Securing Communications Between External Users and Wireless Body Area Networks

Chunqiang Hu[1,2], Fan Zhang[1], Xiuzhen Cheng[1]
1. Dept. of Computer Science, The George Washington U. Washington DC, DC, USA
{chu, zfwise, cheng}@gwu.edu

Xiaofeng Liao[2]
2. College of Computer Science Chongqing U. Chongqing, China
xfliao@cqu.edu.cn

Dechang Chen [3]
3. Division of Epidemiology and Biostatistics Uniformed Services University of the Health Sciences, MD, USA
{dechang.chen}@usuhs.edu

ABSTRACT

Wireless Body Area Networks (BANs) are expected to play a crucial role in patient-health monitoring in the near future. Establishing secure communications between BAN sensors and external users is key to addressing the prevalent security and privacy concerns. In this paper, we propose the primitive functions to implement a secret-sharing based Ciphertext-Policy Attribute-Based Encryption (CP_ABE) scheme, which encrypts the data based on an access structure specified by the data source. We also design two protocols to securely retrieve the sensitive patient data from a BAN and instruct the sensors in a BAN. Our analysis indicates that the proposed scheme is feasible, can provide message authenticity, and can counter possible major attacks such as collusion attacks and battery-draining attacks.

Categories and Subject Descriptors

C.2 [**COMPUTER-COMMUNICATION NETWORKS**]: General—*Data communications; Security and protection*; D.4.6 [**Security and Protection**]: [Access controls; Cryptographic controls; Authentication]; J. [**Computer Applications**]: J.3LIFE AND MEDICAL SCIENCES[Medical information systems; Health]

General Terms

Algorithms, Design, Security, Verification

Keywords

Wireless body area networks; access control tree; bilinear map; secure communications; attribute-based cryptosystem.

1. INTRODUCTION

In recent years, innovative health-oriented networking and wireless communication technologies have been developed and they have become an intrinsic part of many modern medical services. Body Area Networking (BAN) as a key enabling technique for E-healthcare systems makes real-time health-related information accessible to medical specialists, who are then enabled to cast appropriate and timely medical treatment to the patients.

HotWiSec'13, April 19, 2013, Budapest, Hungary.

Unlike conventional sensor networks, a BAN deals with more sensitive and important patient information that has significant security, privacy, and safety concerns, which may prevent the wide adoption of this technology. As a sensor that collects patient information, all it cares is to distribute the information to authorized doctors and other experts securely. However, this is a very challenging problem [4,9]. First, data should be transmitted in a secure channel; but securing wireless communications is not trivial. Second, node authentication is the most fundamental step towards a BAN's initial trust establishment, key generation and rekeying, and subsequent secure communications. Third, the high computational cost of asymmetric cryptography leaves symmetric encryption the only viable option; but key-distribution in symmetric encryption is challenging. Fourth, due to the limitation of memory spaces in sensors, a data sink, which has considerably larger memory, is employed to store the data. To ensure the security of the data, we need to have certain level of protection to the data sink. However, a smartphone-like device serving as the data sink can be physically lost or stolen, and an attacker can read the data once the device is captured. Additionally, recent research disclosed that smartphones suffer from severe privacy concerns since many applications often cross the line and read sensitive data at their free will (for example, many smart phone apps read the user's location information).

To overcome the challenges mentioned above, we resort to the so-called Ciphertext-Policy Attribute-Based Encryption (CP_ABE) [2], which was proposed as a new means of providing role-based access control on encrypted data. ABE has been exploited to secure the communications between a BAN and its external users in [6] and [1], with [6] focusing on securing the communications between the data controller and an external user via fuzzy ABE and [1] addressing self-protecting electronic medical records (EMRs) on mobile devices and offline communications using the basic ABE. In this paper, we employ CP_ABE such that a sensor can control the access to its data by constructing an access structure defined by a set of attributes. Data are then stored in ciphertext format at the data sink and the trust we put on the data sink is thus drastically decreased as the data sink does not have the key to decrypt the stored ciphertext. Since CP_ABE belongs to the asymmetric encryption family, which implies a high computational cost, we propose to utilize CP_ABE to encrypt a data encryption key, based on which the data is encrypted by symmetric encryption.

