Secure and Efficient data communication protocol for Wireless Body Area Networks

Chunqiang Hu, *Student Member, IEEE,* Hongjuan Li, Xiuzhen Cheng, *Fellow, IEEE,* Xiaofeng Liao, *Senior Member, IEEE*

Abstract—Wireless Body Area Networks (WBANs) are expected to play a major role in the field of patient-health monitoring in the near future, which gains tremendous attention amongst researchers in recent years. One of the challenges is to establish a secure communication architecture between sensors and users, whilst addressing the prevalent security and privacy concerns.In this paper, we propose a communication architecture for BANs, and design a scheme to secure the data communications between implanted /wearable sensors and the data sink/data consumers (doctors or nurse) by employing Ciphertext-Policy Attribute Based Encryption (CP_ABE) [1] and signature to store the data in ciphertext format at the data sink, hence ensuring data security. Our scheme achieves a role-based access control by employing an access control tree defined by the attributes of the data. We also design two protocols to securely retrieve the sensitive data from a BAN and instruct the sensors in a BAN. We analyze the proposed scheme, and argue that it provides message authenticity and collusion resistance, and is efficient and feasible. We also evaluate its performance in terms of energy consumption and communication/computation overhead.

Index Terms—Wireless Body Area Networks; Access control tree; Secure communications; Attribute-based cryptosystem; signature.

1 Introduction

In recent years, innovative health-oriented networking and wireless communication technologies have been developed, which become an intrinsic part of many modern medical devices. The implantable medical devices (IMDs) [3], including pacemakers, cardiac defibrillators, insulin pumps, neurostimulators, etc., utilize their wireless radios to deliver timely patient information, leading to a better health care monitoring system. Current advances make it possible to deploy battery-powered miniaturized IMDs on, in, or around the human body for long-term healthcare monitoring [4]. IMDs report their data to a data sink by wireless communication channels. The data sink can be an IMD designed to store data or a smartphone, which has the ability to communicate with a remote healthcare agency through cellular networks or the Internet. All those IMDs, which will later be simply referred as sensors, and the data sink together consist a small-scale wireless sensor network, called a Wireless Body Area Network (WBAN). WBAN 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. The soaring national health expenditures and escalating age-related disabilities are shifting the emphasis from the hospital to the home [5], which makes WBANs a perfect candidate for enabling in-home monitoring and diagnosis, especially for people having chronic diseases.

Unlike conventional sensor networks, a WBAN 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 [6]. As a sensor that collects patient information, all it cares is to distribute the information to authorized doctors and other experts securely. However, there are challenges everywhere: Data should be transmitted in a secure channel, and we all know the challenges in securing wireless communication channels. Node authentication is the most fundamental step towards a BAN's initial trust establishment, key generation, and subsequent secure communications. There exist research that enables embedded sensors to establish a session key with each other by leverage physiological signals such as Electrocardiograph (ECG) [7], [8], [9], [10], [11], [12], [13], [14], [15]. Also, we can pre-distribute keys or secrets in sensors if necessary. From the perspective of cryptography, the high computation cost of asymmetric cryptography leaves symmetric encryption as the only viable option. But the key-distribution in symmetric encryption is challenging. And symmetric encryption is not a good choice for broadcasting a message because it involves some challenging issues, such as key-management and access control. At the same time, due to the limitation of memory space in sensors, a data sink, which has considerably larger memory and computation power, is employed to store 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 he captures the device. Moreover, 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, almost all apps read user's location).

Here is a basic scenario: a set of sensors with limited computation power and storage are implanted into or attached to a human body for data collection. The sensor wants to distribute its collected data securely to authorized doctors and other experts. The only thing that the sensor needs to know is that the doctor or expert has the privilege to access its data. There is no need for the sensor to know in detail who the doctor is. Meanwhile, the data produced may be requested by more than one authorized

The preliminary version of this paper appears in [2]

- *Chunqiang Hu, Hongjuan Li and Xiuzhen Cheng are with the Department of Computer Science, The George Washington University, Washington DC, 20052, USA.E-mail: {chu, cheng}@gwu.edu*
- *Xiaofeng Liao is with the College of Computer Science, Chongqing, Chongqing, 400715, China. E-mail: xfliao@cqu.edu.cn*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMSCS.2016.2525997, IEEE Transactions on Multi-Scale Computing Systems

data consumer, as long as they all have the access privilege. To be more specific, we need a role-based access control. For example, the data produced by a sensor that monitors the ECG signal may only want the doctors in GWU hospital, Cardiac Surgery Center to read it, and there are many doctors that have the required property. Moreover, the storage in a sensor is limited and the data collected should be stored in a data sink that has a larger storage. As we mentioned before, a data sink might be compromised physically or virtually. Therefore we need to eliminate the trust we put on the data sink by encrypting the stored data at the data sink. Thus the data sink itself has no access to the original data: it is just a storage device and the only functionality required is to store and index the data. In this paper, we propose a framework that makes this scenario secure by designing a protocol that facilitates role-based encrypted access control and reduces the trust we place on the data sink.

We propose a novel encryption and signature scheme based on CP_ABE in this paper to address the secure communication problem and provide the required security services mentioned above for BANs. A sensor can control the access to the data it has produced by constructing an access structure. For example, by constructing the access structure ($\{$GWU hospital$\}$ AND $\{$Vascular Surgery OR Cardiac Surgery$\}$), the data requires that only doctors or experts in GWU hospital, Vascular Surgery Center or Cardiac Surgery Center can have the access right. Data are stored in ciphertext format at the data sink and the trust we put on the data sink is now drastically decreased as the data sink does not have the key to decrypt the stored ciphertext. However, the scheme belongs to the asymmetric encryption family, which implies a high computational cost. This problem is addressed by using the scheme to encrypt a session key and then the data is encrypted by symmetric encryption based on the session key.

Our contribution can be summarized as follow:

- We propose a framework that enables authorized doctors and experts to access a patient's private medical information securely.
- Instead of using software or other mechanism to perform access control, we use encryption and signature method to provide a role-based encrypted access control. The sensor has the ability to control who has access to its data by constructing an access structure for the data.
- We minimize the trust that people usually put on the data sink by storing the data in ciphertext. The compromise of the data stored at the data sink does not necessarily indicate that the data is compromised.
- We evaluate the performance of the proposed scheme in terms of energy consumption and communication/computation overhead.

The notations and their meaning utilized in this paper to describe our scheme are presented in table 1:

The rest of the paper is organized as follows. Section 2 introduces the motivation of the study and overviews the related work. We present the system model in Section 3, and develop the main idea of the communication protocols in Section 4. Section 5 analyzes the security of the proposed protocol, and presents the performance analysis, followed by a conclusion drawn in Section 6.

TABLE 1
The notations

Notations	means
$\mathbb{G}_1, \mathbb{G}_2$	The two bilinear groups of prime order p
H	A Hash function
$\mathbb{Z}_p$	The Integers Modulo p
M	Plaintext message
AES	Advanced Encryption Standard (128-bit)
g	A generator of $\mathbb{G}_1$
$T(i)$	A function
K	Session key
K_1	Access Token

2 Motivation and Related Work

2.1 Motivation

In a healthcare or an assisted-living BAN, the data controller (could be a mobile device such as a smart phone) should be accessed by a number of parties such as the primary doctor of the patient and the doctors and nurses on duty of the day when the patient is hospitalized. To make the matter even more complex, a patient might be sent to a different hospital each time. One can see that different parties have different access rights - e.g., the primary doctor and the doctors on duty should have the full access right; a nurse should have restricted access right compared with a doctor; and the patient him/herself should have even less access right to avoid mis-configuration of the system by mistakes. In the design of BAN security mechanisms, we therefore face a critical technical challenge: how to properly regulate the access rights of these involved personnel while providing a strong access control to the sensitive patient data?

