An Improved IEEE 802.15.6 Password Authenticated Association Protocol

Xin Huang, Dawei Liu, Jie Zhang
Department of Computer Science and Software Engineering
Xi'an Jiaotong-Liverpool University
Suzhou, China
Email: Xin.Huang@xjtlu.edu.cn, Dawei.Liu@xjtlu.edu.cn, Jie.Zhang03@xjtlu.edu.cn

***Abstract*—A body area network is a network of nodes around human bodies. Body area networks are associated with human body, and collect sensitive information of users. If this information cannot be properly protected, users would stop using these applications. IEEE Std 802.15.6 is a standard for body area networks; it provides several security association protocols. In this paper, password authenticated association protocol in the standard is reviewed. First of all, several attacks are found. In addition, we proposed an improved protocol that can avoid these attacks.**

I. Introduction

Body area network (BAN) is a network of nodes to be deployed in, on, or around the human body. BANs are useful in many applications including sports training and chronic diseases monitoring.

BANs requires security protections. BANs are associated with human body, and collect sensitive information of users. If this information cannot be properly protected, users would stop using these applications. The first step of security protection is usually security association, or cryptographic key establishment procedure.

So far, many BAN association protocols have been proposed. Most of them are based on a pre-deployment phase, in which secrets are distributed. Some other interesting protocols are listed as follows: Balfanz et al. [2], Gehrmann et al [4], Vaudenay [9], Cagalj et al. [3], Wong and Stajano [10], and Huang et al. [5], [8], [7], [6].

IEEE Std 802.15.6 [1] is a standard for BANs, which provides a series of security association protocols. This paper studies the password authenticated association protocol in IEEE 802.15.6 specification. Our main research questions are:

- Is the password authenticated association protocol secure?
- If not, how can we improve it?

After study this protocol, we find several attacks against it. Also, an improved protocol is proposed and evaluated.

The rest of this paper is organized as follows. Section 2 explains password authenticated association protocol. Section 3 describes three attacks against this protocol. Section 4 gives an improved password authenticated association protocol; this improved protocol is also evaluated in this section. Finally, conclusions are made in Section 5.

II. Password Authenticated Association

In this section, we will introduce IEEE 802.15.6 password authenticated association protocol. The protocol establishes a security association between two communicating parties, a node (the initiator I) and a hub (the responder R), for establishing a shared master key (MK).

This security association protocol is based on the Elliptic Curve Diffie-Hellman key exchange protocol. The elliptic curve is characterized as:

$$y^2 = x^3 + ax + b \mod p,$$
$$\text{with } a, b \in GF(p), 4a^3 + 27b^2 \neq 0$$

Relevant symbols in the equation are explained as follows.

- (x, y) is a point in the curve.
- $GF(p)$ is a prime finite field.
- p is an odd prime.
- a and b are coefficients.
- $G = (G_x, G_y)$ is the base point.
- r is the order of the base point G.

Suppose SK_I and PK_I are the private key and public key of I; SK_R and PK_R are the private key and public key of R. Their relations are:

$$PK_I = SK_I \times G,\ PK_R = SK_R \times G$$

where $\times$ denotes scalar multiplication of the base point G by an integer.

Other symbols used in this section are listed below.

- N_I: nonce generated by the initiator
- N_R: nonce generated by the responder
- PK'_I: scrambled public key of the initiator
- RMB$(\cdot)$ and LMR$(\cdot)$ represent the right most 128 bits and the left most 128 bits of a certain input, respectively.
- PW: the password shared between I and R.
- The cipher-based message authentication code (CMAC) algorithm is specified in the NIST Special Publication 800-38B. The notation $CMAC(K, M, L)$ represents the L-bit output of the CMAC applied under key K to message M based on the AES forward cipher function.

This protocol assumes that the initiator and the responder pre-share a password. From the standpoint of security, this protocol can be simplified as follows.