More specifically, we design four primitive algorithms to implement a secret-sharing based CP_ABE and propose two protocols to securely retrieve the sensitive patient data from a BAN and instructs the sensors in a BAN. We also analyze the security strength of our scheme and prove its correctness. The rest of the paper is organized as follows. We present the preliminaries and the system model in Section 2, and develop the main idea of the communication protocols in Section 3. Section 4 analyzes the security of the proposed protocols, followed by a conclusion in Section 5.

2. PRELIMINARIES AND SYSTEM MODEL

2.1 Preliminaries

We now introduce some preliminary knowledge regarding the cryptographic primitives used in this paper.

2.1.1 Bilinear Maps

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two bilinear groups of prime order p, and g be a generator of $\mathbb{G}_1$. Our proposed scheme makes use of a bilinear map: $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ with the following properties:

1. *Bilinear:* A map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ is bilinear if and only if for all $P, Q \in \mathbb{G}_1$ and all $a, b \in \mathbb{Z}_p$, we have $e(P^a, Q^b) = e(P, Q)^{ab}$. Here $\mathbb{Z}_p = \{0, 1, \ldots, p-1\}$ is the Galois field of order p.
2. *Non-degeneracy:* The generator g satisfies $e(g, g) \neq 1$.
3. *Computability:* There exists an efficient algorithm to compute $e(P, Q)$ for $\forall P, Q \in \mathbb{G}_1$.

Note that the hardness [7] of the decision version of Bilinear Map - i.e., the decisional bilinear Diffie-Hellman problem (DBDH) - forms the basis for the security of our scheme.

2.1.2 Secret Sharing Schemes

Another important cryptographic primitive used by our scheme is secret sharing. Secret sharing schemes were first developed by Shamir [8] and then extensively studied by different researchers [3, 5]. We provide a brief overview as follows: In the context of a *dealer* sharing a secret with a number of *participants* $u_1, \ldots, u_n$, a participant learns the secret if and only if it can cooperate with at least $t-1$ other participants (on sharing what they learn from the dealer), where $t \leq n$ is a pre-determined parameter. The secret to be shared by the dealer is denoted by $s \in \mathbb{Z}_p$, where $p > n$. Before secret sharing, each respondent (participant) u_i should obtain its secret key $x_i \in \mathbb{Z}_p$, which is only known by u_i and the dealer.

The dealer follows a two-step process. First, it constructs a polynomial function $f(x)$ of degree $t-1$, i.e.

$$f(x) = s + \sum_{j=1}^{t-1} a_j x^j, \tag{1}$$

by randomly choosing each a_j i.i.d. with a uniform distribution from $\mathbb{Z}_p$. Note that all (additive and multiplication) operations used in (1) and throughout the rest of the paper are modular arithmetic (defined over $\mathbb{Z}_p$) as opposed to real arithmetic. Also note that s forms the constant component of $f(x)$ - i.e., $s = f(0)$. Then, in the second step, the dealer transmits to each u_i a shared secret s_i, where

$$s_i = f(x_i). \tag{2}$$

We now show how t or more users can cooperate to recover the s by sharing the secret shares received from the dealer. Without loss of generality, let $u_1, \ldots, u_t$ be the t cooperating users. These t users can reconstruct the secret $s = f(0)$ from $s_1 = f(x_1), \ldots, s_t = f(x_t)$ by computing

$$s = f(0) = \sum_{j=1}^{t} \left(s_j \prod_{i \in [1,n], i \neq j} \frac{0 - x_j}{x_i - x_j} \right). \tag{3}$$

2.2 System Model

In this paper, we consider a BAN communication system depicted in Fig. 1. There are four major entities in this system: Key Generation Center (KGC), Sensor (implanted and wearable devices), Data Sink (a dedicated BAN data controller or a mobile device such as a smart phone), and Data Consumer (doctors and nurses), whose major functions are summarized in the following four subsections.

