symmetry

MDPI

Article

An Improved Protocol for the Password Authenticated Association of IEEE 802.15.6 Standard That Alleviates Computational Burden on the Node

Jie Zhang [1,2,†], Xin Huang [1,*,†], Paul Craig [1], Alan Marshall [2] and Dawei Liu [1]

1 Department of Computer Science and Software Engineering, Xi'an Jiaotong-Liverpool University, Suzhou 215123, China; Jie.Zhang3@liverpool.ac.uk (J.Z.); P.Craig@xjtlu.edu.cn (P.C.); Dawei.Liu@xjtlu.edu.cn (D.L.)

2 School of Electrical Engineering and Electronics and Computer Science, University of Liverpool, Liverpool L69 3BX, UK; Alan.Marshall@liverpool.ac.uk

* Correspondence: xin.huang@xjtlu.edu.cn; Tel.: +86-512-8816-1511

† These authors contributed equally to this work.

Academic Editor: Young-Sik Jeong

Received: 12 July 2016; Accepted: 10 November 2016; Published: 17 November 2016

Abstract: The IEEE Std 802.15.6 is an international standard for wireless body area networks (WBANs). It contains many aspects of communications, and also provides security services, since some communications in WBANs can carry sensitive information. In this standard, the password authenticated association is a protocol for two participants to identify each other and establish a new master key based on a pre-shared short password. However, recent research shows that this protocol is vulnerable to several attacks. In this paper, we propose an improved protocol which can resist all of these attacks. Moreover, the improved protocol alleviates computational burden on one side of the two participants, the node, which is usually less powerful compared with the other side, the hub.

Keywords: body area networks; password authenticated association; security; key establishment; IEEE 802.15.6

1. Introduction

A wireless body area network (WBAN) is a wireless network of wearable computing devices including implanted devices embedded inside the body or attached on the skin, and accompanied devices which humans can carry by hand, in clothes pockets or in bags [1–4]. WBAN applications [5,6] are growing and becoming more indispensable in people's lives due to the increasing accessability of network service and computing devices. Despite the great progress in networking and computing technology, security is one significant factor that influences users' choice of WBAN applications, since such applications involve a lot of personal information and therefore are vulnerable to security issues.

IEEE Standard (Std) 802.15.6 [7] is an international standard for wireless communication between nodes and hubs in WBANs. It provides strong security for communications that carry sensitive information. In the security services of this standard, the security association procedure activates a pre-shared or generates a new shared master key (MK) between a node and a hub. Several security association protocols suitable for a variety of use cases are provided in this standard. Among these protocols, password authenticated association [8,9] is a protocol for a node and a hub to generate a new shared MK from a pre-shared secret, i.e., the password. However, recent research shows that this protocol is vulnerable to several attacks, such as Man-in-the-Middle and impersonation attacks illustrated in [10], and the off-line dictionary attack and there being a lack of forward secrecy,

which are discussed in [11,12]. To eliminate these attacks, the authors in [10] also proposes a modified version to this protocol.

In this paper, an improved password authenticated association protocol is proposed. In the rest of this paper, we denote this protocol by the *improved protocol*, protocol in [10] by the *modified protocol* and protocol in the IEEE 802.15.6 standard by the *standard protocol*. Compared with the modified protocol and the standard protocol, the improved protocol eliminates all the above attacks on one hand. Moreover, it alleviates computational burden on the node. Since the node usually has limited computational power compared with the hub, the improved protocol is meaningful in practise.

The remaining part of this paper is organized as follows: Section 1 contains preliminaries and symbols that are useful in this paper. In Section 3, we review the standard protocol and available attacks in literature. In Section 4, the improved protocol is proposed and its security and performance are analyzed in Sections 5 and 6, respectively. Section 7 shows a use case of this improved protocol. Related works are provided in Section 8. Finally, Section 9 concludes this paper.

2. Preliminaries and Symbols

2.1. Elliptic Curve Public Key Cryptography

2.1.1. Elliptic Curve

The IEEE 802.15.6 password authenticated association protocol is based on the Diffie–Hellman key exchange [13] employing the elliptic curve public key cryptography (ECC). An elliptic curve E can be characterized by the following equation [14,15]:

$$\begin{array}{l} y^2 \equiv x^3 + ax + b \bmod p \\ \text{with } a, b \in GF(p), 4a^3 + 27b^2 \neq 0 \end{array} \tag{1}$$

where (x, y) is a point on the curve; a and b are coefficients; p is an odd prime; and $GF(p)$ is a prime finite field. For the choices of a suitable elliptic curve, the IEEE Std 802.15.6 suggests using Curve p-256 in FIPS Pub 186-3. Values of a, b, p, the base point $G = (G_x, G_y)$ and the order r of G are given in the standard.

2.1.2. Elliptic Curve Diffie–Hellman

Elliptic curve Diffie–Hellman (ECDH) is an anonymous key agreement protocol that allows two parties, each having an elliptic curve public-private key pair, to establish a shared secret over an insecure channel [16]. Suppose SK_A and SK_B are private keys of two communicating parties A and B, respectively. SK_A and SK_B are random integers from the set $\{1, ..., r-1\}$. The corresponding public keys PK_A and PK_B are computed as follows:

$$PK_A = SK_A \times G, PK_B = SK_B \times G \tag{2}$$

where $\times$ denotes scalar multiplication of G by an integer. In the ECDH protocol, A and B exchange their public keys and compute $(x_k, y_k) = SK_A \times PK_B$ and $(x_k, y_k) = SK_B \times PK_A$, respectively. The shares key is x_k, i.e., the X coordinate of the point.

