A Novel Key Management Protocol in Body Area Networks

Jian Shen, Sangman Moh, Ilyong Chung
Dept. of Computer Engineering
Chosun University
Gwangju, Republic of Korea
E-mail: s_shenjian@126.com, {smmoh, iyc}@chosun.ac.kr

***Abstract*—Body Area Networks (BANs) have emerged as an enabling technique for e-healthcare systems, in which the data of a patient's vital body parameters and movements can be collected by small wearable or implantable sensors and communicated using short-range wireless communication techniques. Due to the shared wireless medium between the sensors in BANs, adversaries can launch various attacks on the e-healthcare systems. The security and privacy issues of BANs are getting more and more important. To provide secure and correct association of a group of sensors with a patient and satisfy the requirements of data confidentiality and integrity in BANs, we propose a novel key management protocol based on elliptic curve cryptography (ECC) and hash chains. The authentication procedure and group key generation are very simple and efficient. Therefore, our protocol can be easily implemented into the power and resource constrained sensor nodes in BANs. From the comparison results, furthermore, we can conclude that the proposed protocol dramatically reduces the computation and communication cost for the authentication and key derivation compared with the previous protocols.**

***Keywords*-Body Area Networks (BANs); security; privacy; sensor association; key management;**

I. Introduction

BANs can be used to continuously or remotely monitor patients' health, which have the potential to revolutionize the capture, processing, and communication of critical data for e-healthcare systems. As we know, the modern e-healthcare systems can provide new ways of hospitalization with quality health care. Wirelessly connected medical sensor nodes placed in, on, and around the body form a BAN for continuous, automated, and remote monitoring of physiological signs to support medical applications [1] [2]. In addition, a patient controller (PC) is needed to perform a multitude of functions in BANs. A PC (such as a PDA or smart phone) can sense and fuse data from sensors across the body, serve as a user interface, and bridge BANs to higher-level infrastructures.

The applications of BANs are primarily in the healthcare domain, especially for continuous monitoring and logging vital parameters of patients suffering from chronic diseases such as diabetes, asthma and heart attacks. Moreover, BAN technology is able to support other personalized applications, such as sports, gaming, entertainment, military and so on [3] [4] [5]. A wide range of applications make BANs have a promising future.

Compared with conventional sensor networks, the most distinguished difference of BANs is that a BAN needs to deal with more important medical information. Data confidentiality and integrity are the most important requirements in BANs, since wireless medium is susceptible to lots of security attacks. In this paper, we propose a novel key management protocol to provide secure sensor association and key derivation in BANs. First of all, in order to withstand security attacks from malicious insiders such as off-duty doctors or discharged patients, mutual authentication between a patient and a healthcare worker should be provided. Secondly, secure sensor association scheme should be considered to offer mutual authentication between medical sensors and a patient controller (PC) such that a healthcare worker can make sure that a group of medical sensor nodes are correctly and securely associated with an intended patient. It is worth noting that the sensor nodes also need to authenticate each other and establish a group key for subsequent communications. During the sensor association procedure, each medical sensor node can share a secret key with the PC, which is able to be used to encrypt and transmit the group key. Thirdly, a key management scheme is required to derive the group key. Our proposed protocol provides mutual authentication between a patient and a healthcare worker, mutual authentication between a sensor node and a PC, and mutual authentication between each pair of sensor nodes. We'd like to emphasize that a shared secret key between each sensor node and the PC is computed based on elliptic curve cryptography (ECC), while the authentication procedure is based upon hash chains. In addition, a group key of the sensor nodes is calculated only by the PC, since the PC is assumed to have no power and resource restriction.

The rest of this paper is organized as follows. In the next section, the related work is briefly discussed. A novel key management protocol in BANs is described in detail in Section III. Security analysis and performance analysis of our protocol are presented in Section IV and Section V, respectively. Finally, the conclusions of this paper are covered in Section VI.