To tackle this challenge, we propose to design an attribute-based security scheme that can support not only differentiated encryption mechanisms but also role-based strong access control. To protect against information exposure due to theft or compromise of the data controller, and to control the access to the data controller or the BAN devices (implanted or wearable sensors), the attribute-based encryption over IBE [16] [1] is to be investigated. In attribute-based encryption, the identity of a user has been replaced by a set of descriptive attributes, which forms a fuzzy identity. The decryption of the ciphertext requires the attributes defined by the sender. For example, in the CP_ABE scheme, the access was defined by an access tree associated with the ciphertext. In this paper, we propose algorithms to regulate the access rights of the users based on the attribute-based encryption over CP_ABE. The performance of this design in terms of energy consumption and communication/computation overhead will be extensively studied.

2.2 Related Work

In this section, we summarize the most relevant existing research along three lines: (1) securing individual (implantable) devices within a BAN; (2) securing the communications within a BAN; and (3) identity-based cryptography for BANs. To the best of our knowledge, no prior work investigated the security of communications between a BAN and its external users except [17] [18], with [17] focusing on securing the communications (data encryption, access control, and digital signature) between the data controller and an external user via fuzzy attribute-based encryption and [18] addressing self-protecting electronic medical records (EMRs) on mobile devices and offline communications using attribute-based encryption.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMSCS.2016.2525997, IEEE Transactions on Multi-Scale Computing Systems

Individual BAN devices: Halperin *et al.* [19] analyzed the security and privacy properties of commercially available Implantable Cardiac Defibrillators (ICDs). They identified a number of radio-based attacks that could compromise the safety and privacy of a patient. Other studies also discussed potential security and privacy risks of Implantable Medical Decives (IMDs) [20] [21] [22]. The existing research in this category is orthogonal to our work presented in this paper, as we focus on securing BAN communications.

Within a BAN: Most existing work in this category focused on securing the transmissions between an implantable device and a BAN controller, which can be a mobile phone carried by the patient. There have been extensive research on leveraging a unique feature of BAN - i.e., its ability to detect/measure vital signs such as inter-pulse-intervals (IPIs) - to establish secret keys and thereby enable secure communications within a BAN [7], [8], [9], [10], [11], [12]. In particular, since the IPI reading of a patient is measurable and fairly consistent over different places of the body, and generally differs substantially from other patients, most existing work assumed that IPI can be retrieved by all body sensors and used as a unique random number generator for cryptographic schemes (after a de-noising procedure such as [23]).

Nonetheless, our studies indicate that this type of vital-sign-based techniques may not suffice for the security requirement of BANs, specifically for the following reasons:

- It has been shown recently [24] that a patient's IPI information may be remotely captured by an ultra-wide-band (UWB) radar device. This leads to a significant security threat as an adversary with a UWB radar can first capture the IPI and then use it to compromise the patient's health information.
- While IPI can be measured over various places of a human body, there are still many devices in a BAN that cannot reliably capture IPI information. Examples include motion sensors placed in shoes, cameras attached to eyeglasses, etc.

There also exists extensive research on *in-situ key establishment* [25], [26], [27] and *key redistribution* [28], [29]. They were proposed for general sensor networks and could be applicable to BANs to secure the inter-device communications.

Identity-based cryptography: With identity-based cryptography, the public key of each user can be easily computed from a string corresponding to the user's identity. Since this eliminates the cost of certificate distribution, identity-based cryptography is especially suitable for BANs.

Tan *et al.* [30] proposed an identity-based encryption scheme for BANs. Nonetheless, it lacks the access control feature which we develop in this paper. Yu *et al.* [31] developed a distributed fine-grained access-control mechanism for wireless sensor networks. But it does not provide message authentication - another important requirement of BAN security.

While identity-based cryptography [32], [33], [34], [35] has been used to provide message authentication before, the application of them to BANs may not be practical for implantable devices due to their extremely limited computation/communication capacity and battery power. In contrast, we develop a data communication scheme in this paper which has significantly lower communication overhead and power consumption.

3 Preliminaries and System model

3.1 Preliminaries

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

3.1.1 Bilinear Maps and Bilinear Diffie-Hellman problems

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 is an efficient algorithm to compute $e(P, Q)$ for $\forall P, Q \in \mathbb{G}_1$.

With a bilinear map, one can obtain the following variation of the Diffie-Hellman problem. Note that the hardness [36] of the decision version of it - i.e., the decisional bilinear Diffie-Hellman problem (DBDH) - forms the basis for the security of our scheme.

Bilinear Diffie-Hellman problem (BDH): Given two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ with the same prime order p, let $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ be a bilinear map and g be a generator of $\mathbb{G}_1$. The objective of BDH is to compute $e(g, g)^{abc}$ in $(\mathbb{G}_1, \mathbb{G}_2, e)$ from the given (g, g^a, g^b, g^c), where $a, b, c \in \mathbb{Z}_p$.

3.1.2 Secret Sharing

Another important cryptographic primitive used by our scheme is secret sharing [37], [38]. In the context of a *dealer* sharing a secret with n *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 $s \in \mathbb{Z}_p$, where $p > n$. Before secret sharing, each participant u_i holds a pairwise secret key $k_i \in \mathbb{Z}_p$, which is only known by u_i and the dealer.

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

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

by randomly choosing $t-1$ i.i.d. coefficients (the a_js) 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 $f(z)$ - i.e., $s = f(0)$. Then, in the second step, the dealer transmits to each u_i a secret share s_i computed from k_i, the secret key known only by u_i and the dealer.

$$s_i = f(k_i), \tag{2}$$

We now show how t or more users can cooperate to recover s by sharing the secret shares received from the dealer. Without loss of generality, let $u_1, \ldots, u_t$ be the cooperating users. These t users can reconstruct the secret $s = f(0)$ from $s_1 = f(k_1), \ldots, s_t = f(k_t)$ by computing

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

Note that the cumulative product in (3) is essentially a Lagrange coefficient. The correctness of (3) can be easily verified based on the definition of $f(z)$.

3.2 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 (the BAN data controller or a mobile device such as a smart phone), and Data Consumer (doctors or nurses). In the following subsections, we summarize the major functions of each entity.

3.2.1 The Key Generation Center (KGC)

The KGC is used to perform system initialization, generate public parameters, and assign a secret key for each of the attributes a data consumer claims to have. 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 and the KGC will generate a secret key for each attribute. One can see that the secret keys are uniquely generated for the data consumer, which implies that random numbers need to be associated with the set of secret keys to prevent collusion attacks. Sensors have all public parameters, which means that each sensor can construct an access tree and encrypt its data according to the access tree. Once a data consumer's attributes satisfy the access tree, it should be able to decrypt the message using the corresponding secret keys.

3.2.2 Implanted and Wearable Sensors

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. These sensors are exploited to monitor vital body parameters or body movements (e.g., endoscopy capsules and motion sensors), and/or control the human body by providing life support, visual/audio feedback, etc. A BAN can be used by its human bearer for a variety of applications, including health care, military combat support, and athletic training, just to name a few.

Implanted devices suffer from extremely restricted resources in terms of battery power, storage, and computation capability. Wearable devices, on the other hand, have much less stringent resource constraints. They are usually battery-powered and the batteries can be changed/recharged relatively easily. Wearable devices far exceed implanted ones in both quantity and heterogeneity. Example wearable devices include the sensors monitoring the cardiovascular system (electrodes on the chest to capture ECG, Peizo sensors on the wrist to measure blood pressure, optical sensors on the toe and earlobe to measure the pulse rate, microphones on the chest to measure heart sounds, etc.), the motion sensors placed on knees or in shoes, small cameras or video cameras attached to the sunglasses, and radars attached to the clothes or the stick to assist visually-disabled persons, etc.