978-1-5090-0243-6/15/$31.00 ©2015 IEEE

1. I computes a password-scrambled public key as follows
$$PK'_I = PK_I - Q(PW)$$
where $Q(PW)$ is a function that maps PW to a point on the elliptic curve. I sends R the identities, a nonce and PK'_I
$$\{R, I, N_I, PK'_I\}$$
2. R responds to I with the identities, a nonce and its public key.
$$\{I, R, N_R, PK_R\}$$
3. R recovers PK_I as follows
$$PK_I = PK'_I + Q(PW)$$
At this stage, I and R compute a shared secret K:
$$K = SK_I \times PK_R = SK_R \times PK_I$$
R computes a message authentication code as follows:
$$M_3 = \text{CMAC}_{64}(\text{RMB}(K), I, R, N_I, N_R)$$
R sends I the identities, the nonce N_R, its public key, and M_3.
$$\{I, R, N_R, PK_R, M_3\}$$
4. I computes a message authentication code as follows:
$$M_4 = \text{CMAC}_{64}(\text{RMB}(K), R, I, N_R, N_I)$$
I sends R the identities, the nonce N_I, its public key, and M_4.
$$\{R, I, N_I, PK_I, M_4\}$$

Finally, each party computes the shared master key MK as follows:
$$MK = \text{CMAC}_{128}(\text{LMB}(K), N_I, N_R)$$

III. Attacks

In this section, we will introduce several attacks against the IEEE 802.15.6 password authenticated association protocol.

A. Initiator Impersonation

The problem of the IEEE 802.15.6 password authenticated association protocol is that the protocol reveals PK_I in step 4. Thus the adversary can impersonate the initiator as follows.

1. A intercepts the step 4.
$$\{R, I, N_I, PK_I, M_4\}$$
Now A knows PK_I. A computes
$$PK'_I - PK_I + PK_A = PK_A - Q(PW)$$
2. A makes R re-run the protocol. A sends R the identities, a nonce and $PK_A - Q(PW)$
$$\{R, I, N_A, PK_A - Q(PW)\}$$
3. R responds to A with the identities, a nonce and its public key $\overline{PK}_R$ in this round.
$$\{I, R, N_R, \overline{PK}_R\}$$
4. R recovers PK_A as follows
$$PK_A = PK_A - Q(PW) + Q(PW)$$
At this stage, A and R compute a shared secret K:
$$K = SK_A \times \overline{PK}_R = \overline{SK}_R \times PK_A$$
where $\overline{SK}_R$ is the private key corresponding to $\overline{PK}_R$. R computes a message authentication code as follows:
$$M_3 = \text{CMAC}_{64}(\text{RMB}(K), I, R, N_A, N_R)$$
R sends A the identities, the nonce N_R, its public key, and M_3.
$$\{I, R, N_R, \overline{PK}_R, M_3\}$$
5. A computes a message authentication code as follows:
$$M_4 = \text{CMAC}_{64}(\text{RMB}(K), R, I, N_R, N_A)$$
A sends R the identities, the nonce N_A, its public key, and M_4.
$$\{R, I, N_A, PK_I, M_4\}$$

Finally, the attacker and the responder (the hub) have a shared master key MK:
$$MK = \text{CMAC}_{128}(\text{LMB}(K), N_A, N_R)$$
The responder thinks that it has the shared master key MK with I, however it actually has the shared master key MK with the attacker.

B. Responder Impersonation

The IEEE 802.15.6 password authenticated association protocol reveals PK_I in step 4. Thus the adversary can also impersonate the responder as follows.

1. A intercepts the step 4.
$$\{R, I, N_I, PK_I, M_4\}$$
Now A knows PK_I. A computes
$$PK_I - PK'_I = Q(PW)$$
2. A makes I re-run the protocol. I sends A the identities, a nonce and $\overline{PK}'_I$ (the scrambled public key of I in this round)
$$\{R, I, N_I, \overline{PK}'_I\}$$
3. A responds to I with the identities, a nonce and its public key.
$$\{I, R, N_A, PK_A\}$$
4. A recovers $\overline{PK}_I$ (the public key of I in this round) as follows
$$\overline{PK}_I = \overline{PK}'_I + Q(PW)$$