ISBN: 978-1-61208-133-5

II. Related Work

In BANs, secure sensor association is a non-trivial issue, because a healthcare worker must check whether a group of sensors are correctly and securely associated with an intended patient before any data communication happens. Lots of previous works focus only on group key agreement in sensor nodes [6], [7], [8], [9].

Recently, Keoh et al. [10] and Li et al. [11] propose some protocols considering both sensor association and key agreement. In [10], each sensor node can be securely associated with the controller using public key based authentication. However, it does not take sensor-to-patient authentication into account. It is easily for malicious nodes to join the BAN to achieve the important medical data. In [11], group device pairing (GDP) is implemented to perform authentication and establish group keys. However, the computation and communication cost of GDP is very high. In particular, in order to achieve the group key, each sensor node needs $n + 3$ times modular exponentiation operations, where n is the total number of sensor nodes in the BAN. It is a large burden for the power and resource constrained medical sensor nodes.

In our protocol, we use ECC and hash chains to perform authentication and key generation. First, a patient and a healthcare worker authenticate each other. Secondly, the authenticated healthcare worker associates the medical sensor nodes with the intended patient. Each node can establish a shared secret key with the patient controller (PC), then the LED blinking pattern can be transmitted using the shared secret key. The healthcare worker can confirm the secure sensor association when all the sensor nodes have the synchronized LED blinking pattern. Thirdly, a group key is computed by the PC, which then distributes the group key to all the sensor nodes by utilizing the shared secret keys. Note here that key distribution based on symmetric key cryptography is fast and efficient. We'd like to emphasize that ECC and hash chains are very efficient methods in cryptography. In particular, a point multiplication in ECC is more efficient than a modular exponentiation in RSA [15] [16]. In addition, hash operation is a kind of lightweight cryptographic primitive. The use of ECC and hash chains can satisfy the requirement of the resource-limited medical sensors in BANs.

III. A Novel Key Management Protocol in BANs

In this section, we elaborate the novel key management protocol in BANs. We start with describing the protocol model and threat model briefly, and then present the design of the proposed protocol in detail. By the way, we need some notations in our protocol, which are showed in Table I.

A. Protocol and Threat Model

The protocol involves three entities: medical sensor, patient's controller (PC) and healthcare worker's device (HWD). In addition, we assume that the hospital is the key generation center (KGC) which is engaged as a trusted third party to issue important things to the patients and healthcare workers. First of all, the KGC chooses a prime number p and decides a elliptic curve E_p with order q over Z_p. Then, the KGC picks a random integer $s \in Z_p^*$ as its private key, and computes its public key $P_{pub} = s\mathcal{G}$, where $\mathcal{G}$ is the generator of the group of points over E_p. Note that the private key s of KGC should be changed periodically. At last, the KGC issues $\{p, E_p, q, P_{pub}\}$ to the patients and healthcare workers, but keeps s secret. It is worth noting that only the registered patients and healthcare workers can obtain these important materials. Our protocol works under the assumption that the medical sensor has a hash function, a random number generator and a re-writeable memory. Since a hash function is a powerful and computational efficient cryptographic tool, in the proposed protocol, we use the hash chains to perform the authentication in the BAN. In addition, a shared secret key between each sensor node and the PC is computed based on ECC, which is more efficient than RSA because the computation cost of a point multiplication is less than that of a modular exponentiation [15] [16]. Much of the details of ECC can be found in [12] [13] [14]. Furthermore, a group key of the group of medical sensor nodes associated with an intended patient needs to be calculated by the PC, which then distribute the group key to all the sensors.

Table I

Frequently used notations

p	A prime number
Z_p	A finite field
E_p	An elliptic curve over Z_p
$\mathcal{G}$	The generator the group of points over E_p
q	The order of the group of points over E_p
s	The private key of KGC
P_{pub}	The public key of KGC
n	The number of medical sensor nodes in the BAN
$h()$	One-way hash function
$h^z()$	z cascade hash operations
ID_c	The identity of a PC
ID_d	The identity of a HWD
N_x	The identity of a medical sensor node with index x
k_c	The secret key of a PC
k_d	The secret key of a HWD
k_x	The secret key of a medical sensor node with index x
$\mathcal{K}_G$	The group key of medical sensor node.
r_c	The random number generated by a PC
r_d	The random number generated by a HWD
r_x	The random number generated by a medical sensor node x

In this paper, first, we'd like to emphasize that only the hospital is trusted, which is considered as the trusted third party KGC. In [10] [11], the authors assume that the patients and healthcare workers are all well-behaved.