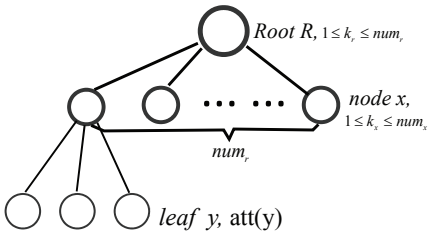

Figure 2: An access control tree structure in a BAN

2.2.1 Key Generation Center (KGC)

The KGC is used to perform system initialization, generate public parameters, and compute a secret key for each data consumer and each sensor based on their attributes. The public parameters should be installed into the sensors before they are deployed (attached to or implanted in a human body) in a BAN. A data consumer should be able to prove to the KGC that it is the owner of a set of attributes.

2.2.2 Sensor

A BAN consists of wireless sensors called BAN devices either embedded on/near the surface (i.e., wearable devices) or implanted in the deep tissue (i.e., implanted devices) of a human body. The BAN devices should have certain computational capability to encrypt the patient's data and store the ciphertext into the data sink. When a data consumer needs the data, he should communicate with the data sink to retrieve the (encrypted) data.

2.2.3 Data Sink

A data sink, which could be the BAN controller or a mobile device such as a smartphone, is used to store the patient's data. We apply the attribute-based encryption proposed by Bethencourt, Sahai, and Waters [2] to encrypt the data and store the ciphertext in the data sink according to the requirements of the BAN.

2.2.4 Data Consumer

Data consumers refer to the doctors and nurses or other experts. To decrypt a message, data consumers should have the attributes that satisfy the access tree specified by the data source (the sensor generating the data).

2.3 Access Control Policy – the Access Tree

Our main idea is to design an attribute-based security scheme that views an identity as a set of attributes, and enforces a lower bound on the number of common attributes between a user's identity and its access rights specified for the sensitive data. We use an access tree to control the data consumers' access to the encrypted data. Fig. 2 illustrates such an access tree structure. In Fig. 2, num_x is the number of child nodes of node x, and $k_x \in [1, num_x]$ is its threshold value, with $k_x = n$ indicating that node x performs the *OR* operation over all the subsets of n child nodes of x, with each subset supporting an *AND* operation. Each leaf node y is described by an attribute att(y) and a threshold value $k_y = 1$. When a data item is generated, its associated attributes defining the access right are used to create a tree for access control, which implies that only the users possessing a certain number of the attributes of the data item can decrypt the encrypted data.

3. THE PROPOSED SECURE DATA COMMUNICATION PROTOCOLS

In this section, we propose the data communication protocols to secure the messages when a data consumer, which could be a doctor or other expert, communicates with the sensors or the data sink, to distribute instructions and commands to the BAN, or retrieve the sensitive data from the BAN.

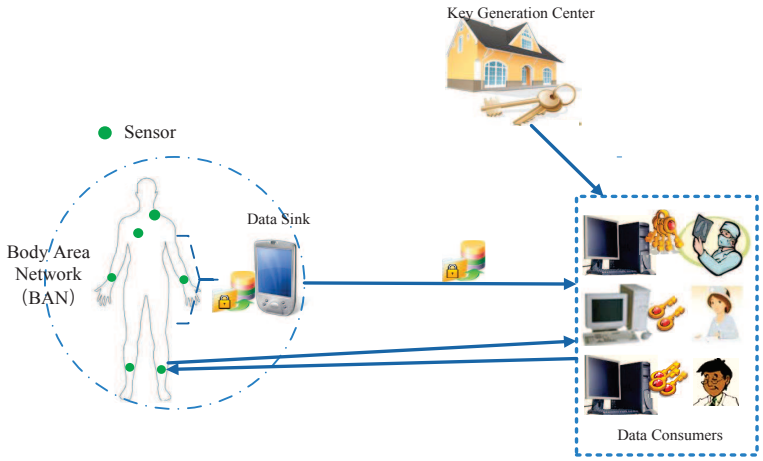

Figure 1: A BAN architecture of a health care application

3.1 Primitive Algorithms

We first introduce the following four primitive algorithms that will be utilized by the communication protocols between the data consumers and a BAN. **Algorithm** 1 presents the system initialization performed by *KGC*. **Algorithm** 2 is executed by *KGC* to generate private keys for the sensors and data consumers based on their attributes. The encryption procedure is detailed in **Algorithm** 3, which encrypts a data encryption key K or an access token K_1 with an access tree T, all specified by the sensor. **Algorithm** 4 implements decryption and authentication, which should be executed by data consumers since they receive only encrypted data from the BAN.