The BAN devices should have certain computation capability to encrypt the patient's data and store the ciphertext into the data sink. When a doctor or a nurse needs the data, she/he needs to communicate with the data sink to retrieve the (encrypted) data.

3.2.3 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 [1] to encrypt the data and store the ciphertext in the data sink according to the requirements of the BAN. After data consumers retrieve a data item from the data sink, they can decrypt the data as long as they possess the secret key for the corresponding attributes specified by the access tree of the data.

In a traditional framework, the data sink is used to authenticate the identity of a data consumer, verify its authorization status, retrieve and encrypt the data requested (with the keys shared by the data consumer and the data sink), and then send the data to the data consumer. Thus the data sink plays a vital role and we have to completely trust it. In other words, if we employ a mobile device such as a smartphone with a database that enables role-based access control as the data sink, we need to trust the smartphone to authenticate the data consumer, check the data consumer's privilege, and establish a secure channel with the data consumer. If the smart phone is physically stolen or lost, the attacker can retrieve the data by analyzing the memory or disk. On the other hand, some applications in a smartphone often cross the line to collect unnecessary data, making such a data sink even more vulnerable to various attacks.

In our framework, we leverage the fact that CP_ABE can enable sensors to store the data in ciphertext; thus the data sink itself has no access to the original data. The only requirement for the data sink is to functionally store the encrypted data and disseminate the data to the data consumers that make requests. By this way we minimize the trust we usually put on the data sink. Therefore if we use a smartphone to store the data, the curious applications that intend to learn the data can obtain only the encrypted version. Based on the above analysis, in this study we assume that the data sink is honest but curious and easy to be compromised.

3.2.4 Data Consumers (DCs)

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. When the first time a data consumer joins the system, he needs to contact the *KGC* to obtain the secret key corresponding to the attributes he claims to have. The detailed method that shows how the data consumer can prove to the *KGC* that he possesses a set of attributes is out of the scope of this paper. For example, a data consumer can go to the *KGC* office and prove to the officer that he is a doctor in both *GWU* hospital and Cardiac Surgery Center. Then the *KGC* should generate a unique set of secret keys for the data consumer. One should notice that the secret keys are the crux to decrypt a message, not the attributes themselves. Attributes are public parameters and everyone could possibly know them. The secret keys for a data consumer are uniquely generated by *KGC*, which typically associates a random number with each key, to enable data consumer's ability to decrypt a message and simultaneously prevent collusion attacks.

Notice that as long as a data consumer has the required attributes, he can decrypt the data and communicate with the sensors. If there are two doctors in the system that possess the same set of attributes, they are the same data consumer from the perspective of the sensors.

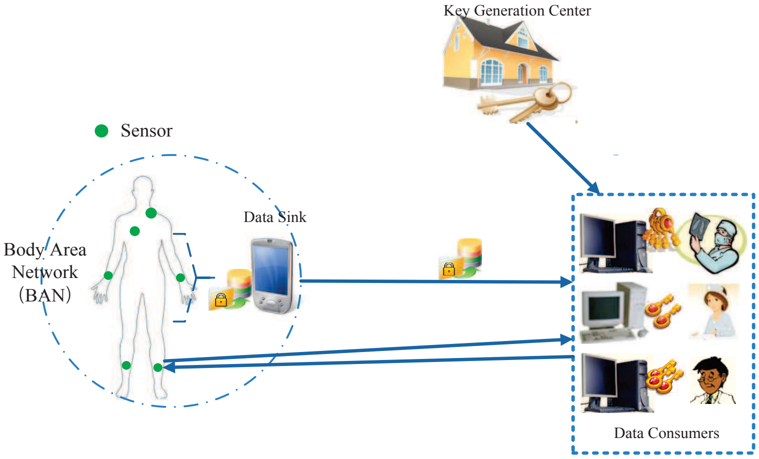


Fig. 1. A BAN architecture of a health care application.

3.3 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 consumer's access to the encrypted data. A similar idea is adopted by [1]. In such an access tree T, each non-leaf node represents a threshold gate, which is described by its children and a threshold value. 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 indicating that node x performs the *OR* operation over all the subsets of k_x child nodes of x, with each subset supporting an *AND* operation. Each leaf node x is described by an attribute and a threshold value $k_x = 1$. When a data item is generated, its associated attributes defining the access rights are used to create a tree for access control, which implies that only the users possessing the attributes of the data item can decrypt the encrypted data.

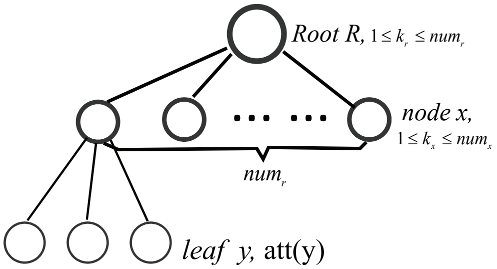


Fig. 2. An access control tree structure in a BAN.

4 The Proposed Secure Data Communication Protocol

In this section, we propose data communication protocols based on CP_ABE [1] 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 get the patient's information or distribute instructions and commands to the BAN. For example, a sensor in the BAN may specify the following access structure for the data it has collected: ({GWU hospital} AND {Vascular Surgery OR Cardiac Surgery}), which indicates that only a doctor, who comes from GWU hospital and works in the Vascular or Cardiac Surgery Center, has the access right to this piece of data. Thus a doctor in GWU hospital that has one of the following sets of attributes: {GWU hospital, Vascular Surgery}, {GWU hospital, Cardiac Surgery}, and {GWU hospital, Cardiac Surgery, Vascular Surgery}, has an access to the data mentioned above. Note that all the data stored in the mobile device are encrypted. Also note that the access right of the data is described by an access tree specified by the patient's sensor that intends to achieve a role-based access control (RBAC). Fig. 3 illustrates the aforementioned example access tree structure.

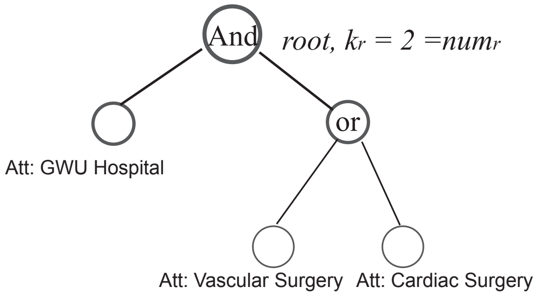


Fig. 3. An example access control structure.

4.1 The Data Communications Between Data Consumers and the Data Sink

Our scheme consists of four Algorithms. **Algorithm** 1 presents the system initialization performed by *KGC*. To be more specific, the *KGC* generates and distributes the public parameters to all the entities in the system. **Algorithm** 2 is executed by *KGC* to generate private keys for the users based on their attributes. For each attribute a user possesses, a private key needs to be generated, which can be used later to decrypt a ciphertext if the attributes satisfy the access tree of the original data. The encryption procedure is detailed in **Algorithm** 3: a session key K needs to be encrypted with an access tree T specified by the sensor, which is called the "sender" in the protocol. **Algorithm** 4 implements decryption and authentication, which should be executed by the data consumer (called "receiver" in the protocol) to get the session key based on his attributes and the corresponding secret keys for his attributes since he receives only encrypted data from the sensor/sender. Note that **Algorithm** 4 is similar to the originally proposed by CP_ABE [1].