However, in fact, some patients and healthcare workers may not be always trusted. For instance, an off-duty doctor or a discharged patient can perform some malicious attacks to obtain the secret keys by eavesdropping, to impersonate as a legitimate group member to join the group, or to modify the information communicated between legitimate group members so as to disrupt key authentication. In our protocol, we consider the situations mentioned above in order to deal with the malicious insiders. Second, we assume that the medical sensors can be attacked by passive attacks and active attacks. In BANs, the medical sensor nodes can be placed in, on, or around a patient's body, which are capable of sensing, storing, processing and transmitting data via wireless communications. As we know, wireless channels are susceptible to passive eavesdropping and message interception. Hence, adversaries can easily perform passive attacks. On the other hand, in active attacks, an adversary not only just records the data, but also can alter, inject, intercept and replay messages. Note that the sensor nodes do not trust each other before association and can be compromised after deployment. At last, the attackers are assumed to be able to eavesdrop on the wireless communication channel and intercept, modify, replay or inject the transmitting data.

B. Design of the protocol

The proposed protocol can be divided into three phases: initialization phase, secure sensor association phase, and key management phase. The detailed procedures are described as follows.

1) Initialization Phase: Suppose that there are n medical sensor nodes with identities $\{N_1, N_2, \cdots, N_n\}$ in a BAN. First of all, the registered PC with the identity ID_c and HWD with the identity ID_d obtain $\{p, E_p, q, P_{pub}\}$ from KGC. Then, the PC generates its own secret key k_c and a random number r_c. Similarly, the HWD derives k_d and r_d.

2) Secure Sensor Association Phase: In this phase, first, the registered PC and HWD authenticate each other so as to withstand the attacks from a malicious off-duty doctor or a malicious discharged patient. Then, a group of medical sensor nodes are securely and correctly associated to the authenticated patient.

(1) The PC and HWD need to authenticate each other before any data communication happens. Seen from Fig. 1, the PC and HWD both generate a random number $\{r_c, r_d\}$ and calculate $\{S_c = k_c P_{pub}, S_d = k_d P_{pub}\}$. After that, they compute hash values $\{A_c, A_d\}$ and exchange the messages $\{S_c, r_c, ID_c\}$ and $\{S_d, r_d, ID_d\}$. In order to authenticate each other, the PC and HWD need to check whether the equations $A_c = h(S_c \parallel r_c \parallel ID_c)$ and $A_d = h(S_d \parallel r_d \parallel ID_d)$ can hold.

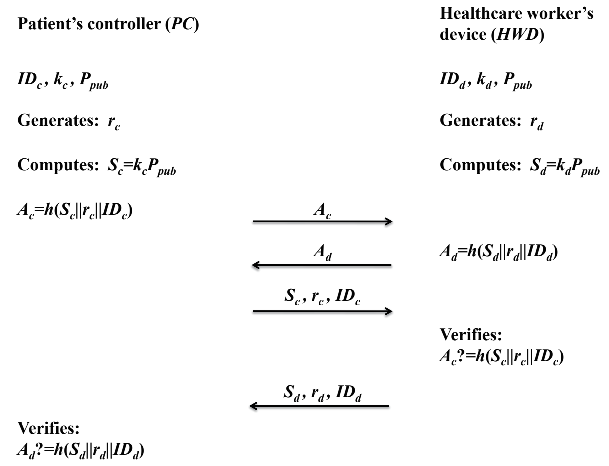
Patient's controller (PC)
Healthcare worker's device (HWD)
ID_c, k_c, P_{pub}
ID_d, k_d, P_{pub}
Generates: r_c
Generates: r_d
Computes: S_c=k_cP_{pub}
Computes: S_d=k_dP_{pub}
A_c=$h(S_c||r_c||ID_c)$
A_c
A_d
A_d=$h(S_d||r_d||ID_d)$
S_c, r_c, ID_c
Verifies:
A_c?=$h(S_c||r_c||ID_c)$
S_d, r_d, ID_d
Verifies:
A_d?=$h(S_d||r_d||ID_d)$

Figure 1. Mutual authentication between PC and HWD.

