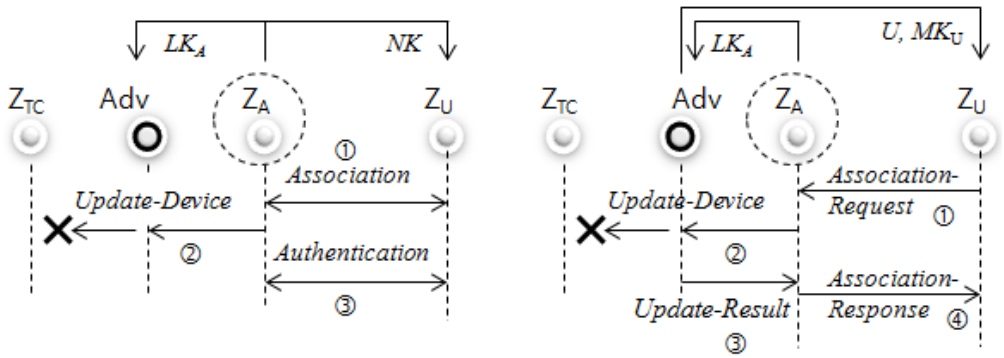
Type-1 and Type-2 attacks, if a device from Group A in **Fig. 7** is physically captured and the network key is extracted from it, Adv can remove any device in Group A, as well as any device in Group B. Namely, many random ZigBee devices can be out of the network.

On the other hand, as a result of successful completion of the proposed Join operation, the application link key LK_{AB} is shared between Z_A and Z_B, and it can be employed to secure the *Leave* command, instead of the network key. However, if the application link key LK_{AB} is also exposed to Adv, we encounter the same security problem for the Leave operation. In such a case, only Z_B is out of the network. Namely, the above three types of attacks can be localized to the group (Group A in **Fig. 7**) to which the victim device belongs, while the remaining devices in the other group (Group B in **Fig. 7**) can remain unaffected.

4.4 Man-In-The-Middle Attacks for Unauthorized Join

Suppose an unauthorized device Z_U without the current network key tries to join the network through Z_A. In the case of the Join operation in **Fig. 2**, after exchanging the *Association* commands between Z_A and Z_U, Z_A sends the *Update-Device* command to Z_{TC}. However, since the information about Z_U is not pre-installed in the device table of Z_{TC}, the *Remove-Device* command is sent back to Z_A, so that the temporary entry for Z_U is deleted from the neighbor table of Z_A, and the subsequent authentication procedure cannot be initiated. In the case of the proposed Join operation in **Fig. 4**, the *Update-Result* command plays the same role of the *Remove-Device* command, when the information about Z_U is not pre-installed in the device table of Z_{TC}. Hence, if Zu is proven to be unauthorized, the temporary entry for Z_U is also deleted from the neighbor table of Z_A, so that the unauthorized join attempt is blocked.

On the other hand, suppose an adversary (Adv) can access the keying materials (*NK* or LK_A) of the router device Z_A, and aid an unauthorized device Z_U to join the network through Z_A, as in **Fig. 8**. Initially, Adv installs *NK* or (U, MK_U) into Z_U. In the case of **Fig. 8 (a)**, after the *Association* commands are exchanged with Z_A and Z_U, Adv blocks the *Update-Device* command sent from Z_A. Then, Z_U initiates the *Authenticaion* protocol with Z_A, based on *NK*. Eventually, Z_U is successfully attached to the network, and can send and receive the broadcasted frames protected by *NK*, even though its information is not in the device table of Z_{TC}.



(a) Join Operation of ZigBee (b) Proposed Join Operation

Fig. 8. Attack Scenarios for Join Operation

In the case of **Fig. 8 (b)**, Z_U should first compute $h(MK_U, TS_U)$. Even if the *Association-Request* $\{TS_U, U, h(MK_U, TS_U)\}$ command is sent to Z_A and the subsequent *Update-Device* $\{TS_A, U^*, L, MIC(LK_A)\}$ command is sent to Z_{TC}, the verification on $h(MK_U, TS_U)$ by Z_{TC} will fail, since (U, MK_U) is not pre-installed in the device table of Z_{TC}. However,

suppose Adv blocks the *Update-Device* command, and responds with the following forged *Update-Result* command to Z_A, where $TS_{TC}^{\#}$, $LK_{AU}^{\#}$, and $Y^{\#}$ are fabricated by Adv.