2.2. Password Authenticated Key Exchange

The password authenticated association protocol in the IEEE 802.15.6 standard is a variation of password authenticated key exchange (PAKE) [8]. A PAKE protocol uses a pre-shared password for an authenticated key establishment. The password is usually short and easy for humans to remember, and is not stored directly in the memory of physical devices for security purpose. Instead, it is input by the users at the beginning of each run of the PAKE protocol.

2.3. Symbols

The association protocol is initiated by the node to generate a shared master key with the hub from a pre-shared password between them. We denote the node as the initiator and the hub as the responder. Some other symbols used in this paper are summarized in Table 1.

Table 1. Symbols and definitions.

Symbol	Meaning
I	identity of the initiator (i.e., the node)
R	identity of the responder (i.e., the hub)
A	identity of an adversary
PW	the pre-shared password
K	the temple Diffie–Hellman key used for computing CMAC
MK	the master key to be generated
$\|$	concatenation of bit strings
SK_I, PK_I	private and public keys of the initiator
SK_R, PK_R	private and public keys of the responder
SK_A, PK_A	private and public keys of the adversary
N_I	a nonce generated by the initiator
N_R	a nonce generated by the responder
N_A	a nonce generated by the adversary
$Q(x)$	a function that maps a positive integer x to a point on the elliptic curve
G	base point in the elliptic curve
$\times$	scalar multiplication
$RMB_n(x)$	the n rightmost bits of x
$LMB_n(x)$	the n leftmost bits of x

3. IEEE 802.15.6 Password Authenticated Association Protocol

We review the IEEE 802.15.6 password authenticated association protocol, i.e., the standard protocol, and discuss its vulnerabilities in this section.

3.1. Description of the Standard Protocol

3.1.1. Set-Up

The initiator and the responder set up their private and public key as follows:

1. Initiator chooses a random SK_I and computes the public key $PK_I = SK_I \times G$.
2. Responder selects its private key SK_R and computes $PK_R = SK_R \times G$.

3.1.2. Master Key Generation

The initiator and the responder execute the following steps to generate a shared master key.

1. The initiator computes a password-scrambled public key

$$PK'_I = PK_I - Q(PW) \tag{3}$$

and sends it to the responder along with a nonce N_I and the identities I and R:

$$M_1 = \{R, I, N_I, PK'_I\}$$

2. After receiving M_1, the responder sends the identities, a nonce and its public key back to the initiator:

$$M_2 = \{I, R, N_R, PK_R\}$$

3. The responder recovers PK_I as follows:

$$PK_I = PK'_I + Q(PW) \tag{4}$$

The initiator and the responder compute the Diffie–Hellman key, respectively, through

$$K = SK_I \times PK_R = SK_R \times PK_I \tag{5}$$

The responder computes a message authentication code

$$MAC_3 = CMAC_{64}(RMB_{128}(K), I\|R\|N_I\|N_R) \tag{6}$$

and then sends the initiator

$$M_3 = \{I, R, N_R, PK_R, MAC_3\}$$

4. The initiator verifies the received MAC_3. If the verification succeeds, the initiator computes a message authentication code

$$MAC_4 = CMAC_{64}(RMB_{128}(K), R\|I\|N_R\|N_I) \tag{7}$$

and sends the responder

$$M_4 = \{R, I, N_I, PK_I, MAC_4\}$$

5. The responder verifies MAC_4. If the verification succeed, both parties compute and activate their new master key as follows:

$$MK = CMAC_{128}(LMB_{128}(K), N_I\|N_R) \tag{8}$$

3.2. Security Problems

The standard protocol uses the password to hide the public key of the initiator through $PK'_I = PK_I - Q(PW)$ in the first step, so that only the responder can recover PK_I from $PK_I = PK'_I + Q(PW)$. However, the protocol reveals PK_I in M_4 of step 4, which means an eavesdropper who intercepts M_4 can acquire $Q(PW)$. In this case, the password is no longer secret in the following runs of the protocol. This is the reason for the vulnerabilities of the standard protocol. Security problems and attacks to this standard protocol in literature are summarized as follows:

1. Impersonation attack. In [10] the authors illustrate an initiator impersonation attack and a responder impersonation attack to the standard protocol. At the end of these attacks, the attackers successfully establish a master key with one side of the communicating parties, while the other side thinks it has the shared master key with the true participant.
2. Man-in-the-Middle attack. In [10], the authors show that an attacker breaks into the communication between the initiator and the responder and modifies the messages at his/her will. At last, the attacker shares two master keys with the initiator and the responder, respectively, while the initiator and the responder think they have a shared master key. Figure 1 is a time-sequence diagram that illustrates the procedure of man-in-the middle attack against the protocol.
3. Off-line dictionary attack. The authors in [11,12] show that a dictionary attacker who eavesdrops messages between the initiator and the responder in a protocol run can obtain PK'_I and PK_I and compute $Q(PW)$ from $Q(PW) = PK_I - PK'_I$. Then, $Q(PW)$ can be used as a verifier and the attacker can try probable PWs from a dictionary of most probable passwords and check them using $Q(PW)$.
4. Lack of forward secrecy. The author in [11,12] illustrates that if SK_I has been compromised by an attacker, the attacker can acquire the Diffie–Hellman key K through computing $K = SK_I \times PK_R$

and MK from $MK = CMAC_{128}(LMB_{128}(K), N_I \| N_R)$ since PK_R, N_I and N_R are sent in the form of plaintext.