(2) After the PC and HWD authenticate each other, a group of sensor node must be securely and correctly associated with the patient. The authenticated PC first generates n secret keys $\{k_1, k_2, \cdots, k_n\}$ and n random numbers $\{r_1, r_2, \cdots, r_n\}$, and then preloads each secret key k_x and each random number r_x to node N_x, for $x = 1, 2, \cdots, n$. Next, the PC computes its own hash chain $h^z(k_c \parallel r_c)$ as well as the hash chain $h^z(k_x \parallel r_x)$ of node N_x, for $x = 1, 2, \cdots, n$. After that, the PC broadcasts all the information $h^z(k_x \parallel r_x)$ to the group of nodes. Note that z is a large constant number and $h^z(m)$ denotes the application of z cascade hash operations starting from m. For instance, $h^2(m) = h(h(m))$, $h^3(m) = h^2(h(m)) = h\left(h^2(m)\right) = h(h(h(m)))$, etc.. At last, the PC publishes $\{p, E_p, q, P_{pub}\}$. In our protocol, we assume that the broadcasting hash chain for node N_x will be updated after each successful authentication. The hash chain $h^z(k_x \parallel r_x)$ of node N_x will be replaced with $h^{z-l}(k_x \parallel r_x)$ when the node N_x have passed through authentication l times. Now, it is supposed that node N_i and PC have passed through authentication u times and v times, respectively. Then the broadcasting hash chains for node N_i and PC are $h^{z-u}(k_i \parallel r_i)$ and $h^{z-v}(k_c \parallel r_c)$, respectively. The processes of authentication and key establishment between node N_i and PC are divided into five steps, which are shown in Fig. 2.

(2-1) The node N_i generates a random number t_i as its secret key and computes the point $A_i = t_i P_{pub} = (x_i, y_i)$ over the elliptic curve E_p and $S_i = h\left(x_i \parallel h^{z-u-1}(k_i \parallel r_i)\right)$, then it sends a message $\{N_i, A_i, S_i\}$ to the PC. Similarly, the PC generates a random number t_c and computes $A_c = t_c P_{pub} = (x_c, y_c)$ and $S_c = h\left(x_c \parallel h^{z-v-1}(k_c \parallel r_c)\right)$, then it sends a message $\{N_c, A_c, S_c\}$ to the node N_i. Note here that x_i and x_c are the x-component of points A_i and A_c, respectively. For security, the secret t_i and t_c cannot be