At this stage, I and A compute a shared secret K:

$$K = \overline{SK}_I \times PK_A = SK_A \times \overline{PK}_I$$

where $\overline{SK}_I$ is the private key corresponding to $\overline{PK}_I$. A computes a message authentication code as follows:

$$M_3 = \text{CMAC}_{64}(\text{RMB}(K), I, R, N_I, N_A)$$

A sends I the identities, the nonce N_A, its public key, and M_3.

$$\{I, R, N_A, PK_A, M_3\}$$

5. I computes a message authentication code as follows:

$$M_4 = \text{CMAC}_{64}(\text{RMB}(K), R, I, N_A, N_I)$$

I sends A the identities, the nonce N_I, its public key, and M_4.

$$\{R, I, N_I, \overline{PK}_I, M_4\}$$

Finally, the attacker and the initiator (the node) have a shared master key MK:

$$MK = \text{CMAC}_{128}(\text{LMB}(K), N_I, N_A)$$

The initiator thinks that it has the shared master key MK with R, however it actually has the shared master key MK with the attacker.

C. Man-in-the-Middle Attack

The adversary can also initiate the man-in-the-middle attack as follows.

1. A intercepts the step 4.

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

Now A knows PK_I. A computes

$$PK_I' - PK_I + PK_A = PK_A - Q(PW)$$

and

$$PK_I - PK_I' = Q(PW)$$

2. A makes I re-run the protocol. I sends A the identities, a nonce and $\overline{PK}_I'$

$$\{R, I, N_I, \overline{PK}_I'\}$$

2′. A makes R re-run the protocol. A sends R the identities, a nonce and $PK_A - Q(PW)$

$$\{R, I, N_A, PK_A - Q(PW)\}$$

3. R responds to A with the identities, a nonce and its public key.

$$\{I, R, N_R, \overline{PK}_R\}$$

3′. A responds to I with the identities, a nonce and its public key.

$$\{I, R, N_A, PK_A\}$$

4. R recovers PK_A as follows

$$PK_A = PK_A - Q(PW) + Q(PW)$$

At this stage, A and R compute a shared secret K:

$$K = SK_A \times \overline{PK}_R = \overline{SK}_R \times PK_A$$

R computes a message authentication code as follows:

$$M_3 = \text{CMAC}_{64}(\text{RMB}(K), I, R, N_A, N_R)$$

R sends A the identities, the nonce N_R, its public key, and M_3.

$$\{I, R, N_R, \overline{PK}_R, M_3\}$$

4′. A recovers $\overline{PK}_I$ as follows

$$\overline{PK}_I = \overline{PK}_I' + Q(PW)$$

At this stage, I and A compute a shared secret K:

$$K = \overline{SK}_I \times PK_A = SK_A \times \overline{PK}_I$$

A computes a message authentication code as follows:

$$M_3 = \text{CMAC}_{64}(\text{RMB}(K), I, R, N_I, N_A)$$

A sends I the identities, the nonce N_A, its public key, and M_3.

$$\{I, R, N_A, PK_A, M_3\}$$

5. I computes a message authentication code as follows:

$$M_4 = \text{CMAC}_{64}(\text{RMB}(K), R, I, N_A, N_I)$$

I sends A the identities, the nonce N_I, its public key, and M_4.

$$\{R, I, N_I, \overline{PK}_I, M_4\}$$

5′. A computes a message authentication code as follows:

$$M_4 = \text{CMAC}_{64}(\text{RMB}(K), R, I, N_R, N_A)$$

A sends R the identities, the nonce N_A, its public key, and M_4.

$$\{R, I, N_A, PK_I, M_4\}$$

Finally, the attacker and the responder (the hub) have a shared master key:

$$\text{CMAC}_{128}(\text{LMB}(K), N_A, N_R)$$

The attacker and the initiator (the node) have a shared master key:

$$\text{CMAC}_{128}(\text{LMB}(K), N_I, N_A)$$

The initiator and responder thinks that they have a shared master key, however each of them actually has a shared master keys with the attacker.

IV. Improved Protocol

In this section, we will introduce an improved IEEE 802.15.6 password authenticated association protocol. Also, we will analyze this improved protocol.