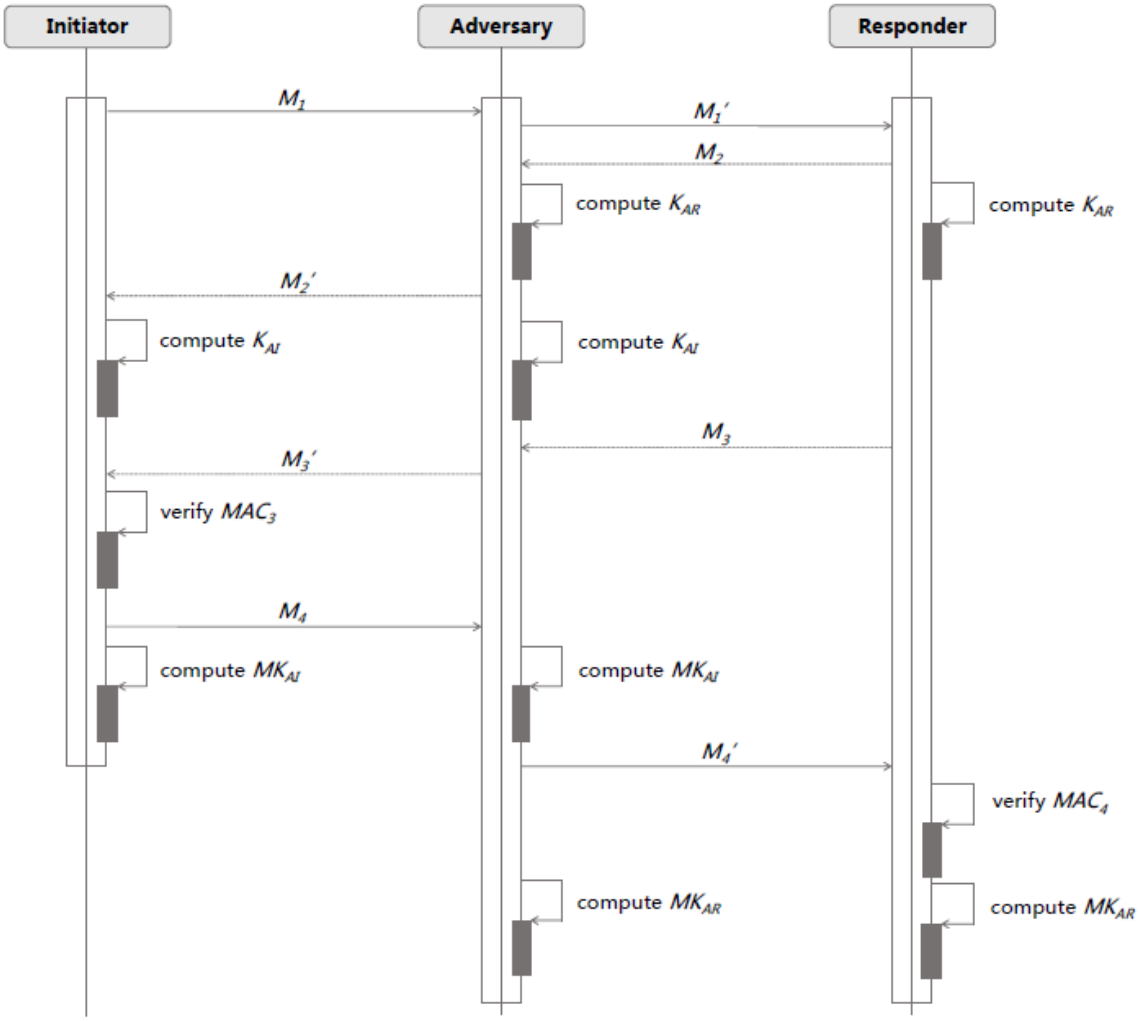

Figure 1. The sequence diagram of Man-in-the-Middle attack.

3.3. The Modified Protocol

The authors in [10] propose a modified protocol to the standard protocol. Specifically, the modified protocol is similar to the standard one except that it does not send PK_I in the clear in M_4. This modification solves most security problems as we mentioned in Section 3.2, but it still fails to provide forward secrecy. We will compare security and performance of the two protocols with those of our new proposed protocol later in this paper.

4. The Improved Protocol

The improved protocol assumes that PK and SK can be reused in each round of protocol. This assumption is reasonable since, in the improved protocol, the temporary Diffie–Hellman key K is derived from two random values chosen by the initiator and the responder, respectively, rather than their public and private keys. The improved protocol is described in detail as follows.