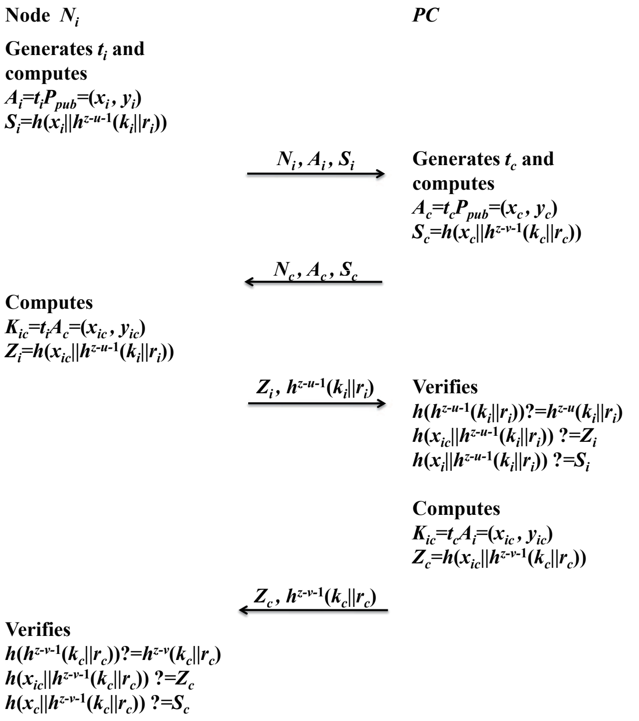
Node N_i
PC
Generates t_i and computes
$A_i = t_i P_{pub} = (x_i, y_i)$
$S_i = h(x_i \| h^{z-u-1}(k_i \| r_i))$
N_i, A_i, S_i
Generates t_c and computes
$A_c = t_c P_{pub} = (x_c, y_c)$
$S_c = h(x_c \| h^{z-v-1}(k_c \| r_c))$
N_c, A_c, S_c
Computes
$K_{ic} = t_i A_c = (x_{ic}, y_{ic})$
$Z_i = h(x_{ic} \| h^{z-u-1}(k_i \| r_i))$
$Z_i, h^{z-u-1}(k_i \| r_i)$
Verifies
$h(h^{z-u-1}(k_i \| r_i)) ?= h^{z-u}(k_i \| r_i)$
$h(x_{ic} \| h^{z-u-1}(k_i \| r_i)) ?= Z_i$
$h(x_i \| h^{z-u-1}(k_i \| r_i)) ?= S_i$
Computes
$K_{ic} = t_c A_i = (x_{ic}, y_{ic})$
$Z_c = h(x_{ic} \| h^{z-v-1}(k_c \| r_c))$
$Z_c, h^{z-v-1}(k_c \| r_c)$
Verifies
$h(h^{z-v-1}(k_c \| r_c)) ?= h^{z-v}(k_c \| r_c)$
$h(x_{ic} \| h^{z-v-1}(k_c \| r_c)) ?= Z_c$
$h(x_c \| h^{z-v-1}(k_c \| r_c)) ?= S_c$

Figure 2. Mutual authentication and key establishment between node N_i and PC.

reused.

(2-2) After receiving the message $\{N_c, A_c, S_c\}$, the node N_i computes a shared secret key $K_{ic} = t_i A_c = t_i t_c P_{pub} = (x_{ic}, y_{ic})$ and $Z_i = h\left(x_{ic} \parallel h^{z-u-1}\left(k_i \parallel r_i\right)\right)$. Then, it delivers a message $\left\{Z_i, h^{z-u-1}\left(k_i \parallel r_i\right)\right\}$ to the PC.

(2-3) Upon receiving the message $\left\{Z_i, h^{z-u-1}\left(k_i \parallel r_i\right)\right\}$ and the previous one $\{N_i, A_i, S_i\}$ from (2-1), the PC checks whether the conditions $h\left(h^{z-u-1}\left(k_i \parallel r_i\right)\right) = h^{z-u}\left(k_i \parallel r_i\right)$, $h\left(x_{ic} \parallel h^{z-u-1}\left(k_i \parallel r_i\right)\right) = Z_i$ and $h\left(x_i \parallel h^{z-u-1}\left(k_i \parallel r_i\right)\right) = S_i$ are satisfied. If they are satisfied, the PC can make sure that the node N_i is a authorized one and subsequently computes a shared secret key $K_{ic} = t_c A_i = t_c t_i P_{pub} = (x_{ic}, y_{ic})$. Next, the PC computes $Z_c = h\left(x_{ic} \parallel h^{z-v-1}\left(k_c \parallel r_c\right)\right)$ and then sends a message $\left\{Z_c, h^{z-v-1}\left(k_c \parallel r_c\right)\right\}$ to the node N_i. If the conditions mentioned above are not satisfied, the authentication fails and the PC beeps. Note that the shared secret key between node N_i and PC are definitely same due to $t_i A_c = t_c A_i = t_i t_c P_{pub}$.