$$Update\text{-}Result\ \{TS_{TC}^{\#}, U^{*}, Result, Y^{\#}, [LK_{AU}^{\#}]LK_A, MIC(LK_A)\}, \text{ where}$$
$$Y^{\#} = h\,(MK_U, TS_U, TS_A, TS_{TC}^{\#})$$

Then, both commands can be normally processed, and finally the application link key $LK_{AU}^{\#}$ is shared between Z_A and Z_U. Namely, Z_U is also successfully attached to the network. However, in both cases ((a) and (b) in **Fig. 8**), Z_U joins the network *incompletely,* since its information is not pre-installed into the device table of Z_{TC}. So, Z_U cannot perform a normal communication with another device in the network. In order for Z_U to communicate with it, Z_U should first contact with Z_{TC}, to request the application link key to be shared with it.

4.5 DoS Attacks against Join Operation

During the Join operation, a DoS attack inducing unnecessary energy consumption can be mounted against Z_{TC} and the router device Z_A. There are two cases: the first is for Adv to send the *Association Request* command to Z_A with a bogus MAC address that is not pre-installed into the device table of Z_{TC}. In this case, after making an entry in its neighbor table, Z_A responds with the corresponding *Association Response* command to Adv, and then sends the *Update-Device* command to Z_{TC}. Since the bogus MAC address is not pre-installed in the device table, Z_{TC} sends a *Remove-Device* command to Z_A, to delete the entry in the neighbor table. In this case (Case 1 in **Fig. 9**), the bogus *Association Request* command induces three unnecessary command frames. The second is for Adv disguising Z_B to send the *Association Request* command to Z_A with a Z_B's MAC address that is pre-installed into the device table of Z_{TC}. In this case (Case 2 in **Fig. 9**), the Z_{TC} performs the unnecessary *Key Establishment* protocol with it, which eventually fails, after exchanging two command frames. So, the bogus *Association Request* command induces five unnecessary command frames. On the other hand, in the proposed Join operation, the bogus *Association Request* command induces only two unnecessary command frames.

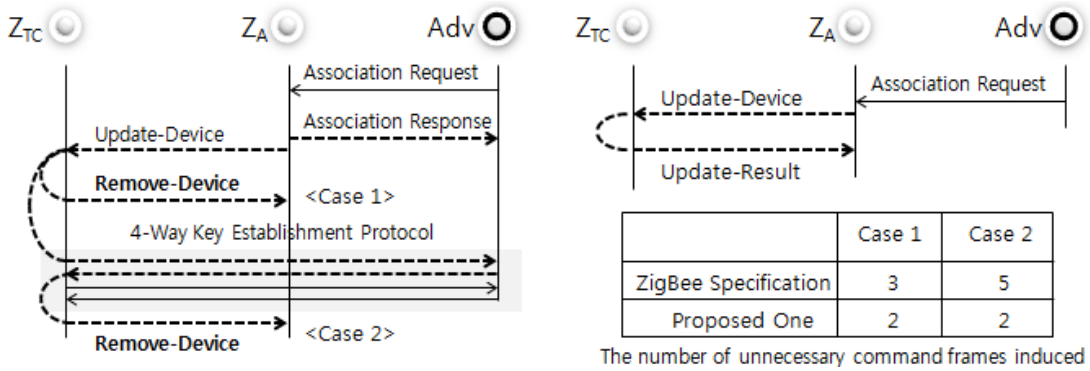


Fig. 9. Comparison of Induced Unnecessary Command Frames

4.6 Security Comparisons of ZigBee and The Proposed One

The security of Join-Leave operation of ZigBee and the proposed one has been investigated in Section 2, 3, 4 and 5, and is summarized in **Table 2**. As indicated in **Table 2**, the security level of the proposed one is equivalent to or better than that of ZigBee, while the performance efficiency of the proposed one is much better than that of ZigBee, which will be shown in

Section 5.