Algorithm 1 System Initialization

```
Selects a prime $p$, a generator $g$ of $\mathbb{G}_0$, and a bilinear map $e : \mathbb{G}_0 \times \mathbb{G}_0 \to \mathbb{G}_1$.
Defines a Lagrange coefficient $\triangle_{i,S}$ for $i \in \mathbb{Z}_p$ and a set $S$ of elements in $\mathbb{Z}_p$: $\triangle_{i,S} = \prod_{j \in S, j \neq i} \frac{x-j}{i-j}$.
Chooses two random exponents $\alpha, \beta \in \mathbb{Z}_p$.
Selects a hash function $H : \{0,1\}^* \to \mathbb{G}_0$. The function $H$ is viewed as a random oracle.
Distributes the public parameters of the system given by
    $$PK = \mathbb{G}_0, g, h = g^{\beta}, e(g,g)^{\alpha} \qquad (4)$$
Computes the master key *MSK* is $(\beta, g^{\alpha})$.
```

Algorithm 2 Key Generation (*MSK*, S)

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

```
Select random number $r_{sn} \in \mathbb{Z}_p$, $K_{sign} = r_{sn}$, and calculate the verification key $K_{ver} = g^{r_{sn}}$.
The *KGC* chooses random numbers $r, r_j \in \mathbb{Z}_p$ for each attribute $j \in S$.
The secret key $SK$ is computed by
    $$SK = (D = g^{\frac{(\alpha+r)}{\beta}}, \forall j \in S : D_j = g^r \cdot H(j)^{r_j}, D'_j = g^{r_j}). \qquad (5)$$
Send $SK$ and $r_{sn}$ to the owner of the attribute set $S$ via a secure channel, and publish $K_{ver}$ for others.
```

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

Inputs: User public parameter PK; session key K; the tree T rooted at node R specifying the access right of the key K.

```
Chooses a polynomial $q_x$ and sets its degree $d_x = k_x - 1$ for each node $x$ in the tree $T$.
Chooses a random $s \in \mathbb{Z}_p$ and sets $q_R(0) = s$;
Chooses $d_R$ random points from $\mathbb{Z}_p$ to completely define the polynomial $q_R$.
for any other node $x$ in $T$ do
    Sets $q_x(0) = q_{parent(x)}(index(x))$.
    Selects $d_x$ random points from $\mathbb{Z}_p$ to completely define $q_x$.
end for
Let $Y$ be the set of leaf nodes in $T$. The ciphertext $CK$ is constructed based on the access tree $T$ as follows:
    $$CK = (T, \tilde{C} = Ke(g,g)^{\alpha s}, C = h^s, \forall y \in Y : C_y = g^{q_y(0)}, C'_y = H(att(y))^{q_y(0)}). \qquad (6)$$
Compute the signature by hashing the bitstring $K$, as $h = H(K)$. We output the signature $\sigma = h^{r_{sn}}$.
Output the message:
    $$CT = (CK, \sigma)$$
```

The following procedure illustrates how a doctor with a set of secret keys for the corresponding attribute set S obtains the required plaintext data from the data sink. Generally speaking, a sensor encrypts the body data using **Algorithm** 3, and then sends the encrypted data to a mobile device (data sink) at a regular interval. When a doctor needs to contact the data sink, decrypt the session key K, and retrieve the data, the following steps need to be performed:

1) The sensor selects a random session key K, encrypts the session key using **Algorithm** 3 and encrypts its data M by AES: $AES(K,M)$. One has to be aware that we do not write the AES algorithm in **Algorithm** 3 since the point we want to make here is that we use **Algorithm** 3 to encrypt a session key K, and the symmetric encryption can be performed as long as the session key is valid (not yet expired).
2) The sensor sends the encrypted data to the mobile device (data sink):
$$\begin{aligned} &Sensor \to datasink : \\ &(ID_s, ID_d, \textbf{Algorithm}3(K), AES(K,M)) \end{aligned} \qquad (7)$$
3) The doctor obtains the encrypted data from the data sink, and then executes **Algorithm** 4 to decrypt the encrypted data to obtain the session key K.
4) The doctor decrypts $AES(K,M)$ using the session key K.

Note that the decryption process makes use of the set S of attributes the doctor possesses to match the access tree T carried by the ciphertext CK. Then the doctor uses the secret keys corresponding to the matched attributes to recursively decrypt CK.

4.2 The Direct Communications Between Data Consumers and Sensors

When a doctor wants to send instructions or commands to a sensor in a BAN, direct communications between the doctor and the sensor are needed. Considering the limitation in computation power and storage of the sensor, we will leverage the data sink again by posting it an access token K_1, which is encrypted with an access tree specified by the sensor. The doctors or experts that have the privilege to decrypt the access token K_1 can prove to the sensor that he has the attribute by sending back the salted hash of K_1. Then the sensor challenges the doctor again by sending another access token to the data sink, which overwrites the previous one. If the doctor can prove his privilege again, a communication channel between the sensor and the doctor is set up with a key derived from the access token. We will explain why we need this two phase commitment in Section 5 when we analyze the attack resistance of our scheme. Generally speaking, our protocol can be divided into three phases: initialization phase, communication establishment phase, and communication phase. The detailed procedures are described as follows.

4.2.1 Initialization Phase

1) The KGC posts the PK according to Algorithm 1 and distributes the attributes to their corresponding owners – an owner of an attribute could be a sensor, a doctor or a nurse.
2) KGC distributes $\{SK_D, K_{sign1}\}$ and $\{SK_N, K_{sign2}\}$ to the corresponding doctor and nurse, respectively.
3) The sensor saves PK, which is typically done before it is embedded into the human body.

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

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

```
function (DecryptNode ($CT, SK, x$))
    if $x$ is a leaf node of $T$ then
        Let $i = att(x)$
        if $i \in S$ then
            Return $\frac{e(D_i, C_x)}{e(D'_i, C'_x)} = e(g,g)^{rq_x(0)}$;        (8)
        else Return $\perp$.
        end if
    else
        for each child node $z$ of $x$ do
            $F_z = DecryptNode(CT, SK, z)$
        end for
        Let $S_x$ be an arbitrary $k_x$-sized set of child nodes of $x$ such that $F_z \neq \perp$ if $z \in S_x$.
        if $S_x$ exists then
            $F_x = \prod_{z \in S_x} F_z^{\triangle_{i,S'_x}(0)}$
                $= \prod_{z \in S_x} (e(g,g)^{r \cdot q_z(0)})^{\triangle_{i,S'_x}(0)}$
                $= \prod_{z \in S_x} e(g,g)^{r \cdot q_x(i) \cdot \triangle_{i,S'_x}(0)}$
                $= e(g,g)^{r \cdot q_x(0)}$,
                where $i = index(z)$ and $S'_x = \{index(z) : z \in S_x\}$.
            Return $F_x$
        else
            Return $F_x = \perp$
        end if
    end if
end function
$A = DecryptNode(CT, Sk, R)$
if $A \neq \perp$ then
    $\tilde{A} = e(C, D)/A = e(g,g)^{\alpha s}$;
end if
The decryption is performed as follows:
    $K' = \tilde{C}/\tilde{A}$.
if $e(\sigma, g) = e(H(K'), g^{r_{sn}})$ then
    The message $K'$ is valid.
end if
```

Note that this initialization phase can be skipped if the actions have been performed when a data consumer retrieves the encrypted data from the data sink.