(2-4) After receiving $\left\{Z_c, h^{z-v-1}\left(k_c \parallel r_c\right)\right\}$ from the PC, the node N_i checks whether the conditions $h\left(h^{z-v-1}\left(k_c \parallel r_c\right)\right) = h^{z-v}\left(k_c \parallel r_c\right)$, $h\left(x_{ic} \parallel h^{z-v-1}\left(k_c \parallel r_c\right)\right) = Z_c$ and $h\left(x_c \parallel h^{z-v-1}\left(k_c \parallel r_c\right)\right) = S_c$ are satisfied. If they are satisfied, the node N_i can verify the authenticity of the PC. Otherwise, the authentication fails and the node N_i beeps.

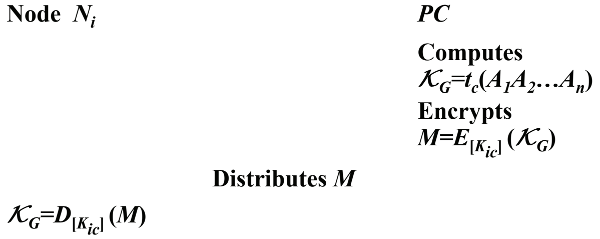
Node N_i
PC
Computes
$\mathcal{K}_G = t_c(A_1 A_2 \ldots A_n)$
Encrypts
$M = E_{[K_{ic}]}(\mathcal{K}_G)$
Distributes M
$\mathcal{K}_G = D_{[K_{ic}]}(M)$

Figure 3. Group key derivation.

(2-5) Finally, node N_i and PC update their broadcasting hash chains to be $h^{z-u-1}\left(k_i \parallel r_i\right)$ and $h^{z-v-1}\left(k_c \parallel r_c\right)$, respectively.

Now, the PC and the node N_i authenticate each other and establish a shared secret key K_{ic}, which can be used to encrypt and transmit the LED blinking pattern. The healthcare worker indicates “authentication accepted” to the controller if the LED blinking patterns of the sensor nodes are same. Furthermore, our scheme can also provide mutual authentication between nodes. The authentication for each pair of nodes is same as the authentication for the node N_i and the PC.

3) Key Management Phase: In this phase, a group key of participating medical sensor nodes is derived. The PC in the BAN is responsible for the key distribution and management since it typically has higher computation capability and storage capability. Observed from the above secure sensor association phase, the PC can receive n points $\{A_1, A_2, \cdots, A_n\}$ from nodes $\{N_1, N_2, \cdots, N_n\}$. Then, it calculates the group key $\mathcal{K}_G = t_c \cdot (A_1 A_2 \cdots A_n)$, which can be subsequently distributed to each sensor node N_i by using the corresponding shared secret key K_{ic}. Observed from Fig. 3, the group key distribution is based on symmetric key cryptography. The PC can distribute the group key to all the nodes in the BAN. For instance, if the PC wants to distribute $\mathcal{K}_G$ to the node N_i, it just needs to encrypt $\mathcal{K}_G$ using the shared secret key K_{ic}. After receiving M, the node N_i can easily derive the group key by decrypting M using the same shared secret key K_{ic}.

IV. Security Analysis

In this section, we show security of the presented protocol in withstanding the attacks of passive and active adversaries.

A. Security Against Passive Adversary

A passive adversary (attacker) tries to learn information about the secret key by eavesdropping on the broadcast channel. In our protocol, an eavesdropper cannot get any information about the secret value t_i due to discrete logarithm problem in elliptic curves. Therefore, the secret value

t_i of each node N_i can be protected and the attacker is not able to learn the information of the group key.

B. Security Against Active Adversary

In active attack, an adversary not only just records the data, but also can alter, inject, intercept and replay messages. The goal of the authentication mechanism is to convince a controller that the nodes he is communicating with are indeed the nodes they claim to be. We show the analysis of the concrete security properties withstanding the active attacks that we concerned in our proposed protocol as follows:

1) Malicious Insiders Resistance: Malicious insiders mean that the misbehaved off-duty doctors or the misbehaved discharged patients. In our protocol, the KGC can issue p, E_p, q, P_{pub} to the registered doctors or patients. Note that $P_{pub} = s\mathcal{G}$, where the private key s of KGC will be updated periodically. For example, if a doctor is off-duty or a patient is discharged from the hospital, then the private key s of the KGC must be changed. Then all the subsequent authentications will fail, since the P_{pub} is changed. Hence, the malicious insiders can not perform attacks without being noticed.