Table 2. Security Comparisons of ZigBee and The Proposed One

	ZigBee		Proposed One	
Replay Attack against Join and Leave	Secure with Frame Counter. But, 4-way handshake for Authentication Protocol		Secure with Timestamp. But, 2-way handshake for Authentication Protocol	
Forgery Attack against Leave	Insecure when network key is exposed. The attack can be mounted on all ZigBee devices.		Insecure when application link key is exposed. However, the attack is localized. (See **Fig. 7**)	
Man-In-The-Middle Attack against Join	Secure with network key. Insecure when network key is exposed. (Incomplete Join)		Secure with application Link key. Insecure when application link key is exposed. (Incomplete Join)	
DoS Attack against Join	unnecessary frames induced		unnecessary frames induced	
	<Case 1> 3	<Case 2> 5	<Case 1> 2	<Case 2> 2

Additionally, in this Section, we compare the key management for the secure Join-Leave operation of ZigBee and the proposed one. A purpose of the *Key Establishment* protocol in **Fig. 2** is to establish the TC link key LK_B between Z_{TC} and Z_B, based on the pre-shared TC master key MK_B. The TC link key is used to protect the network key to be delivered to Z_B. Based on the network key, mutual authentication is performed between Z_A and Z_B through the *Authentication* protocol, which is a final step of the Join operation.

In the proposed Join operation of **Fig. 4**, both the 4-way *Key Establishment* protocol and *Transport-Key* command frame are omitted. Instead, a new *Update-Result* command frame is defined, and additional fields associated with the key establishment are embedded into the 4 command frames, *Association-Request*, *Update-Device*, *Update-Result*, and *Association -Response*. In particular, the role of the *Update-Result* command frame is two-fold: the first is to notify Z_A of the result of processing the *Update-Device* command frame, and the second is to convey the keying material, namely $< [LK_{AB}]LK_A >$ to Z_A and $< TS_A, TS_{TC}, Y >$ to Z_B, where the integrity of TS_A and TS_{TC} is guaranteed by $Y = h\ (MK_B, TS_B, TS_A, TS_{TC})$. Based on the keying material, the TC link key $LK_B = kdf\ (MK_B, B, TC, TS_B, TS_{TC})$ can be shared between Z_{TC} and Z_B, and the application link key $LK_{AB} = kdf\ (MK_B, B, A, TS_B, TS_A)$ can also be shared between Z_A and Z_B. Eventually, the subsequent *Authentication* protocol and the Leave operation can be secured, using the application link key, instead of the network key.

It is possible to use the application link key LK_{AB} for the *Authentication* protocol of **Fig. 2**, and the Leave operation of **Fig. 3**. However, in order to do so, an additional 3-way *Key Distribution* protocol defined in the ZigBee specification [1] should be performed among Z_A, Z_B, and Z_{TC} beforehand, which induces a long delay, and consumes more energy in the participating ZigBee devices.

4.7 Rekeying Issue in ZigBee

Even though the network key is not employed for the new Join and Leave operations, the network key plays an important role in securing the broadcast frames for route maintenance at the network layer. In addition to use the network key for the Join-Leave operation, the ZigBee specification [1] defines how to update the network key. The TC broadcasts a new network key NK_{new} encrypted with the old network key NK_{old}, through the *Transport-Key* frame.

- *Transport-Key* $\{FC_{TC}, [NK_{new}]NK_{old}, MIC(NK_{old})\}$
- *Switch-Key* $\{FC_{TC}, MIC(NK_{old})\}$

After receiving a *Switch-Key* frame subsequently broadcasted from the TC, all ZigBee devices begin using the new network key. The ZigBee specification uses the word "periodically" when the network key update issue is referred, but gives no further guidelines. It should also be updated in case of both Join and Leave events [19] and device compromise event [10].There are two security weaknesses in the network key update of ZigBee. First, perfect forward security is not guaranteed, since the compromise of the old network key leads to that of the new network key. Second, if the network key is exposed, an adversary disguising TC can broadcast the bogus network keys, so that the network keys are not synchronized among the TC and ZigBee devices. The network key update should be triggered by TC only. Regarding these security weaknesses, the ZigBee specification also mentions that the new network key can be individually encrypted by the TC link key shared with each device, and sent (unicast) to each ZigBee device. However, if the ZigBee network consists of too many devices, a scalability problem occurs. In this case, key management schemes in multicast dynamic groups, such as LKH (Logical Key Hierarchy) rekeying [15] and OFT (One-way Function Tree) rekeying [16], might be solutions to solve the scalability problem.

5. Performance Analysis

Fig. 10 shows a frame format for the *Leave*, *Update-Device* and *Authentication* commands, while **Fig. 11** shows that of *Association-Request* and *Response* commands. The length of each field in the command is in bytes, and depends on the type of the command.