1. The initiator chooses a random value R_I and computes

$$U_I = R_I + SK_I \tag{9}$$

and

$$PK'_I = PK_I - Q(PW) \tag{10}$$

Then, the initiator sends message M_1 to the responder.

$$M_1 = \{I, R, U_I, PK'_I, N_I\}$$

2. The responder chooses a random value R_R and computes

$$U_R = R_R + SK_R \tag{11}$$

and

$$T_R = U_R \times G \tag{12}$$

Then, the responder sends message M_2 to the initiator

$$M_2 = \{R, I, T_R, PK_R, N_R\}$$

3. The responder recovers PK_I as follows:

$$PK_I = PK'_I + Q(PW) \tag{13}$$

The initiator computes the Diffie–Hellman key through

$$K = (T_R - PK_R) \times R_I = G \times R_R \times R_I \tag{14}$$

The responder computes K as follows

$$K = (U_I \times G - PK_I) \times R_R = G \times R_R \times R_I \tag{15}$$

With the K, the responder computes a message authentication code

$$MAC_3 = CMAC_{64}(RMB_{128}(K), I\|R\|N_I\|N_R) \tag{16}$$

and then sends the initiator

$$M_3 = \{I, R, N_R, PK_R, MAC_3\}$$

4. The initiator verifies the received MAC_3. If the verification succeeds, the initiator computes a message authentication code

$$MAC_4 = CMAC_{64}(RMB_{128}(K), R\|I\|N_R\|N_I) \tag{17}$$

and sends the responder

$$M_4 = \{R, I, N_I, MAC_4\}$$

5. The responder verifies MAC_4. If the verification succeeds, both parties compute and activate their new master key as follows:

$$MK = CMAC_{128}(LMB_{128}(K), N_I\|N_R) \tag{18}$$

5. Security Analysis

In Section 3.2, we listed all the attacks to the standard protocol, and in this section, we will prove the security of the improved protocol under all of these attacks.

5.1. Impersonation Attack

Proposition 1. *Suppose the initiator and the responder have shared a password PW secretly, and an attacker is not able to impersonate the initiator to establish the master key MK with the responder.*

Proof. Assume A_I is an attacker who attempts to impersonate the initiator and establish MK with the responder. A_I attacks the protocol as follows:

1. A_I initializes the protocol with the responder by sending the first message M_{A1} as follows:

$$M_{A1} = \{I, R, U_A, PK'_I, N_A\}$$

where $U_A = R_A + SK_A$ and R_A and N_A are random values generated by A_I.

2. After receiving M_{A1}, the responder chooses a random value R_R and computes $U_R = R_R + SK_R$ and $T_R = U_R \times G$. Then, the responder replies A_I with M_2:

$$M_2 = \{R, I, T_R, PK_R, N_R\}$$

3. The responder recovers PK_I and computes $K = (U_A \times G - PK_I) \times R_R$. Then, the responder computes $MAC_3 = CMAC_{64}(RMB_{128}(K), I\|R\|N_A\|N_R)$ and sends the following message M_3 to A_I:

$$M_3 = \{I, R, N_R, PK_R, MAC_3\}$$

4. At this step, A_I needs to send the responder with MAC_{A4}, which should be equivalent with $CMAC_{64}(RMB_{128}(K), R\|I\|N_R\|N_A)$ so that it can pass the verification at the beginning of the next step.

In step 4, in order to compute a valid MAC_{A4}, A_I has to calculate K equals $K = (U_A \times G - PK_I) \times R_R = (R_A \times G + PK_A - PK_I) \times R_R$. However, without any of PK_I and R_R, A_I has no choice but to guess such a MAC_{A4}. The probability of guessing a valid MAC_{A4} is $\frac{1}{2^{64}}$.

Alternatively, in the first piece of message M_{A1}, the adversary A_I can send a U_I intercepted in previous protocol runs instead of U_A. In this case, K computed by the responder in step 3 equals $(U_I \times G - PK_I) \times R_R$, $G \times R_I \times R_R$ and $(T_R - PK_R) \times R_I$. It is still infeasible for A_I to compute the K since R_R and R_I are unknown to A_I.

From the above analysis, now we can draw the conclusion that the probability for A_I successfully impersonating the initiator and establishing a master key with the responder is $\frac{1}{2^{64}}$, which is a minor value in a life circle of a normal node in WBAN applications. □

Proposition 2. *Suppose the initiator and the responder have shared a password PW secretly, and an attacker is not able to impersonate the responder to establish the master key MK with the initiator.*

Proof. Assume A_R is an attacker who intends to impersonate the responder and establish MK with the initiator. A_R attacks the protocol as follows:

1. The initiator sends A_R with M_1, which is the same with the step 1 in the improved protocol:

$$M_1 = \{I, R, U_I, PK'_I, N_I\}$$

2. After receiving M_1, A_R replies the initiator with M_{A2}:

$$M_{A2} = \{R, I, T_A, PK_A, N_A, \}$$

with $T_A = U_A \times G$ and $U_A = R_A + SK_A$, where SK_A is the private key of A_R and R_A and N_A are random values generated by A_R.

3. At this step, A_R needs to send the initiator with MAC_{A3} involved in M_{A3}, so that it can pass the verification at the beginning of the next step.

The MAC_{A3} is checked to be valid only if it equals $CMAC_{64}(RMB128(K), I\|R\|N_I\|N_A)$. In order to generate a valid MAC_{A3}, A_R can compute the CMAC output by inputting K, I, R, N_I, N_A or guess the 64-bit result. To compute the CMAC output, A_R has to calculate K that equals the K calculated by the initiator through $K = (T_A - PK_A) \times R_I = G \times R_A \times R_I$. However, since R_I is unknown to A_R, it is infeasible for A_R to acquire a valid K. Therefore, the adversary can only guess a valid MAC_{A3} with a successful probability at $\frac{1}{2^{64}}$. Otherwise, the protocol will stop at the beginning of step 4 and the attack will fail.