3.2 System Initialization

Before a BAN is deployed, the following system initialization procedure needs to be carried out:

1. The KGC computes the public parameter PK according to **Algorithm** 1, and posts PK to all sensors and data consumers.

2. The KGC computes a private key for each sensor and each data consumer based on their possessed attributes according to **Algorithm** 2.

Note that the set of attributes possessed by a sensor defines the access right to the data collected by the sensor, and the access right to the sensor itself. The access control policy is presented in Section 2.3. In the following two subsections we will present two protocols to illustrate the procedures for a data consumer to retrieve from a BAN the sensitive patient data and to send commands to the BAN sensors.

3.3 Retrieving Data From A BAN

The following procedure illustrates how a data consumer with a set of private keys (corresponding to its attribute set S) obtains the sensitive patient data from the data sink.

1. The sensor selects a random data encryption key K, encrypts K using **Algorithm** 3, and then encrypts its data M by AES: $AES(K, M)$. One can see that we do not include AES in **Algorithm** 3 since we use **Algorithm** 3 to encrypt the data encryption key K, which is used to perform symmetric encryption for the patient data.

2. The sensor sends the encrypted data (the ciphertexts of K and M) to the data sink, where ID_s and ID_d are respectively the identity of the sensor and the data sink.

$$Sensor \rightarrow Data\ Sink: \\ (ID_s, ID_d, Encryption(PK, K, T), AES(K, M)) \quad (8)$$

3. The data consumer obtains the encrypted data from the data sink, and then executes **Algorithm** 4 to retrieve the data encryption key K.

4. The data consumer decrypts $AES(K, M)$ using the data encryption key K.

Note that this procedure implies that a data encryption key K is needed for each data item. In practice, the data generated by the same sensor during a certain time interval can be encrypted by the same data encryption key K to conserve the sensor's computational

Algorithm 1 System Initialization

1: Selects a prime p, a generator g of $\mathbb{G}_0$, and a bilinear map $e : \mathbb{G}_0 \times \mathbb{G}_0 \rightarrow \mathbb{G}_1$.
2: Defines a Lagrange coefficient $\triangle_{i,S}$ for $i \in \mathbb{Z}_p$ and a set S of elements in $\mathbb{Z}_p$: $\triangle_{i,S} = \prod_{j \in S, j \neq i} \frac{x-j}{i-j}$.
3: Chooses two random exponents $\alpha, \beta \in \mathbb{Z}_p$.
4: Selects a hash function $H : \{0,1\}^* \rightarrow \mathbb{G}_0$. The function H is viewed as a random oracle.
5: Distributes the following public parameters to all sensors and data consumers:

$$PK = \mathbb{G}_0, g, h = g^{\beta}, e(g,g)^{\alpha} \quad (4)$$

6: Computes the master key *MSK*: (β, g^{α}).

Algorithm 2 Key Generation (*MSK*, S)

Inputs: The master key *MSK* and the set of attributes S possessed by an entity (could be a sensor or a data consumer) requesting a private key.

1: The *KGC* selects random numbers r and $r_j \in \mathbb{Z}_p$ for each attribute $j \in S$.
2: The private key SK is computed by

$$SK = (D = g^{\frac{(\alpha + r)}{\beta}}, \\ \{(D_j = g^r \cdot H(j)^{r_j}, D'_j = g^{r_j}) \mid \forall j \in S\}) \quad (5)$$

Algorithm 3 Encryption*(PK, K, T)*

Inputs: Public parameters PK; data encryption key K; and the tree T rooted at node R specifying the access right of the key K.