4.2.2 Communication Establishment Phase

1) First, the sensor chooses an access token K_1, and encrypts $K_{Tdate} = K_1 || datetime$ with the Algorithm 3. Then the sensor sends the encrypted token to the data sink: $Sensor \rightarrow Data\ sink : (ID, Algorithm3(K_{Tdate}), Hash(K_{Tdate}))$. The data sink stores the ciphertext and sends it to doctors or nurses when requested.
2) The sensor updates K_{Tdate} at certain time interval, for example, a day.
3) When a doctor obtains $(ID, Algorithm3(K_{Tdate}))$, he decrypts the ciphertext according to Algorithm 4. Then the doctor sends a salted hash of the access token to the sensor to convince the sensor that he has the required privilege: $Doctor \rightarrow Sensor : H' = H(K_{Tdate})$.
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 $K'_{Tdate} = K'_1 || datetime$ using the same access tree with Algorithm 3. The sensor sends the encrypted K'_{Tdate} to the doctor as a challenge, and then to the data sink to overwrite the previously encrypted access token.

$$\begin{aligned} &Sensor \rightarrow Doctor \quad and \quad Data\ sink : \\ &(ID, Algorithm3(K'_{Tdate}), Hash(K'_{Tdate})) \end{aligned} \tag{9}$$

5) The data sink overwrites the previously encrypted access token with the new one.
6) The doctor decrypts $(ID, Algorithm3(K'_{Tdate}), Hash(K'_{Tdate}))$, and then sends the salted hash $H' = H(K'_1 || datetime)$ to the sensor: $Doctor \rightarrow Sensor : H' = H(K'_1 || datetime)$. The sensor verifies whether or not $H' = H$. If it is true, the doctor and the sensor go to the next phase to start secure communications based on the shared secret K'_1

4.2.3 Communication Phase

The phase contains the following two steps.

1) The doctor sends the instruction I to the sensor by using the shared secret K'_1

$$\begin{aligned} &Doctor \rightarrow Sensor : \\ &(ID_d, ID_s, AES(K'_1, I), H = Hash(K'_1 || I || ID_s)). \end{aligned} \tag{10}$$

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

Notice that as long as a data consumer has the required attributes, he can learn the session key established for doctor Dr and sensor Sn. This indicates that any other doctor with the same required set of attributes can obtain the session key, hence all the communications between Dr and Sn. In our proposed scheme, all data consumers with the same set of required attributes makes no difference in the eyes of the sensor since the sensor uses the CP_ABE to perform a role-based access control. We leave the establishment of a private channel between a particular doctor and a sensor as future work.

4.3 Expiration Time of The Session Key

In this subsection, we discuss the life time of the session key, which implies the feasibility of the communication protocol. Generally specking, the CP_ABE provides asymmetric encryption with a high computation cost. Thus we choose asymmetric encryption to encrypt the session-key for establishing symmetric encryption. Then what is the expiration time of a session key?

Notice that in Section 4.2.3, a session-key is established between a sensor and a doctor for their direct communications. We didn't specify how long this session should last, which means that we didn't give an expiration time to the session-key. The expiration time depends on the requirement of the real system. If the session key is not expired, the doctor can directly contact the sensor again even after minutes of suspension. The tradeoff is obvious here: the longer the expiration time, the less the security strength. But the computation cost is less too.

In Section 4.1, we utilize a session-key generated by the sensor to prefer symmetric encryption on the produced data. Our recommendation is that a session-key for data encryption should be changed once every day. This again reflects the tradeoff we have argued before: the longer the time, the less the security, the less the computation cost.

4.4 The Revocation of The Data Consumer

In this subsection, we discuss the revocation of a data consumer. Revocation is a practical problem in real life applications. A doctor may be transferred to another hospital and his secret keys for the attributes should be revoked.

Indeed, this is a problem regarding how to revoke a user in the ABE system. In a typical ABE system, an attribute is associated with a time stamp. For example, instead of using "GWU hospital" as an attribute, we use "GWU hospital_2015" as the attribute. When the KGC generates the corresponding secret key for this attribute to the data consumer, the secret key can only correctly decrypt the message encrypted based on the attribute "GWU hospital_2015". Next year, all sensors need to encrypt their data with the attribute "GWU hospital_2016" and the data consumer should contact the KGC to get the secret key for the new attribute "GWU hospital_2016". This means that in the above example the data consumer should refresh his secret keys once a year. Then if a doctor is transferred to another hospital, he will not obtain a new secret key before a new year comes. In real life applications, we can set the revocation period to be a month or a week, which depends on the application situation. The tradeoff is obvious here: the shorter the period, the higher the frequency of the secret key updates, thus the higher the computation cost. There exists some research on real-time key revocation, which requires that once a user has been revoked, the update happens immediately [39]. However they still have a higher cost than the periodical revocation scheme mentioned before. Our argument here states that updating keys per week or per month is practical for our real life application usage.

5 Analysis of the Proposed Scheme

In this section, we prove the correctness of the scheme, analyze its security from the aspects of collusion resistance and authenticity, and then evaluate its performance in terms of energy consumption and computation/communication overhead.

5.1 The Correctness of the Proposed Scheme

In this subsection, we show that the scheme is indeed feasible and correct. **Algorithm** 4 can verify whether the received session key has been forged or falsified. From CP_ABE [1], 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 $e(\sigma, g) = e(H(K'), g^{r_{sn}})$, K' is valid. When the doctor receives a valid K', he could decrypt the ciphertext using the K' to obtain the message M.

5.2 Security analysis

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

5.2.1 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 neither a doctor Dr nor a Nurse Ns possesses a sufficient number of attributes to successfully decrypt the ciphertext CK alone. There are two reasons to make a successful collusion attack impossible. First, Dr and Ns have different attribute sets because they have different rights to access the data. Second, Dr and Ns may collude by combining their attributes in any way. However, they are not able to combine their secret keys (the SKs) to get a secret key for the combined set of attributes according to **Algorithm** 2. Thus they could not decrypt the message, indicating that the proposed scheme is secure against collusion attacks.

5.2.2 Session Key Authentication

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

5.2.3 Two Phase Commitment

Usually, there is an access token stored at the data sink. Whenever a data consumer wants to initialize a direct communication with a sensor, he needs to decrypt the access token and prove itself to the sensor.

We add the second phase of authentication in our protocol proposed in Section 4.2 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 doctor may accidentally leak its access token to an attacker. The second phase of authentication can effectively correct the error by generating a new access token.

Note that the sensor needs to send the new encrypted access token to the data sink and the data sink needs to replace the old one with the new one. This helps to defend against the following malicious attack: Suppose somehow an attacker obtains the access token and contacts the sensor for the challenge. Of course 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 computation 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 energy-draining attack.

5.3 Performance Analysis

In this subsection, we present a quantitative performance study. Our main concern is the energy consumption spent on message computation and transmission. Since the message size is directly related to the energy consumption on message transmission, which is linearly proportional to the message size [32], [40], we start from analyzing the message size.

5.3.1 Message Size

Considering the communication protocol between the data consumer and the data sink presented in section 4.1. When the doctor first accesses the data sink, he needs to obtain the session key from **Equation** (7). Then the total message size of decryption can be computed as follows according to **Equation** (6) and **Equation** (7):

$$\begin{aligned} Size &= |ID_s| + |ID_d| + |Algorithm3(K)| \\ &= |ID_s| + |ID_d| + |T| + |\tilde{C}| + |C| \\ &+ |\sigma| + |C_y| + |C'| \end{aligned} \tag{11}$$

It is sufficient for ID_r and ID_d to have 1-byte respectively, and have a four-byte $|T|$ for each data item in a typical BAN. The size of the parameters in **Equation** (11) is variable. In our evaluation, the bilinear e employs the Tate pairing. The elliptic curve is defined over $\mathbb{F}_p$. The order q of $\mathbb{G}_1$ and $\mathbb{G}_2$ is a 20-byte prime. In order to deliver a level of security equivalent to that of 1024-bit RSA, p should be a 64-byte prime if $\mathbb{G}_2$ is a q-order subgroup of the multiplicative group of the finite field $\mathbb{F}^*_{p^2}$. In the above analysis, we can set p to be 42.5 bytes in length for the finite field $\mathbb{F}^*_{p^3}$, and 20 bytes in length for the finite field $\mathbb{F}^*_{p^6}$. Therefore, the total message size of **Equation** (11) can be $6 + 5|p|$ bytes, ranging from 106 to 326 bytes.