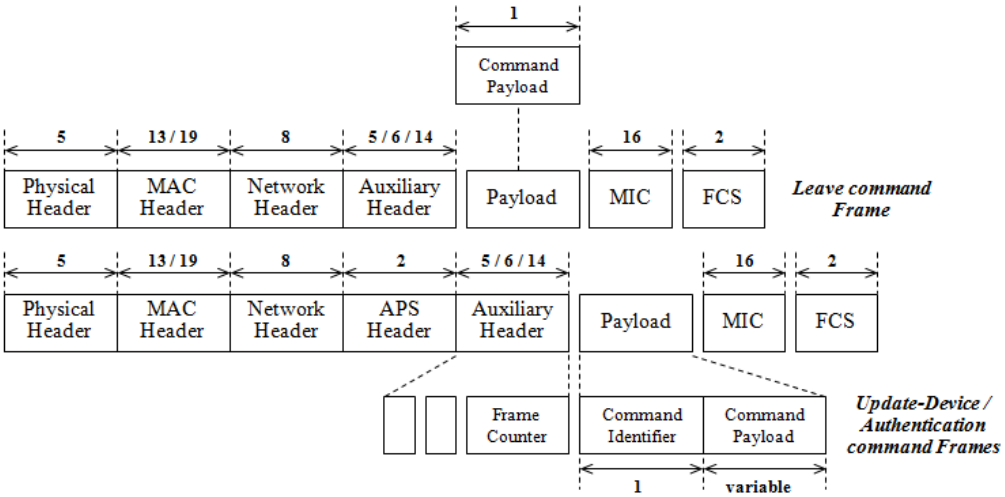


Fig. 10. Frame Format for *Leave* / *Update-Device* / *Authentication* Commands [1]

The newly defined *Update-Result* command can have the same frame format as that of **Fig. 10**. The security-related parameters of the 2-way *Authentication* commands, *Update-Device* and *Update-Result* commands, can be put into the “Command Payload” field in **Fig. 10**, whose length is variable. On the other hand, for the new security-related parameters of the *Association-Request* and *Response* commands introduced in Section 3, two new fields can be added into the dotted boxes in **Fig. 11**.

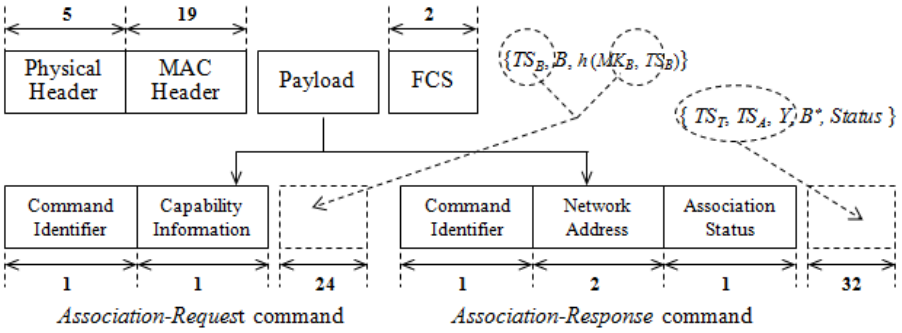


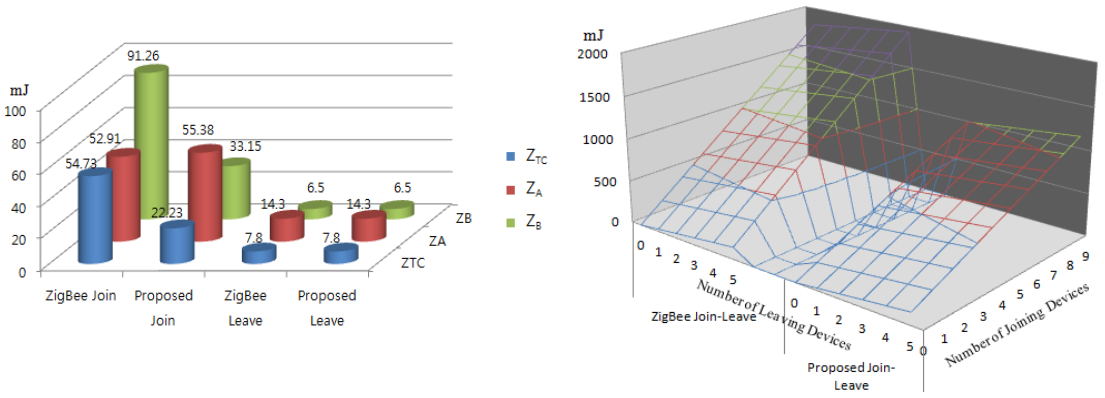
Fig. 11. Frame Format for *Association-Request* / *Response* Commands [2]