□

From Propositions 1 and 2, we can see impersonation attacks fail no matter if the attacker impersonates the initiator or the responder.

5.2. Man-in-the-Middle Attack

Proposition 3. *Suppose the initiator and the responder have successfully shared a password PW, a Man-in-the-Middle attacker is not able to complete the improved protocol between the initiator and the responder without being detected.*

Proof. Suppose A is a Man-in-the-Middle attacker between the initiator and the responder. A participants the improve protocol as follows:

1. The initiator sends A with M_1 which is the same with M_1 in the improved protocol:

$$M_1 = \{I, R, U_I, PK'_I, N_I\}.$$

2. A replaces M_1 with M_{1A} and sends it to the responder:

$$M_{A1} = \{I, R, U_A, PK'_I, N_A\}.$$

3. The responder replies A with M_2 which is the same with M_2 in the improved protocol:

$$M_2 = \{R, I, T_R, PK_R, N_R\}.$$

4. A sends M_{A2} to the initiator:

$$M_{A2} = \{R, I, T_A, PK_A, N_A\}.$$

5. At this step, the Diffie–Hellman key K_{IA} between A and the initiator and K_{RA} between A and the responder are determined. Specifically, the initiator calculates $K_{IA} = (T_A - PK_A) \times R_I = G \times R_A \times R_I$, and the responder calculates $K_{RA} = (U_A \times G - PK_I) \times R_R = (R_A \times G + PK_A - PK_I) \times R_R$.

   The responder computes $MAC_3 = CMAC_{64}(RMB_{128}(K_{RA}), I\|R\|N_A\|N_R)$ and sends A with M_3:

$$M_3 = \{I, R, N_R, PK_R, MAC_3\}$$

6. A should send the initiator with

$$M_{A3} = \{I, R, N_A, PK_A, MAC_{A3}\}.$$

   where $MAC_{A3} = CMAC_{64}(RMB_{128}(K_{IA}), I\|R\|N_I\|N_A)$

7. The initiator verifies MAC_{A3}.

8. A should send the responder with

$$M_{A4} = \{R, I, N_A, MAC_{A4}\},$$

where $MAC_{A4} = CMAC_{64}(RMB_{128}(K_{RA}), I\|R\|N_A\|N_R)$.

9. The responder verifies MAC_{A4}.

Since A does not have any of R_I, R_R, PK'_I, it is infeasible for A to compute K_{IA} and K_{RA}, and therefore A can not compute correct MAC_{A3} in step 3_A and MAC_{A4} in step 4_A. Without valid MAC_{A3} and MAC_{A4}, the initiator will stop the protocol at the beginning of step 4, and the responder will stop at the beginning of step 5, which means A fails to establish an MK either with the initiator or the responder. □

5.3. Off-Line Dictionary Attack

Proposition 4. *Suppose the initiator and the responder have successfully shared a password PW, and a passive eavesdropper who records one or more sessions of the improved protocol cannot eliminate a significant number of possible passwords.*

Proof. In the improved protocol, values that are sent in the clear include I, R, U_I, PK'_I, N_I, T_R, PK_R, N_R, MAC_3 and MAC_4. In order to carry out an off-line dictionary attack, the adversary needs to acquire information that can help him to check possible passwords from a dictionary. Among all of these values sent in the clear, PW has a relationship only with PK'_I through the equation $PK'_I = PK_I - Q(PW)$. PK_I is kept secretly in the protocol, and $PK_I = SK_I \times G$, where SK_I is a random integer. Therefore, PK_I is a random value and is unknown to the adversary. The equation of $PK'_I = PK_I - Q(PW)$ and the value of PK'_I do not give more information of PW to the attacker. Based on this acquired knowledge, the attacker is unable to eliminate possible passwords. □

According to Proposition 4, an off-line dictionary attack to the improved protocol is infeasible.

5.4. Forward Secrecy

Proposition 5. *Suppose the initiator and the responder have successfully shared a password PW, and compromise of the long-term secret keys of a set of principals does not compromise the MKs established in previous runs of the improved protocol involving those principals.*

Proof. The principals of this protocols are the initiator and the responder, and the long-term secret keys of these principals are the private keys SK_I and SK_R, the password PW and the public key PK_I that is masked during transmission. Assume the adversary A compromises these long-term secrets of the initiator and the responder, and then (s)he has SK_I, SK_R, PW and PK_I. In order to calculate an MK established in a previous run, A needs to compute MK from the formula $MK = CMAC_{128}(LMB_{128}(K), N_I\|N_R)$, where K is a necessary input in that run. Note that A can not use these values to run the protocol with the principals, since, in this case, the MK does not belong to a previous run but is established in the current run. Therefore, A has to compute K through $K = (T_R - PK_R) \times R_I$, $K = (U_I \times G - PK_I) \times R_R$ or $K = G \times R_R \times R_I$. All three of the formulas require at least one of R_I and R_R. However, R_I and R_R are random values chosen by the initiator and the responder, respectively, in each run of the protocol, which means that these values change in every protocol run and are kept unknown to A. Without any of R_I and R_R, A fails to compromise the MK, although (s)he compromises all the long-term secret keys and values. □

From Proposition 5, we can see the improved protocol provides forward secrecy.