When the communication channel is established, The message size is

$$Size = |ID_s| + |ID_d| + |AES(K, M)|. \tag{12}$$

which equals 18 bytes.

For the direct communication protocol between a data consumer and a sensor presented in the section 4.2, we can still obtain the total message size of decryption as follows according to **Equation** (6) and **Equation** (9):

$$\begin{aligned} Size &= |ID| + |Algorithm3(K_1'||date)| + |Hash(K_1'||date)| \\ &= |ID| + |T| + |\tilde{C}| + |C| + |\sigma| \\ &+ |C_y| + |C'| + |Hash(K_1'||date)| \end{aligned} \tag{13}$$

With a similar analysis, we obtain the total message size of **Equation** (13), which is $5|p| + 21$ bytes, ranging from 121 to 341 bytes.

After the data consumer successfully establishes the connection, the size of the exchanged messages can be:

$$Size = |ID_d|+|ID_s|+|AES(K_1', I)|+|H+|Hash(K_1'||I||ID_s)|, \tag{14}$$

which equals 34 bytes.

Fig. 4 demonstrates the relationship between the total message size and the number of users at different security levels. The curves in Fig. 4 indicate that the message size is independent of the number of users. For establishing a connection, our scheme needs a large message size; however, when the connection is established, the message size for the communications between a data consumer and a sensor is small. So does the communication between a data consumer and the data sink. Fig. 5 illustrates the functional relationship between the message size and the security level. We observe that the message size has a linear relationship with the security level for establishing a connection. Once the connection is established, the message size is independent of the security level.

TABLE 2
Communication Overhead

	Encryption	Decryption
DCs-data sink connection	$(5\|p\| + 6)$ bytes	1
DCs-Sensor connection	$(10\|p\| + 42)$ bytes	1
DCs-data sink communication	18 bytes	1
DCs-Sensor communication	34 bytes	1

5.3.2 Communication Overhead

The ciphertext needs to be stored in the data sink and transmitted to the data consumers when requested; thus the communication overhead is mainly related to the size of the encrypted data. From the above analysis, we know that the ciphertext size is $5|p| + 6$ for establishing the connection between a data consumer and the data sink, and is $2 * (5|p| + 21) = 10|p| + 42$ bytes for establishing the connection between a data consumer and the sensor. After the establishment of the connection, the ciphertext size is 18 bytes between the data consumer and the data sink, and is 34 bytes between the data consumer and the sensor. The overheads is shown in Table 2, which indicates that the communication overhead is mainly caused by establishing connection. We observe that the communication overhead is increasing along with the security level for establishing a connection in Fig. 6; however, the communication overhead is independent of the security level for data transmissions. The reason is that the ciphertext size is independent of the number of users after the connections between data consumers and sensors or the connections between data consumers and data sink are established. In other words, the ciphertext size does not change with security level.

5.3.3 Energy Consumption on Communications

In this subsection, we evaluate the energy consumption of Encryption in our scheme using the method proposed by [32]. As shown in [40], a Chipcon CC1000 radio used in Crossbow MICA2DOT motes consumes 28.6 μJ and 59.2 μJ to receive and transmit one byte, respectively. In our proposed scheme, he message size is $5|p| + 6$ bytes to establish the connection between a data consumer and the data sink, leading to a total energy consumption (on both transmitting and receiving messages) of $(5|p| + 6) * (28.6 + 59.2)\mu J = (0.439|p| + 0.5268)\ mJ$ for one data transmission. After the establishment of the connection, the message size is 18 bytes, leading to an energy consumption of $(18 * (28.6 + 59.2))\mu J = 1.5804\ mJ$. When there are N transmissions, the total energy consumption on communications is $(0.439|p| + 0.5268 + 1.5804 * (N - 1)) = (1.5804N + 0.439|p| - 1.0536)mJ$. On the other hand, the message size is $(5|p| + 21)$ to establish a direct connection between a data consumer and a sensor, leading to a total energy consumption of

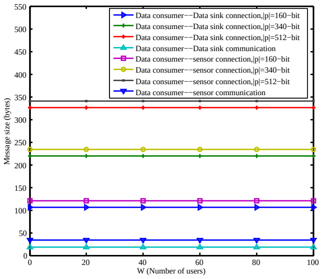


Fig. 4. Message size *vs.* W, the number of users.

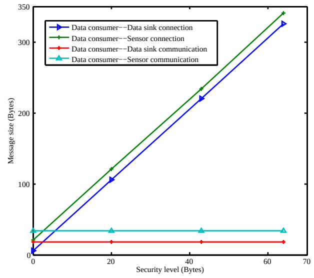


Fig. 5. Message size *vs.* security level.

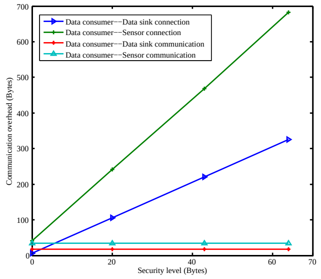


Fig. 6. The relationship between communication overhead and the security level.

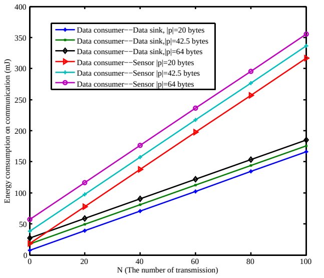


Fig. 7. Energy consumption on communications with regard to the number of ciphertext transmissions.

$2*(5|p|+21)*(28.6+59.2)\mu J=(0.878|p|+3.6876)mJ$ for one data transmission. After the establishment of the connection, the message size is 34 bytes, leading to the energy consumption of $(34*(28.6+59.2))\mu J \ = \ 2.9852\ mJ$. Thus the total energy consumption on communications is $(0.878|p|+3.6876+2.9852*(N-1)) = (2.9852N+0.878|p|+0.7024)mJ$ of N transmissions. We show the comparison between our proposed scheme and the baseline approaches on energy consumption in Table 3. Note that to evaluate the energy consumptions of the baseline approaches that use broadcasts, we adopt the model proposed by [32].

TABLE 3
Energy Consumption on Communications

The schemes	Energy consumption (mJ)
DCs–data sink, $\lvert p\rvert = 20$ bytes	$1.5804N + 7.7264$
DCs–data sink, $\lvert p\rvert = 42.5$ bytes	$1.5804N + 17.6039$
DCs–data sink, $\lvert p\rvert = 64$ bytes	$1.5804N + 27.0424$
DCs–Sensor, $\lvert p\rvert = 20$ bytes	$2.9852N + 18.2624$
DCs–Sensor, $\lvert p\rvert = 42.5$ bytes	$2.9852N + 38.0174$
DCs–Sensor, $\lvert p\rvert = 64$ bytes	$2.9852N + 56.8944$
* Certificate-based scheme $\lvert p\rvert = 64$ bytes	$146.99N$
* Merkle hash tree scheme $\lvert p\rvert = 64$ bytes	$144.56N$
* ID-based scheme $\lvert p\rvert = 64$ bytes	$111.02N$

Note: The certificate-based scheme, Merkle hash tree based scheme, and ID-based scheme are all proposed in [32]

[*] The number of transmissions is N.

Since energy consumption is linearly proportional to the message size [32], [40]. Fig.7 and Fig.8 illustrate the energy consumption on communications as a function of the number of transmissions N. Comparing the consumption between the two figures, one can see that our proposed scheme consumes significantly lower energy consumption than the schemes in [32] once the connection is established.