Table 3 shows the number of bytes of each command frame used in ZigBee and the proposed one, where both the *Authentication* and *Key Establishment* protocols of ZigBee are 4-way handshake protocols consisting of 4 command frames each. On the other hand, the 2-way *Authentication* protocol of the proposed one consists of only 2 command frames.

Table 3. Frame Length in Bytes for Each Command in ZigBee and the Proposed One

	Update-Device	Update-Result	Association-Request	Association-Response	Authentication Protocol	Remove-Device	Leave	Transport Key	Key Establishment
ZigBee	63	N/A	28	30	80/80/63/63	60	50	86	68/68/68/68
Proposed	80	91	52	62	62/79	60	50	N/A	N/A

In order to evaluate energy consumption, only the number of bytes sent and received by the devices is taken into consideration. We did not include the relatively insignificant levels of energy consumed, since 1 bit transmitted in a sensor network consumes as much power as 800 -1000 instructions [17]. Assuming the energy consumption during transmission or reception of a 1-byte message equals 0.13 mJ [18], **Fig. 12 (a)** shows the energy consumption (mJ) of each device, when sending and receiving the frames in **Table 3** during the Join and Leave operations, assuming no security attack is mounted.



(a) Energy Consumption of Each Device for a Single Join /Leave Operation

(b) Cumulative Energy Consumption of 3 Devices as the Number of Joining/Leaving Devices increases

Fig. 12. Energy Consumption of ZigBee Devices during the Join/Leave Operation

As shown in **Fig. 12 (a)**, the energy consumptions in Z_{TC} and Z_B of the proposed Join operation are much less than those of the ZigBee Join operation. **Fig. 12 (b)** shows the cumulative energy consumption in 3 devices (Z_{TC}, Z_A, Z_B) as the number of joining / leaving devices increases. The main difference of energy consumption between the ZigBee and the proposed one is due to whether the 4-way *Key Establishment* protocol is included, or not, in the Join operation. The reason that the energy consumption in Z_A of the proposed one is a little higher than that of ZigBee in **Fig. 12 (a)** is due to the additional fields embedded into the command frames associated with the key establishment and association. In particular, there is no difference between the ZigBee Leave operation and the proposed one in terms of the frame length of the commands in them, which means that the energy consumptions for both Leave operations remains the same. In conclusion, the proposed Join and Leave operations are more efficient than the original ZigBee ones, as shown in **Fig. 12 (b)**.

6. Concluding Remarks

Since security plays an important role in several ZigBee applications, various security mechanisms are employed to protect ZigBee frames and infrastructure. In this paper, Join and Leave operations of ZigBee have been investigated. A couple of weaknesses of ZigBee have been pointed out in terms of security and efficiency, and a new security mechanism has been proposed to address them. The proposed Join operation is more energy-efficient than that of ZigBee since the number of command frames involved in the Join operation has been reduced in half. In particular, the application link key can be derived as a result of the proposed Join operation, and it can be used to secure both *Authentication* protocol and Leave operation, instead of the network key. It has been shown that it is more secure and efficient than that of the original Join and Leave operations of ZigBee.

References

[1] ZigBee-2007, ZigBee-2007 Specification. ZigBee Alliance, USA, 2008.
Article (CrossRef Link)

[2] IEEE 802.15.4-2006 Wireless Medium Access Control and Physical Layer Specifications for Low-Rate Wireless Personal Area Networks. IEEE, USA, 2006.
Article (CrossRef Link)

[3] L. Eschenauer and V. Gligor, "A Key Management Scheme for Distributed Sensor Networks," in *Proc. of Computer and Communications Security*, pp. 22-31, Nov. 2002.
Article (CrossRef Link)

[4] H. Chan, A. Perrig, and D. Song, "Random Key Pre-distribution Schemes for Sensor Networks". in *Proc. of IEEE Symposium on Security and Privacy*, pp. 112-120, May 2003.
Article (CrossRef Link)