6. Performance

In order to observe the performance of the improved protocol, we evaluate the computation and communication cost theoretically. In addition, we also test the performance through a set of experiments.

6.1. Evaluation

The overall burden of the protocol contains three parts: communication cost, computation cost on the node and computation cost on the hub. For the communication cost, we count all of the pieces of messages transmitted between the node and the hub within a run of the protocol. In order to evaluate the computation cost, we count the number of cryptographic algorithm CMAC and scalar multiplication of an element from the ecliptic curve by an integer, since other operations such as addition and subtraction require minor computation cost.

Denote the cost of transmitting a piece of message by $\mathcal{M}$, the cost of executing one CMAC algorithm by $\mathcal{H}$, and the cost of executing the operation of scalar multiplication one time by $\mathcal{S}$, and we compare the evaluated cost of the improved protocol with the modified protocol and the standard protocol in Table 2.

Table 2. Evaluation of performance.

Protocol	Computation Cost on Node	Computation Cost the Hub	Total Computation Cost	Communication Cost
improved protocol	$\mathcal{S}+2\mathcal{H}$	$3\mathcal{S}+2\mathcal{H}$	$4\mathcal{S}+4\mathcal{H}$	$4\mathcal{M}$
modified protocol	$2\mathcal{S}+2\mathcal{H}$	$2\mathcal{S}+2\mathcal{H}$	$4\mathcal{S}+4\mathcal{H}$	$4\mathcal{M}$
standard protocol	$2\mathcal{S}+2\mathcal{H}$	$2\mathcal{S}+2\mathcal{H}$	$4\mathcal{S}+4\mathcal{H}$	$4\mathcal{M}$

From Table 2, we can see that the improved protocol reduces computation cost on the node, while overall computation and communication cost does not increase. One time-consuming operation $\mathcal{S}$ is done by the hub on behalf of the node. Since the hub is more powerful compared with the node, the improved protocol is more affordable for WBAN applications.

6.2. Experiments

The improved protocol contains the algorithm of CMAC and ECC key-generation (generating a private key and using scalar multiplication to compute the public key). We test the runtime of these algorithms on the node through a set of experiments. In the experiments, we use Arduino Uno as the node, SHA-256 as the CMAC algorithms and the ATECC108A crypto chip from Atmel to execute the ECC key-generation. The elliptic curve is curve p-256 in Federal Information Processing Standards (FIPS) Pub 186-3. Description of the node is listed in Table 3, and the results are summarized in Table 4.

Table 3. Details of the node (implemented on Arduino Uno).

Micro Controller	16 MHz, 8 bit (ATmega328)
SRAM	2 KB
EEPROM	1 KB
Flash memory	32 KB (bootloader 0.5 K)

Table 4. Run-time of involved cryptographic algorithms on the node.

Algorithm	Length of Keys (Bits)	Runtime (ms)
ECC key generation	–	48
SHA-256	512	3

From Table 4, we can see that the run-time of executing these algorithms is affordable for the node, which means that the improved protocol is suitable for WBAN applications.

7. Use Case

As described before, our improved protocol reduced the computational burden on one side of communication. This is a significant strength for some applications in wireless sensor networks. Here, we describe a smart lock system that uses our improved protocol to generate a master key. The specific system and the usage of the improved protocol are described as follows.

7.1. Smart Lock System

As is shown in Figure 2, the smart lock system consists of a lock which is a physical host embedded with a computational device, and a phone which has installed a smart lock application. The aim of this system is using this phone application to securely lock or unlock the lock. Obviously, the computationally limited lock is the initiator and the relatively powerful phone is the responder. The smart lock system includes the following three phases, and our protocol is involved in the first phase.

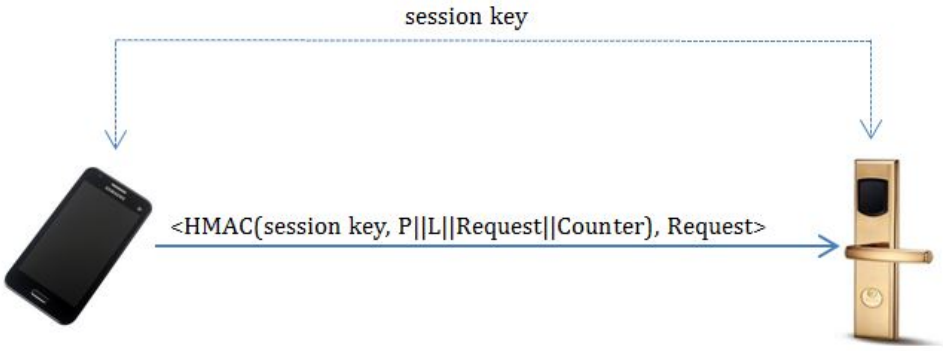

Figure 2. Smart lock system.