5.3.4 Computation Cost

We now consider the computation overhead of the proposed scheme on a 32-bit Intel PXA255 processor running at 400 MHz. According to [41], it takes approximately 752 ms to compute Tate pairing (as used in our approach) on a 32-bit ST22 smartcard microprocessor running at 33 MHz. Correspondingly, the computation of Tate pairing on PXA255 takes about $33/400 \times 752 \approx 62.04$ ms. Using the same estimation method, we obtain that it takes 18.48 ms to verify the ECDSA-160 signature according to the analysis in [32]. Note that we omit the computation overhead of hash operations and symmetric encryption operations which have a significantly lower computation cost [42].

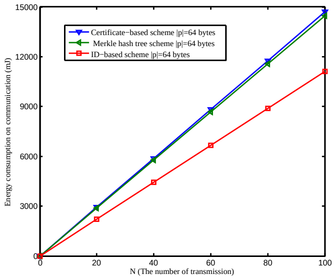


Fig. 8. Energy consumption on communications with regard to the number of ciphertext transmissions of the schemes in [32].

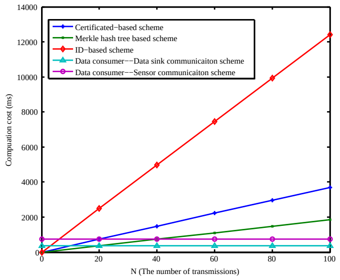


Fig. 9. Computation cost with regard to the number of transmissions.

We assume the number of transcations is N per user. In the certificate-based scheme [32], the computation cost is mainly incurred by the verification of two ECDSA signatures. Thus, the total computation cost is $2*18.48N = 36.96N$ ms. In the Merkle hash tree based scheme, the computation cost is mainly incurred by the verification of one ECDSA signature (i.e., $18.48N$ ms). In the ID-based scheme, the computation cost is mainly incurred by the two Tate pairings, with a total computation cost of $2*62.04N = 124.08N$ ms. In our proposed scheme, For the data consumer and data sink communications, the computation cost is mainly incurred by at most five Tate Pairings, with a total cost of $5*62.04 = 310.2$ ms, to established the connection; once the connection is established, the data consumer does not need to compute the Tate Pairing again until the session key is renewed. For the data consumer and sensor communications, the computation cost is mainly incurred by at most 10 Tate Paring, with a total cost of $10*62.04 = 620.4$ ms to establish the connection. Once the connection is established, the data consumer does not compute the Tate Paring again until the token is updated. We summarize the results of the energy consumption for our proposed scheme in Table 4.

Fig. 9 depicts the computation cost of the proposed scheme and the other schemes (the certificate-based scheme, the Merkle hash tree based scheme, and the ID-based scheme) proposed in [32]. One can make the following observations from the curves. First, the proposed scheme has a lower computation cost than other schemes. When we consider the energy consumption incurred by both computation and communications, our proposed scheme is relatively efficient when N is large. Moreover, the proposed scheme is based on an emerging technique, which is under rapid development, so one can expect the computation cost of the

TABLE 4
Computation cost with regard to the number of transmissions per user.

The schemes	computation cost (ms)
* Certificate-based scheme	$36.96N$
* Merkle hash tree scheme	$18.48N$
* ID-based scheme	$124.08N$
DCs-data sink communication scheme	310.2
DCs-Sensor communication scheme	620.4

Note: the certificate-based scheme, the Merkle hash tree based scheme, and the ID-based scheme are the ones proposed in [32].

* The number of transmissions is N.

proposed scheme to decrease significantly in the future.

5.4 Limitations of the Proposed Scheme

We would like to note the following limitations of the scheme proposed in this paper:

- The computation cost of establishing connections is higher than other schemes under our comparison study, leading to potential concerns of efficiency. Note that once a connection is established, the computation cost is low. Nonetheless, it is important to note that when both the cost incurred by connection establishment and that during communications are taken into consideration, our proposed scheme can be more desirable than the other schemes.
- While our proposed scheme ensures security for the communications among the data sink, data consumers, and sensors, the security of the data sink itself (from software security perspective) still needs to be properly maintained by other techniques.

6 Conclusion and Future Work

In this paper, we propose an efficient attribute-based encryption and signature scheme, which is a one-to-many encryption method. In other words, the message is meant to be read by a group of users that satisfy certain access control rules in a BAN. Meanwhile, we design a protocol to secure the data communications between implanted /wearable sensors and the data sink/data consumers.

Our future research lies in the following directions: design a more efficient encryption approaches with less computation and storage requirement (CP_ABE with constant ciphertext length), which could be better suitable for practical situations (the multi-authority CP_ABE scheme) in BAN. However, there are extra computation cost in multi-authority CP_ABE scheme and CP_ABE with constant ciphertext length. The challenge is how to reduce the computation cost for better use in BAN. Note that the communication architecture for BAN proposed in this paper serves at the basis of our future research and we shall further propose new approaches to enhance and extend this architecture.

Acknowledgments

The authors would like to thank all the reviewers for their helpful comments. This project was supported by the National Natural Science Foundation of China under Grants No.61170249 and No.61472331.

References

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

[2] C. Hu, F. Zhang, X. Cheng, X. Liao, and D. Chen, "Securing communications between external users and wireless body area networks," in *Proceedings of the 2nd ACM workshop on Hot topics on wireless network security and privacy*. ACM, 2013, pp. 31–36.

[3] D. Panescu, "Emerging technologies [wireless communication systems for implantable medical devices]," *Engineering in Medicine and Biology Magazine, IEEE*, vol. 27, no. 2, pp. 96–101, 2008.

[4] X. Liang, X. Li, Q. Shen, R. Lu, X. Lin, X. Shen, and W. Zhuang, "Exploiting prediction to enable secure and reliable routing in wireless body area networks," in *INFOCOM*. IEEE, 2012, pp. 388–396.

[5] S. Ali, V. Sivaraman, and D. Ostry, "Zero reconciliation secret key generation for body-worn health monitoring devices," in *ACM Wisec*. ACM, 2012, pp. 39–50.

[6] L. Shi, M. Li, S. Yu, and J. Yuan, "Bana: body area network authentication exploiting channel characteristics," in *ACM Wisec*. ACM, 2012, pp. 27–38.

[7] C. Poon, Y. Zhang, and S. Bao, "A novel biometrics method to secure wireless body area sensor networks for telemedicine and m-health," *IEEE Communications Magazine*, vol. 44, no. 4, pp. 73–81, 2006.

[8] K. Venkatasubramanian, A. Banerjee, and S. Gupta, "PSKA: Usable and secure key agreement scheme for body area networks," *IEEE Transactions on Information Technology in Biomedicine*, vol. 14, no. 1, pp. 60–68, 2010.

[9] ——, "EKG-based key agreement in body sensor networks," in *INFOCOM Workshops 2008*. IEEE, 2008, pp. 1–6.

[10] S. Cherukuri, K. Venkatasubramanian, and S. Gupta, "Biosec: a biometric based approach for securing communication in wireless networks of biosensors implanted in the human body," in *Parallel Processing Workshops, 2003 International Conference on*, 2003, pp. 432–439.

[11] K. Venkatasubramanian, A. Banerjee, and S. Gupta, "Plethysmogram-based secure inter-sensor communication in body area networks," in *Military Communications Conferenc*, 2008, pp. 1–7.

[12] 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.

[13] J. Zhou, Z. Cao, and X. Dong, "Bdk: secure and efficient biometric based deterministic key agreement in wireless body area networks," in *Proceedings of the 8th International Conference on Body Area Networks*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2013, pp. 488–494.

[14] S. Pirbhulal, H. Zhang, S. C. Mukhopadhyay, C. Li, Y. Wang, G. Li, W. Wu, and Y.-T. Zhang, "An efficient biometric-based algorithm using heart rate variability for securing body sensor networks," *Sensors*, vol. 15, no. 7, pp. 15 067–15 089, 2015.