1: Selects a polynomial q_x and sets its degree $d_x = k_x - 1$ for each node x in the tree T.
2: Selects a random $s \in \mathbb{Z}_p$ and sets $q_R(0) = s$;
3: Selects d_R random points from $\mathbb{Z}_p$ to completely define the polynomial q_R.
▷ Traverse the tree in preorder and compute a polynomial for each tree node.
4: **for** any node x in T in preorder **do**
5: Sets $q_x(0) = q_{parent(x)}(index(x))$.
6: Selects d_x random points from $\mathbb{Z}_p$ to completely define q_x.
7: **end for**
8: Let Y be the set of leaf nodes in T. The ciphertext CK is constructed based on the access tree T as follows:

$$\begin{aligned} CK &= (T, \tilde{C} = Ke(g,g)^{\alpha s}, C = h^s, \\ &\quad \{(C_y = g^{q_y(0)}, C'_y = H(att(y))^{q_y(0)}) \mid \forall y \in Y\}, \\ &\quad e(H(K|e(g,g)^{\alpha s}), g)). \end{aligned} \tag{6}$$

power and energy. In such a case, K serves as a session key to encrypt all the data for a particular session.

3.4 Sending Instructions To A BAN

When a data consumer wants to send instructions or commands to a sensor in a BAN, a direct communication session between the data consumer and the sensor is needed. This procedure requires an access token specified by the sensor to grant the access right to the data consumer possessing certain attributes. It involves two phases, described in the following two subsections.

3.4.1 Communication Establishment Phase

1. First, the sensor selects an access token K_1, and encrypts $K_1||date$ with **Algorithm** 3. Then the sensor sends the encrypted token together with its hash $h = H(K_1||date)$ to the data sink:

$$Sensor \rightarrow Data\ Sink :$$
$$(ID_s, Encryption(PK, K_1||date, T), H(K_1||date)).$$

   The data sink stores the encrypted token and sends it to the data consumers when requested.

2. The sensor updates $K_1||date$ and the corresponding ciphertext at the data sink at a certain time interval, for example, once per day.

3. When a data consumer obtains the encrypted access token $Encryption(PK, K_1||date, T)$, he decrypts the ciphertext according to **Algorithm** 4. Then the data consumer sends a salted hash of the access token to the sensor to convince the sensor that he has the required privilege: $Data\ Consumer \rightarrow Sensor : h' = H(K_1||date)$.

4. The sensor receives the proof and then verifies whether or not $h' = h$. If succeeds, the sensor generates a new access token K'_1, and encrypts $K'_1||date$ using the same access tree with **Algorithm** 3. The sensor sends the encrypted $K'_1||date$ and $h = H(K'_1||date)$ to the data consumer as a challenge, and then to the data sink to overwrite the previously encrypted access token.

$$Sensor \rightarrow Data\ Consumer \quad and \quad Data\ sink :$$
$$(ID_s, Encryption(PK, K'_1||date, T), H(K'_1||date)) \tag{9}$$

Algorithm 4 Decryption (*CT*, *SK*)

Inputs: A ciphertext $CT = (T, \tilde{C}, C, \{C_y, C'_y \mid \forall y \in Y\})$; the primate key SK; and the set of possessed attributes S.

1: **function** (DecryptNode (CT, SK, x))
2: **if** x is a leaf node of T **then**
3: Let $i = att(x)$
4: **if** $i \in S$ **then**

$$\text{Return } \frac{e(D_i, C_x)}{e(D'_i, C'_x)} = e(g,g)^{rq_x(0)}; \tag{7}$$

5: **else** Return $\perp$.
6: **end if**
7: **else**
8: **for** each child node z of x **do**
9: $F_z = DecryptNode(CT, SK, z)$
10: **end for**
11: Let S_x be an arbitrary k_x-sized set of child nodes of x such that $F_z \neq \perp$ if $z \in S_x$.
12: **if** S_x exists **then**

$$\begin{aligned} F_x &= \prod_{z \in S_x} F_z^{\triangle_{i,S'_x}(0)} = \prod_{z \in S_x} (e(g,g)^{r \cdot q_z(0)})^{\triangle_{i,S'_x}(0)} \\ &= \prod_{z \in S_x} e(g,g)^{r \cdot q_x(i) \cdot \triangle_{i,S'_x}(0)} \\ &= e(g,g)^{r \cdot q_x(0)}, \\ &\quad \text{where } i = index(z) \text{ and } S'_x = \{index(z) : z \in S_x\}. \end{aligned}$$