1. **Master Key Generation.** The lock and the phone secretly input the short password and then execute our improved protocol. After this stage, a relatively long master key is shared by the lock and the phone.
2. **Session Key Generation.** With the master key, the lock and the phone execute the session key generation protocol (such protocols are available in literature) to generate their session key for this round of communication.
3. **Secure Communication.** The newly generated session key is used for this round of communication between the phone and the lock. We describe the steps as:

   (1) The phone computes

   $$MAC = HMAC(sessionkey, P\|L\|Request\|Counter)$$

   and sends the request ($LOCK/UNLOCK$) with the MAC to the lock. Here, P and L denote the identity of the phone and the lock, and $Counter$ denotes the value of counter.

   (2) The lock verifies the MAC. If the verification succeeds, the lock executes the request to lock or unlock; otherwise, it does not execute the request or responds with a failure message.

7.2. Analysis

The smart lock system is secure since the session key is kept secretly by the two participants. An adversary can not request the system to lock or unlock because they can not compute the correct *MAC* without the session key. Therefore, the security of the session key is significant for the security of the whole system. Our improved protocol provides secure generation for the master key, which, in turn, guarantees the security of the session key.

Additionally, the device embedded in the lock is a less powerful device compared with a normal cell phone. Our password-based authenticated association protocol in the first phase reduces the computational cost of the lock, which makes the smart lock system more practicable.

8. Related Works

8.1. Comparison

In Section 6.1, we compared the cost of the improved protocol with other related protocols in Table 2. The comparison in terms of security of these protocols is listed in Table 5, where $\surd$ means being secure under the corresponding attacks or providing the corresponding security feature, while $\times$ means being insecure or not providing.

Table 5. Comparison of security ("$\surd$" denotes the protocol resist the attack or possess the security feature, and "$\times$" denotes the the protocol does not resist the attack or does not possess the security feature).

Attacks/Security Feature	Improved Protocol	Modified Protocol	Standard Protocol
Impersonation attack	$\surd$	$\surd$	$\times$
Man-in-the-Middle attack	$\surd$	$\surd$	$\times$
Off-line dictionary attack	$\surd$	$\surd$	$\times$
Forward secrecy	$\surd$	$\times$	$\times$

8.2. Password-Based Two-Party Key Exchange

Several password-based authenticated key exchange protocols have been proposed. In this subsection, we compare our improved protocol with three kinds of two-party key exchange protocols that are based on passwords.

8.2.1. Encrypted Key Exchange Using Diffie–Hellman

Diffie–Hellman-based Encrypted Key Exchange (EKE) protocols transmit the public keys encrypted using the password. The original protocol is proposed by Bellovin and Merritt in [17]. Variants and extensions of this protocol have been proposed. Such protocols are proved to be secure in the random-oracle model. However, in practice, attacks against these protocols exist since the two parties are not able to verify the integrity of the received messages. If an attacker maliciously modifies the message, the two participants will generate different keys while they are not aware.

The IEEE Std password authenticate association protocol and our improved protocol are developed from these kinds of protocols. As in the IEEE std protocol and our improved protocol Hash-based Message Authenticated Code (HMAC) is used for verifying the integrity of messages transmitted between the two parties, the above attacks against the original Diffie–Hellman-based EKE protocols are eliminated.

8.2.2. RSA-Based Protocols

Rivest-Shamir-Adleman (RSA)-Based Protocols use the RSA algorithm as the basis of the password authentication key exchange scheme. In [18], MacKenzie proposed a variant of RSA based open key exchange protocol called SNAPI (Secure Network Authentication with Password Information).

Verification for the integrity of transmitted messages is involved in this protocol. However, this protocol is not suitable for wireless sensor networks since sensors are usually not powerful enough to run the RSA algorithm.

8.2.3. Protocols Using a Server Public Key

Some password-based authenticated key exchange protocols use a server public key in addition to the pre-shared password. Such protocols include the Gong-Lomas-Needham-Saltzer (GLNS) compact protocol proposed by Gong et al. in [19], Gong's Optimal GLNS nonce-based protocol in [20], Kwon–Song Protocol in [21] and Halevi–Krawczyk Protocol in [22]. However, all four of the protocols used public key encryption, which is too high in computational cost for sensor devices. Moreover, the former two protocols need the participation of a server.

9. Conclusions

In low-power, low-complexity wireless sensor network applications such as WBANs, the communications security requirements mainly include authentication between participants, as well as confidentiality and integrity of transmitted messages. Mechanisms that aim to satisfy these requirements usually need a secret key to be held by participants. Therefore, key establishment and management are significant for security services in communications networks. The password authenticated association protocol is a scheme for the participants to generate a master key from a pre-shared password.

Considering the asymmetric power of the two participants in WBANs, we propose an improved password authenticated association protocol that reduces the computational cost on the less powerful participant of communication. The improved protocol can resist both impersonation attacks and Man-in-the-Middle attacks. A master key between the node and the hub will be established securely and efficiently through this protocol, and, afterwards, this is used for pairwise temporal key (PTK) creation, and the PTK is the key used in encryption and decryption process to provide authentication, confidentiality and integrity for communication.

The improved protocol requires one scalar multiplication and two HMAC computations on the nodes (i.e., the initiator). Since the computational costs of these algorithms are acceptable to devices with limited power in WBANs, the improved protocol is suitable for applications in WBANs.

Acknowledgments: This work has been supported by the Xi'an Jiaotong-Liverpool University research development fund projects RDF140243 and RDF150246, as well as by the Suzhou Science and Technology Development Plan under grant SYG201516, and Jiangsu Province National Science Foundation under grant BK20150376.