A. Protocol Description

The difference between the improved protocol and the original version is that PK_I will not be sent in step 4. The whole protocol procedure is introduced as follows.

1. I computes a password-scrambled public key as follows
$$PK'_I = PK_I - Q(PW)$$
where $Q(PW)$ is a function that maps PW to a point on the elliptic curve. I sends R the identities, a nonce and PK'_I
$$\{R, I, N_I, PK'_I\}$$
2. R responds to I with the identities, a nonce and its public key.
$$\{I, R, N_R, PK_R\}$$
3. R recovers PK_I as follows
$$PK_I = PK'_I + Q(PW)$$
At this stage, I and R compute a shared secret K:
$$K = SK_I \times PK_R = SK_R \times PK_I$$
R computes a message authentication code as follows:
$$M_3 = \text{CMAC}_{64}(\text{RMB}(K), I, R, N_I, N_R)$$
R sends I the identities, the nonce N_R, its public key, and M_3.
$$\{I, R, N_R, PK_R, M_3\}$$
4. I computes a message authentication code as follows:
$$M_4 = \text{CMAC}_{64}(\text{RMB}(K), R, I, N_R, N_I)$$
I sends R the identities, the nonce N_I, its public key, and M_4.
$$\{R, I, N_I, M_4\}$$

Finally, each party computes the shared master key MK as follows:
$$MK = \text{CMAC}_{128}(\text{LMB}(K), N_I, N_R)$$

B. Security

Security properties of our new protocol is analyzed below.

1) Authenticity: Firstly, assume that the attacker aims to replace PK_R to PK_A (responder impersonation or in a man-in-the-middle attack). In this case, the attacker need to either compromise CMAC_{64} (the probability is able to ignored) or compute the same K in I, which is $SK_I \times PK_R$. The best way of doing this is to get PK_I from PK'_I, replace PK_R to PK_A, and compute $SK_A \times PK_I$. The probability is no better than the attacker guess the correct PW, which is $1/2^{l(PW)}$ (unless the attacker can do a brute force attack).

Alternatively, the attacker aims to replace PK_I to PK_A (initiator impersonation or in a man-in-the-middle attack). In this case, the attacker need to either compromise CMAC_{64} (the probability is able to ignored) or compute the same K in R, which is $SK_R \times (PK'_A + Q(PW))$. The probability is no better than the attacker guess the correct PW, which is $1/2^{l(PW)}$.

In summary, authenticity of the new protocol is dependent on the strength of the password. However, if the password and $Q(PW)$ are longer than 64 bits, CMAC_{64} becomes the main factor that influence the security of this protocol.

2) The adversary is not able to derive the private key: The only way that the attacker can derive the private key, for example SK_R, is that the attacker compute $SK_R = PK_R/G$. These equivalent to solve the elliptic curve discrete logarithm problem.

3) The adversary is not able to derive the session key: In order to compute the session key, the adversary must know the private key. However, as we analyzed above, the adversary is not able to derive the private key.

4) Known-key security: Suppose a previous session key K_1 is disclosed, the adversary is unable to derive the current session key K_2. The reason is that the initiator and responder re-generate the private and public key in each round.

5) Perfect forward secrecy: Suppose the password is disclosed in the future, the adversary is unable to derive the current session key. The reason is that the session key is computed using temporary public/private keys and nonces, which will be discarded after each protocol run.

C. Performance

In Table I, the performance of our protocol is compared to several other association protocols. $asyI$ is the the number of asymmetric cryptographic computations in I; $asyR$ is the the number of asymmetric cryptographic computations in R; Msg is the number of messages transmitted in the protocol; and related notes are in the last column.

As we can see, the performance of this protocol is acceptable. Given that hub is usually powerful, asymmetric cryptographic computation twice or once will not make much difference. The main advantage of this protocol is that it does not need other communication channels such as visible light communication channels (VLC) or human users compare two 16-bit digest in two devices, which are required in many other association protocols.

Note that manual messages such as button pushing is not counted as a message. '-' means that the protocol has not specified its key exchange method. Correspondingly, $+k$ means that k messages will be added to the number of messages in a key exchange protocol.