13: Return F_x
14: **else**
15: Return $F_x = \perp$
16: **end if**
17: **end if**
18: **end function**
19: $A = DecryptNode(CT, SK, R)$
20: **if** $A \neq \perp$ **then**
21: $\tilde{A} = e(C, D)/A = e(g,g)^{\alpha s}$;
22: **end if**
23: The decryption is performed as follows:

$$K' = \tilde{C}/\tilde{A}.$$

24: **if** $e(H(K'|\tilde{A}), g) = e(H(K|e(g,g)^{\alpha s}), g)$ **then**
25: The message K' is valid.
26: **end if**

5. The data sink replaces the previously encrypted access token with the new one.

6. The data consumer decrypts $Encryption(PK, K'_1||date, T)$, and then sends the salted hash $h' = H(K'_1||date)$ to the sensor: $Data\ Consumer \rightarrow Sensor : h' = H(K'_1||date)$. The sensor verifies whether or not $h' = h$. If it is true, the data consumer and the sensor go to the next phase to start secure communications based on the shared secret K'_1.

3.4.2 Communication Phase

This phase contains the following two steps.

1. The data consumer sends the instruction I to the sensor by using the shared secret K'_1:

$$Data\ Consumer \rightarrow Sensor :$$
$$(ID_d, ID_s, AES(K'_1, I), h = H(K'_1||I||ID_s)). \tag{10}$$

2. The sensor decrypts the message and obtains I'. Then it computes $h' = H(K'_1||I'||ID_s)$. if $h' = h$, the message integrity is proven.

4. ANALYSIS OF THE PROPOSED SCHEME

In this section, we prove the correctness of the scheme, and analyze its security from the aspects of resistance to possible major attacks and authenticity.

4.1 The Correctness of the Proposed Scheme

In this subsection, we show that our proposed scheme is indeed feasible and correct. **Algorithm** 4 can verify whether the received data encryption key has been forged or falsified. From **Algorithm** 4, we have

$$\begin{aligned} K' &= \tilde{C}/\tilde{A} = \tilde{C}/(e(C,D)/A) \\ &= \tilde{C}/(e(h^s, g^{(\alpha+r)/\beta})/e(g,g)^{rs}) \\ &= Ke(g,g)^{\alpha s}/(e(g^{\beta s}, g^{\alpha+r/\beta})/e(g,g)^{rs}) \\ &= Ke(g,g)^{\alpha s}/(e(g,g)^{\beta s\cdot(\alpha+r)/\beta}/e(g,g)^{rs}) \\ &= Ke(g,g)^{\alpha s}/(e(g,g)^{(\alpha s+rs)}/e(g,g)^{rs}) \\ &= Ke(g,g)^{\alpha s}/e(g,g)^{\alpha s} = K. \end{aligned}$$

Thus if $e(H(K'|\tilde{A}), g) = e(H(K|e(g,g)^{\alpha s}), g)$, K' is valid. When a data consumer receives a valid K', he could decrypt the ciphertext using K' to obtain the message M.

4.2 Security Analysis

In this subsection, we analyze the security strength of the proposed scheme by examining how it can counter possible major attacks.

4.2.1 Collusion Attack Resistance

In our application of CP_ABE, the set of attributes composes the identity. In order to provide different users with different access rights, the scheme provides an access tree structure for each encrypted data item, and requires only a subset of the attributes for decryption. Thus our scheme can defend against collusion attacks although the original ABE does suffer from such an attack.

For example, assume that none of two data consumers possesses a sufficient number of attributes to successfully decrypt the ciphertext CK alone but their combined attributes contain the set of attributes required by CK. We claim that it is impossible for these two data consumers to make a successful collusion attack because they are not able to combine the secret keys associated with their attributes to get a private key that can be used to decrypt the ciphertext CK according to **Algorithm** 2 and **Algorithm** 4, since a unique secret random number r is selected for each data consumer to compute his private key. Thus the proposed scheme is secure against collusion attacks.