[15] F. M. Bui and D. Hatzinakos, "Biometric methods for secure communications in body sensor networks: resource-efficient key management and signal-level data scrambling," *EURASIP Journal on Advances in Signal Processing*, vol. 2008, p. 109, 2008.

[16] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM conference on Computer and Communications Security*, 2006, pp. 89–98.

[17] 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.

[18] 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*. ACM, 2011, pp. 75–86.

[19] D. Halperin, T. Heydt-Benjamin, B. Ransford, S. Clark, B. Defend, W. Morgan, K. Fu, T. Kohno, and W. Maisel, "Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses," in *Proceedings of the 2008 IEEE Symposium on Security and Privacy*, 2008, pp. 129–142.

[20] W. Maisel, M. Moynahan, B. Zuckerman, T. Gross, O. Tovar, D. Tillman, and D. Schultz, "Pacemaker and icd generator malfunctions," *JAMA: the journal of the American Medical Association*, vol. 295, no. 16, pp. 1901–1905, 2006.

[21] D. Halperin, T. Kohno, T. Heydt-Benjamin, K. Fu, and W. Maisel, "Security and privacy for implantable medical devices," *Pervasive Computing, IEEE*, vol. 7, no. 1, pp. 30–39, 2008.

[22] M. Rushanan, A. D. Rubin, D. F. Kune, and C. M. Swanson, "Sok: Security and privacy in implantable medical devices and body area networks," in *Security and Privacy (SP), 2014 IEEE Symposium on*. IEEE, 2014, pp. 524–539.

[23] A. Juels and M. Wattenberg, "A fuzzy commitment scheme," in *Proceedings of the 6th ACM conference on Computer and Communications Security*, 1999, pp. 28–36.

[24] H. B. Lim, D. Baumann, and E.-P. Li, "A human body model for efficient numerical characterization of uwb signal propagation in wireless body area networks." *IEEE transactions on Biomedical Engineering*, vol. 58, no. 3, pp. 689–697, 2011.

[25] L. Ma, X. Cheng, F. Liu, F. An, and M. Rivera, "iPAK: An in-situ pairwise key bootstrapping scheme for wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 8, pp. 1174–1184, August 2007.

[26] F. Liu and X. Cheng, "LKE: A self-configuring scheme for location-aware key establishment in wireless sensor networks," *IEEE Transaction on Wireless Communications*, vol. 7, no. 1, pp. 224–232, January 2008.

[27] F. Liu, X. Cheng, L. Ma, and K. Xing, "SBK: A self-configuring framework for bootstrapping keys in sensor networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 7, pp. 858–868, 2008.

[28] L. Eschenauer and V. Gligor, "A key-management scheme for distributed sensor networks," in *Proc. Ninth ACM Conf. Computer and Comm. Security (CCS 02)*, 2002, pp. 41 – 47.

[29] W. Du, J. Deng, Y. Han, and P. Varshney, "A pairwise key predistribution scheme for wireless sensor networks," in *Proc. 10th ACM Conf. Computer and Comm. Security (CCS 03)*, 2003, pp. 228 – 258.

[30] 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*, 2008, pp. 148–153.

[31] S. Yu, K. Ren, and W. Lou, "FDAC: Toward fine-grained distributed data access control in wireless sensor networks," in *INFOCOM 2009, IEEE*, 2009, pp. 963–971.

[32] K. Ren, W. Lou, K. Zeng, and P. Moran, "On broadcast authentication in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 6, no. 11, pp. 4136–4144, 2007.

[33] J. Li, D. Wei, and H. Kou, "Secure monitoring scheme based on identity-based threshold signcryption for wireless sensor networks," in *4th International Conference on Wireless Communications, Networking and Mobile Computing*, 2008, pp. 1–4.

[34] J. Liu, J. Baek, J. Zhou, Y. Yang, and J. Wong, "Efficient online/offline identity-based signature for wireless sensor network," *International Journal of Information Security*, vol. 9, pp. 287–296, 2010.

[35] I. Kim and S. Hwang, "An efficient identity-based broadcast signcryption scheme for wireless sensor networks," in *The 6th International Symposium on Wireless and Pervasive Computing (ISWPC)*, 2011, pp. 1–6.

[36] A. Sahai. and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology–EUROCRYPT 2005*, 2005, pp. 457–473.

[37] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

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

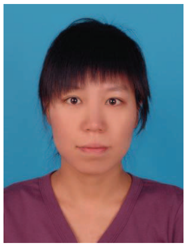
[39] J. Hur and D. Noh, "Attribute-based access control with efficient revocation in data outsourcing systems," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 22, no. 7, pp. 1214–1221, 2011.

[40] A. Wander, N. Gura, H. Eberle, V. Gupta, and S. Shantz, "Energy analysis of public-key cryptography for wireless sensor networks," in *PerCom 2005*, 2005, pp. 324–328.

[41] G. Bertoni, L. Chen, P. Fragneto, K. Harrison, and G. Pelosi, "Computing tate pairing on smartcards," in *[Online]. http://www. st. com/stonline/products/families/smartcard/ches2005 v4. pdf*, 2005.

[42] W. Dai. (2009) Crypto++ 5.6. 0 benchmarks. http://www.cryptopp.com/benchmarks.html.

Chunqiang Hu received his B.S. degree in Computer Science from Southwest University, Chongqing, China, in 2006; and M.S. degree in Computer Science from Chongqing University, Chongqing, China, in 2009. He was a visiting scholar at The George Washington University from Jan., 2011 to Dec., 2011. He is a PhD candidate in Computer Science at the George Washington University, Washington DC. His research interests include Wireless and Mobile Security, Secret Sharing, and Cryptography.



Hongjuan Li received her B.S. degree in Computer Science from Dalian Jiaotong University, Dalian, China, in 2008; and M.S. degree in Computer Science and Technology from Dalian University of Technology, Dalian, China, in 2011. She is currently pursuing her PhD. degree in Computer Science at The George Washington University. Her research interests include Cognitive Radio Networks, Ad Hoc and Sensor Networks, Wireless and Mobile Security, Algorithm Design and Analysis.

Xiuzhen Cheng received the M.S. and Ph.D. degrees in computer science from the University of Minnesota, Twin Cities, Minneapolis, in 2000 and 2002, respectively. She is a full Professor with the Department of Computer Science, The George Washington University, Washington, DC. She worked as a Program Director for the US National Science Foundation (NSF) for six months in 2006 and joined the NSF again as a part-time Program Director in April 2008. Her current research interests include Wireless and Mobile Computing, Mobile Health and Safety; Wireless and Mobile Security, Cognitive Radio Networking, and Algorithm Design and Analysis. Dr. Cheng received the NSF CAREER Award in 2004. She is a Fellow of the IEEE.

Xiaofeng Liao received the B.S. and M.S. degrees in mathematics from Sichuan University, Chengdu, China, in 1986 and 1992, respectively, and the PhD degree in circuits and systems from the University of Electronic Science and Technology of China in 1997. From 1999 to 2001, he was involved in postdoctoral research at Chongqing University, where he is currently a professor. From November 1997 to April 1998, he was a research associate at the Chinese University of Hong Kong. From October 1999 to October 2000, he was a research associate at the City University of Hong Kong. From March 2001 to June 2001 and March 2002 to June 2002, he was a senior research associate at the City University of Hong Kong. From March 2006 to April 2007, he was a research fellow at the City University of Hong Kong. He has published more than 200 international journal and conference papers. His current research interests include Neural Networks, Nonlinear Dynamical Systems, Bifurcation and Chaos, and Cryptography. He is a Senior Member of IEEE.

Nov. 14, 2015