V. Conclusion

In this paper, we have provided a review of password authenticated association protocol in the IEEE 802.15.6 standard. Several attacks have been found. In addition, an improved protocol is proposed. Based on our analysis,

Table I
PERFORMANCE

Protocol	asyI	asyR	Msg	Note
Improved protocol	2	2	4	user input password
IETa [5]	2	2	5	user compare digests
IETb [5]	2	2	5	user compare digests
health [8]	2	2	3	user compare digests
HLCa [7]	2	1	3	require VLC
HLCb [7]	2	1	4	require VLC
VLCa [6]	2	2	3	require VLC
VLCb [6]	2	1	5	require VLC
DH-SC [3]	2	2	5	user compare short strings
DH-DB [3]	2	2	$6+k$	k messages for distance bounding
DH-IC [3]	2	2	5	
MANA I [3]	-	-	+1	user enters a random key and a MAC
MANA II [3]	-	-	+2	user compares random key and MAC values
MANA III [3]	-	-	+5	user enters a random key to both devices
Vaudenay [9]	-	-	4	string comparison in out-of-band channels
Wong-Stajano I [10]	2	2	3	hash comparison in out-of-band channels
Wong-Stajano II [10]	2	2	6	string comparison in out-of-band channels
Wong-Stajano III [10]	2	2	4	string comparison in out-of-band channels

this new protocol can prevent attacks we found. This protocol also have some other good security properties, for example, known-key security and perfect forward secrecy. The performance study shows that the computation and communication burden is acceptable, and it does not rely on some special communication channels (for example, visible light communication channels).

In the next stage, we will try to find more improved protocols with good security properties and performance.

ACKNOWLEDGMENT

This work has been supported in part by the XJTLU RDF140243, by the Natural Science Foundation of Jiangsu Province under Grant BK20150376 & BK20140404, by the Jiangsu University Natural Science Research Programme under Grant 13KJB510035, and by the Suzhou Science and Technology Development Plan under Grant SYG201516 & SYG201405.

REFERENCES

[1] Ieee standard for local and metropolitan area networks-part 15.6: Wireless body area networks. 2012.

[2] D. Balfanz, D. K. Smetters, P. Stewart, and H. C. Wong. Talking to strangers: Authentication in ad-hoc wireless networks. In *Proceedings of the 9th Annual Network and Distributed System Security Symposium (NDSS)*, pages 7–19. Citeseer, 2002.

[3] M. Cagalj, S. Capkun, and J. P. Hubaux. Key agreement in peer-to-peer wireless networks. *Proceedings of the IEEE*, 94(2):467–478, 2006.

[4] C. Gehrmann and K. Nyberg. Security in personal area networks. *Security for Mobility*, pages 191–230, 2004.

[5] Xin Huang, Bangdao Chen, Andrew Markham, Qinghua Wang, Zheng Yan, and A William Roscoe. Human interactive secure key and identity exchange protocols in body sensor networks. *Information Security, IET*, 7(1):30–38, 2013.

[6] Xin Huang, Xiong Gao, and Zheng Yan. Security protocols in body sensor networks using visible light communications. *International Journal of Communication Systems*, 2015.

[7] Xin Huang, Shangyuan Guo, Bangdao Chen, and AW Roscoe. Bootstrapping body sensor networks using human controlled led-camera channels. In *Internet Technology And Secured Transactions, 2012 International Conference for*, pages 433–438. IEEE, 2012.

[8] Xin Huang, Qinghua Wang, Chen Bangdao, Andrew Markham, Riku Jäntti, and AW Roscoe. Body sensor network key distribution using human interactive channels. In *Proceedings of the 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies*, page 143. ACM, 2011.

[9] S. Vaudenay. Secure communications over insecure channels based on short authenticated strings. In *Advances in Cryptology–CRYPTO 2005*, pages 309–326. Springer, 2005.

[10] F. L. Wong and F. Stajano. Multichannel security protocols. *IEEE Pervasive Computing*, pages 31–39, 2007.