4.2.2 Data Encryption Key Authentication

Assume that a data consumer wants to get the data encryption key K from a sensor. Before K is stored in the data sink, the sensor has encrypted it with **Algorithm** 3. When the data consumer intends to obtain K from the data sink, he needs to get his private key $SK = (D = g^{\frac{(\alpha+r)}{\beta}}, \{(D_j = g^r \cdot H(j)^{r_j}, D'_j = g^{r_j})\} \mid \forall j \in S)$ from KGC, which is computed by **Algorithm** 2. The data consumer decrypts the ciphertext and verifies its authenticity by **Algorithm** 4: if $e(H(K'|\tilde{A}), g) = e(H(K|e(g,g)^{\alpha s}), g)$ is established, the decrypted data encryption key *K* is valid; otherwise, it is discarded.

4.2.3 Two Phase Commitment

We add the second phase of authentication in our protocol proposed in Section 3.4 by letting the sensor generate an access token again and challenge the data consumer. This two-phase commitment can protect the session from the following two vulnerable scenarios: i) an attacker may get a chance to obtain the access token since the attacker has the time to do the crack off-line (the access token refreshes at a certain time period); and ii) the data consumer may accidentally leak its access token to an attacker. The second phase of authentication can effectively correct the corresponding errors by generating a new access token.

Note that the sensor needs to send the newly encrypted access token to replace the old one in the data sink. This helps to defend against the following malicious attack: Suppose somehow an attacker obtains the access token and contacts the sensor for the challenge. The attacker's chance of winning the challenge game is negligible. But the attacker can keep on requesting new challenges, which consumes the sensor's computational power and drains its battery quickly. By replacing the old access token with the new one in the data sink, we eliminate the chance of such a malicious battery-draining attack.

5. CONCLUSION AND FUTURE WORK

In this paper, we present an efficient one-to-many attribute-based encryption scheme to secure the communications between the data consumers and the BAN (querying the data sink for sensitive patient data and sensing commands to implanted/wearable sensors). In our scheme, data are stored in ciphertext format at the data sink to support the access by different consumers with different sets of attributes. Such a mechanism successfully decreases the trust placed on the data sink. Our future research lies in the following direction: design a more efficient encryption approach with less computational and storage requirement, which could be better suitable for practical BAN applications.

Acknowledgments

This project was supported in part by US National Science Foundation under grants CNS-1017662 and CNS-0963957, and in part by the National Natural Science Foundation of China under grants 60973114 and 61170249.

6. REFERENCES

[1] J. Akinyele, M. Pagano, M. Green, C. Lehmann, Z. Peterson, and A. Rubin. Securing electronic medical records using attribute-based encryption on mobile devices. In *Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices*, pages 75–86. ACM, 2011.

[2] J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy attribute-based encryption. In *Proceedings of the 2007 IEEE Symposium on Security and Privacy*, pages 321–334, 2007.

[3] M. Dehkordi and S. Mashhadi. An efficient threshold verifiable multi-secret sharing. *Computer Standards & Interfaces*, 30(3):187–190, 2008.

[4] C. Hu, X. Cheng, F. Zhang, D. Wu, X. Liao, and D. Chen. OPFKA: Secure and efficient ordered-physiological-feature-based key agreement for wireless body area networks. In *INFOCOM*, 2013.

[5] C. Hu, X. Liao, and X. Cheng. Verifiable multi-secret sharing based on lrsr sequences. *Theoretical Computer Science*, 445:52–62, August 2012.

[6] C. Hu, N. Zhang, H. Li, X. Cheng, and X. Liao. Body area network security: A fuzzy attribute-based signcryption scheme. *to appear in IEEE Journal on Selected Areas in Communications (JSAC), Special Issue on Emerging Technologies in Communications*, 2012.

[7] A. Sahai. and B. Waters. Fuzzy identity-based encryption. In *Advances in Cryptology–EUROCRYPT 2005*, pages 457–473, 2005.

[8] A. Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[9] C. Tan, H. Wang, S. Zhong, and Q. Li. Body sensor network security: an identity-based cryptography approach. In *Proceedings of the first ACM conference on Wireless network security*, pages 148–153, 2008.