Author Contributions: Jie Zhang and Xin Huang attacked the standard protocol, and conceived and designed the improved protocol; Paul Craig and Alan Marshall analyzed the result; Jie Zhang wrote the paper; and Paul Craig and Alan Marshall modified the English language of the manuscript; Dawei Liu and Xin Huang contributed analysis tools.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

WBAN	Multidisciplinary Digital Publishing Institute
PTK	Pairwise Temporal Key
MAC	Message Authentication Code
EKE	Encrypted Key Exchange
SNAPI	Secure Network Authentication with Password Information
CMAC	Cypher-based message authentication code
SRAM	Static Random Access Memory
EEPROM	Electrically Erasable Programmable Read-Only Memory
SHA	Secure Hash Algorithm

References

1. Huang, X.; Chen, B.; Markham, A.; Wang, Q.; Yan, Z.; Roscoe, A.W. Human interactive secure key and identity exchange protocols in body sensor networks. *Inf. Sec.* **2013**, *7*, 30–38.
2. Huang, X.; Wang Q.; Chen, B.; Markham, A.; Jäntti, R.; Roscoe, A.W.F. Body sensor network key distribution using human interactive channels. In Proceedings of the 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies, Barcelona, Spain, 26–29 October 2011.
3. Ullah, S.; Higgins, H.; Braem, B.; Latre, B.; Blondia, C.; Moerman, I.; Saleem, S.; Rahman, Z.; Kwak, K.S. A comprehensive survey of wireless body area networks. *J. Med. Syst.* **2012**, *36*, 1065–1094.
4. Movassaghi, S.; Abolhasan, M.; Lipman, J.; Smith, D.; Jamalipour, A. Wireless body area networks: A survey. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 1658–1686.
5. Abdmeziem, M.R.; Tandjaoui, D. An end-to-end secure key management protocol for e-health applications. *Comput. Electr. Eng.* **2015**, *44*, 184–197.
6. Jovanov, E. Wireless technology and system integration in body area networks for m-health applications. In Proceedings of the 27th Annual International Conference of the Engineering in Medicine and Biology Society, Shanghai, China, 1–4 September 2005; pp. 7158–7160.
7. IEEE Standards. IEEE Standard for Local and Metropolitan Area Networks-Part 15.6: Wireless Body Area Networks. 2012. Available online:http://standards.ieee.org/about/get/802/802.15.html (accessed on 29 February 2012).
8. Boyd, C.; Mathuria, A. *Protocols for Authentication and Key Establishment*; Springer Science & Business Media: Berlin, Germany, 2013.
9. Abdalla, M. Password-based authenticated key exchange: An overview. In *Provable Security*; Springer: Berlin, Germany, 2014; pp. 1–9.
10. Huang, X.; Liu, D.; Zhang, J. An improved IEEE 802.15.6 password authenticated association protocol. In Proceedings of the 4th IEEE/CIC International Conference on Communications in China (ICCC 2015), Shenzhen, China, 2–4 November 2015.
11. Toorani, M. Security analysis of the IEEE 802.15.6 standard. *Int. J. Commun. Syst.* **2016**, doi:10.1002/dac.3120.
12. Toorani, M. On vulnerabilities of the security association in the IEEE 802.15. 6 standard. In *Financial Cryptography and Data Security*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 245–260.
13. Diffie, W.; Hellman, M.E. New directions in cryptography. *IEEE Trans. Inf. Theory* **1976**, *22*, 644–654.
14. Koblitz, N. Elliptic curve cryptosystems. *Math. Comput.* **1987**, *48*, 203–209.
15. Miller, V. Use of elliptic curves in cryptography. In *Advances in Cryptology, CRYPTO85 Proceedings*; Springer: Berlin/Heidelberg, Germany, 1986; pp. 417–426.
16. Barker, E.; Chen, L.; Roginsky, A.; Smid, M. *Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography*; Technical Report; National Institute of Standards and Technology (NIST): Gaithersburg, MD, USA, 2012.
17. Bellovin, S.M.; Merritt, M. Encrypted key exchange: Password-based protocols secure against dictionary attacks. In Proceedings of the 1992 IEEE Computer Society Symposium on Research in Security and Privacy, Oakland, CA, USA, 4–6 May 1992; pp. 72–84.
18. MacKenzie, P.; Patel, S.; Swaminathan, R. Password authenticated key exchange based on RSA. In *Advances in Cryptology-Asiacrypt 2000*; Springer: Berlin/Heidelberg, Germany, 2000; pp. 599–613.
19. Gong, L.; Lomas, M.; Needham, R.M.; Saltzer, J.H. Protecting poorly chosen secrets from guessing attacks. *IEEE J. Sel. Areas Commun.* **1993**, *11*, 648–656.
20. Gong, L. Optimal authentication protocols resistant to password guessing attacks. In Proceedings of the 8th IEEE Computer Security Foundations Workshop, County Kerry, Ireland, 13–15 June 1995; pp. 24–29.
21. Kwon, T.; Song, J. Efficient and secure password-based authentication protocols against guessing attacks. *Comput. Commun.* **1998**, *21*, 853–861.
22. Halevi, S.; Krawczyk, H. Public-key cryptography and password protocols. *ACM Trans. Inf. Syst. Sec. (TISSEC)* **1999**, *2*, 230–268.