2) Implicit Key Authentication: Implicit key authentication is a fundamental security property, which implies that only the users with whom A wants to agree upon a common key may be able to compute a key. In our protocol, the sensor nodes agreeing upon a group key are controlled by the PC. It is clear that our protocol provides implicit key authentication.

3) Known Session Key Security: Known session key security indicates that an adversary having obtained some previous session keys still cannot deduce the session keys of the current run of the protocol. In our protocol, each node N_i selects a random number t_i as secret for each session and calculates $A_i = t_i P_{pub}$. It is impossible for the adversary to derive certain secret key t_i so as to obtain the current shared secret key K_{ic} or the current group key $\mathcal{K}_G$.

4) Key-Compromise Impersonation Resistance: Key-compromise impersonation security ensures that the compromise of one user's long-term private key cannot expose the other user's long-term private key. In our protocol, each user's long-term private key k_i is individually generated by the PC. Therefore, the adversary having obtained a certain user's long-term private key cannot expose the long-term private key of other user's.

V. Performance Analysis

In this section, we will compare the performance of the proposed protocol with the protocol presented by Li et al [11]. Our protocol uses ECC and hash chains to perform authentication and key generation because ECC and hash chains are very efficient methods in cryptography. In our protocol, the most expensive operation is the point multiplication, while in Li et al.'s protocol the most expensive operation is modular exponentiation. It has been shown in [15] [16] that a point multiplication needs less computation time than a modular exponentiation unless the exponent is chosen as some specific value.

Table II
Performance comparison of Li's protocol and our protocol

	Li et al.'s protocol	Our protocol
Computations for each node to achieve authentication and compute a shared secret key	$(3+n)\,e + 9h^{(1)}$	$2p + 5h^{(2)}$
Total number of transmissions for the protocol	20	10

(1) n is the number of medical sensor nodes in the BAN, e is the modular exponentiation operation, and h is the hash operation.
(2) p is the point multiplication operation over elliptic curve.

We summarize the performance comparison of the proposed protocol with Li et al.'s in Table II. As shown in Table II, in Li et al.'s protocol [11], each node needs to perform $3+n$ times modular exponentiation operations and 9 times hash operations. Moreover, it is required to send 20 transmissions in their protocol to run the authentication and key generation. However, in the proposed protocol, each sensor node needs to perform only two point multiplications over an elliptic curve ($A_i = t_i P$ and $K_{ic} = t_i A_c$) and five hash operations. In addition, our protocol only needs 10 transmissions. Therefore, the proposed protocol is more efficient than Li et al.'s protocol. It significantly reduces the overhead of communication for sensor node to achieve secure connectivity. We'd like to emphasize that we do not compare the performance of our protocol with Keoh et al.'s protocol since Keoh et al.'s protocol uses public key cryptography to execute the authentication which is different from symmetric key cryptography used in our protocol.

VI. Conclusion

The BAN system provides a flexible, wearable infrastructure for acquisition, processing and wireless transmission of medical data and information at the human body. Using BAN, multiple vital signs can be wirelessly monitored even in mobile environments.

In this paper, we propose a novel enhanced secure sensor association and key management protocol based on elliptic curve cryptography (ECC) and hash chains in order to provide secure and correct association of a group of sensors with a patient and satisfy the requirements of data confidentiality and integrity in BANs. The authentication procedure and group key generation are very simple and efficient. Therefore, our protocol can be easily implemented into the power and resource constrained sensor nodes in BANs. Compared with the previous works in BANs, our protocol needs less computation and communication cost for the authentication and key derivation. Meanwhile, our protocol can provide mutual authentication between PC and HWD, mutual authentication between PC and nodes, and mutual authentication between nodes. We believe that our protocol is attractive to the applications of BANs.