[5] S Zhu, S Setia, and S Jajodia, "LEAP+: Efficient Security Mechanisms for Large-scale Distributed Sensor Networks," ACM Transactions on Sensor Networks, vol. 2, no. 4, pp. 500–528, 2006.
Article (CrossRef Link)

[6] J. Deng, C. Hartung, R. Han, and S. Mishra, "A Practical Study of Transitory Master Key Establishment for Wireless Sensor Networks," in *Proc. of First International Conference on Security and Privacy for Emerging Areas in Communications Networks*, pp. 289-302, 5-9 Sept. 2005.
Article (CrossRef Link) PMid:20369924

[7] X. Zhang, J. He and Q. Wei, "EDDK: Energy-Efficient Distributed Deterministic Key Management for Wireless Sensor Networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2011, Article No. 12, Jan. 2011.

Article (CrossRef Link)
[8] A. Perrig, R. Szewczyk, V. Wen, D. E. Culler, and J. D. Tygar, "SPINS: Security Protocols for Sensor Networks," in *Proc. of ACM Mobile Computing and Networking*, pp. 189–199, 2001.
Article (CrossRef Link)
[9] S. Lee and J. Kim, "Design of Authentication Protocol for LR-WPAN using Pre-Authentication Mechanism," in *Proc. of The Sixth IEEE Consumer Communications and Networking Conference*, pp. 1-5,.10-13 Jan. 2009.
Article (CrossRef Link)
[10] B. Tian, S. Han, L. Liu, S. Khadem, and S. Parvin, "Towards Enhanced Key Management in Multi-phase ZigBee Network Architecture," *Computer Communications*, vol. 35, pp. 579-588, 2012.
Article (CrossRef Link)
[11] E. Yüksel, H. R. Nielson, and F. Nielson, "A Secure Key Establishment Protocol for ZigBee Wireless Sensor Networks," *The Computer Journal*, vol. 54, no. 4, pp. 589-601, 2011.
Article (CrossRef Link)
[12] G. Dini and M. Tiloca, "Considerations on Security in ZigBee Networks," in *Proc. of 2010 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, pp.58-65, 7-9 June, 2010.
Article (CrossRef Link)
[13] D. Whiting, R. Housley, and N. Ferguson, "Counter with CBC-MAC (CCM)," RFC 3610, Sep. 2003.
Article (CrossRef Link)
[14] H. Krawczyk, M. Bellare, and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication," RFC 2104, Feb. 1997.
Article (CrossRef Link)
[15] D. Wallner, E. Harder, and R. Agee, "Key Management for Multicast: Issues and Architectures," IETF, RFC 2627, 1999.
Article (CrossRef Link)
[16] D. McGrew, A. David, T. Alan, and A. Sherman, "Key Establishment in Large Dynamic Groups using One-way Function Trees," *IEEE Transactions on Software Engineering*, vol.29, no.5, pp. 444-458, May 2003.
Article (CrossRef Link)
[17] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, "System Architecture Directions for Networked Sensors," in *Proc. of the Ninth International Conference on Architectural Support for Programming Languages and Operating Systems*, vol. 35, pp. 93-104, Dec. 2000,.
Article (CrossRef Link)
[18] M. Simek and P. Moravek, "Modeling of Energy Consumption of ZigBee Devices in Matlab Tool," Elektrorevue, vol. 2, no. 3, pp. 41-46, 2011.
Article (CrossRef Link)
[19] E. Yüksel, H. R. Nielson, and F. Nielson, "Key Update Strategies for Wireless Sensor Networks," *International Journal of Information and Electronics Engineering*, vol. 2, no. 2, pp. 141-145, Mar. 2012.
Article (CrossRef Link)
[20] L. Chen, "Recommendation for Key Derivation using Pseudorandom Functions," Revised NIST Special Publication 800-108, Oct. 2008.
Article (CrossRef Link)

Chang-Seop Park has been with the Department of Computer Science at Dankook University, Republic of Korea, since 1990. He has a Ph.D. and a M.Sc. from Lehigh University (1990 and 1987), as well as a B.A. from Yonsei University (1983). He has been working on the wireless mobile network security during the last 5 years. His research interests include network security, cryptographic protocols, and coding theory.

Bong-Hwan Kim has received his B.Sc. in computer science from Dankook University in 2012 and has been completing his M.Sc. in computer science. His research interests include network security, wireless security, and wireless body area network.