REFERENCES

[1] M. Patel and J. Wang, "Applications, Challenges, and Prospective in Emerging Body Area Networking Technologies," *IEEE Wireless Communications*, pp. 80-88, Feb. 2010.

[2] K. Lorincz, D. Malan, T. F. Jones, A. Clavel, V. Shnayder, G. Mainland, M. Welsh, and S. Moulton, "Sensor Networks for Emergency Response: Challenges and Opportunities," *IEEE Pervasive Computing*, vol.3, no. 4, pp. 16-23, Oct.-Dec. 2004.

[3] M. Hanson, H. Powell, A. Barth, K. Ringgenberg, B. Calhoun, J. Aylor, J. Lach, "Body Area Sensor Networks: Challenges and Opportunities," *Computer*, vol.42, no. 1, pp. 58-65, Jan. 2009.

[4] M. Li, W. Lou, and K. Ren, "Data Security and Privacy in Wireless Body Area Networks," *IEEE Wireless Communications*, vol. 17, no. 1, pp. 51-58, Feb. 2010.

[5] E. Jovanov, A. Milenkovic, C. Otto, and P. C. Groen, "A Wireless Body Area Network of Intelligent Motion Sensors for Computer Assisted Physical Rehabilitation," *Journal of NeuroEngineering and Rehabilitation*, vol. 2, no. 1, pp. 1-10, Mar. 2005.

[6] C. C. Tan, H. Wang, S. Zhong, and Q. Li, "IBE-Lite: A Lightweight Identity-Based Cryptography for Body Sensor Networks," *IEEE Trans. on Information Technology in Biomedicine*, vol. 13, no. 6, pp. 926-932, Nov. 2009.

[7] O. G. Morchon, H. Baldus, and D. S. Sanchez, "Resource-Efficient Security for Medical Body Sensor Networks," in *BSN'06*, pp. 80-83, Apr. 2006.

[8] T. Donovan, J. Donoghue, C. Sreenan, D. Sammon, P. Reilly, and K. A. Connor, "A Context Aware Wireless Body Area Network (BAN)," in *proc. of the Pervasive Health Conference*, pp. 1-8, Apr. 2009.

[9] K. Malasri and L. Wang, "Addressing Security in Medical Sensor Networks," in *HealthNet'07*, pp. 7-12, 2007.

[10] S. L. Keoh, E. Lupu, and M. Sloman, "Securing Body Sensor Networks: Sensor Association and Key Management," *proc. of the 7th Annual IEEE Int. Conference on Pervasive Computing and Communications (PerCom)*, Galveston, Texas, Mar. 9-13, pp. 1-6, 2009.

[11] M. Li, S. Yu, W. Lou, and K. Ren, "Group Device Pairing based Secure Sensor Association and Key Management for Body Area Networks," *proc. of IEEE INFOCOM*, San Diego, CA, Mar. 14-19, pp. 1-9, 2010.

[12] N. Koblitz, "Elliptic curve cryptosystems," *Math. Comput.*, vol. 48, pp. 203-209, 1987.

[13] H. Silverman, *The Arithmetic of Elliptic Curves*, 2nd ed. Springer, 2000.

[14] W. Stallings, *Cryptography and Network Security*, 4th ed. Prentice Hall, 2005, pp. 301-313.

[15] R. Watro, D. Kong, S. Cuti, C. Gardiner, C. Lynn, and P. Kruus, "TinyPK: Securing Sensor Networks with Public Key Technology," in *Proc. of the 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'04)*, Washington, DC, USA, pp. 59-64, Oct. 2004.

[16] D. J. Malan, M. Welsh, and M. D. Smith, "A Public-Key Infrastructure for Key Distribution in TinyOS Based on Elliptic Curve Cryptography," in *1st IEEE International Conference on Sensor and Ad Hoc Communications and Networks (SECON'04)*, Santa Clara, California, pp. 71-80, Oct. 2004.