OPFKA: Secure and Efficient Ordered-Physiological-Feature-based Key Agreement for Wireless Body Area Networks

Chunqiang Hu*†, Xiuzhen Cheng*, Fan Zhang*, Dengyuan Wu*, Xiaofeng Liao†, Dechang Chen ‡
*Department of Computer Science, The George Washington University, Washington DC 20052, USA
†College of Computer Science, Chongqing University, Chongqing 400030, China
‡Division of Epidemiology and Biostatistics, Uniformed Services University of the Health Sciences, MD 20814, USA
Email: {chu, cheng}@gwu.edu,{zfwise, andrewwu}@gwmail.gwu.edu, xfliao@cqu.edu.cn, dechang.chen@usuhs.edu

***Abstract*—Body Area Networks (BANs) are expected to play a major role in patient health monitoring in the near future. Providing an efficient key agreement with the prosperities of *plug-n-play* and *transparency* to support secure inter-sensor communications is critical especially during the stages of network initialization and reconfiguration. In this paper, we present a novel key agreement scheme termed *Ordered-Physiological-Feature-based Key Agreement (OPFKA)*, which allows two sensors belonging to the same BAN to agree on a symmetric cryptographic key generated from the overlapping physiological signal features, thus avoiding the pre-distribution of keying materials among the sensors embedded in the same human body. The secret features computed from the same physiological signal at different parts of the body by different sensors exhibit some overlap but they are not completely identical. To overcome this challenge, we detail a computationally efficient protocol to securely transfer the secret features of one sensor to another such that two sensors can easily identify the overlapping ones. This protocol possesses many nice features such as the resistance against brute force attacks. Experimental results indicate that OPFKA is secure, efficient, and feasible. Compared with the state-of-the-art PSKA protocol, OPFKA achieves a higher level of security at a lower computational overhead.**

***Index Terms*—Body Area Networks (BANs); secure inter-sensor communications; Inter-Pulse-Interval (IPI); physiological feature based key agreement.**

I. Introduction

Body area networking is a promising technology for real-time monitoring of physiological signals to support various medical applications [1]. It is enabled by the rapid development of wireless sensor networks and biomedical engineering techniques [2]–[6]. A typical body area network (BAN) consists of a number of wearable and implanted sensors to monitor the parameters of the human body and the surrounding environments such that it can assist the human body by providing life support, visual/audio feedback, etc [1].

Unlike conventional sensor networks, BANs deal with medical information with a more stringent security and privacy requirement. The sensitive nature of the collected data makes a BAN the target for adversaries to explore; and that sensors communicate wirelessly makes the BAN even more vulnerable. The lack of adequate security protections may not only lead to a breach of the patient's privacy, but also give a chance for the adversaries to threat the patient's safety by modifying the data from the BAN, which may result in wrong diagnosis and treatments [7]. Since wireless communication is one of the most vulnerable aspects of a BAN, securing inter-sensor communications plays a critical role in securing the BAN.

BANs rely on cryptographic keys to perform authentication and provide data confidentiality and integrity. Keys are usually distributed to sensors by key distribution protocols, which typically require some form of keying information pre-deployment [8]–[12]. However, with the increasing size of a BAN, traditional approaches involve a considerable latency during the network initialization or any subsequent adjustment process (e.g., phase deployment), owing to the needs for information pre-deployment. We intend to provide an efficient security scheme with the prosperities of *plug-n-play* and *transparency*. That is, users can add, remove, and tune the sensors of a BAN without reconfiguring the network but can still enjoy the benefits of secure communications. Such characteristics can help to minimize communication overhead during the initialization process and thus reveal less personal identifiable information of the patient. Some schemes have been proposed to meet these needs. For instances, Plethysmogram [7] and PSKA [13] have been presented to avoid keying information pre-deployment. However, the security level of these techniques is not high enough due to the limitation placed by the feature size and the high complexity of computing chaff points, as analyzed later in this paper.

We propose a security scheme termed *Ordered-Physiological-Feature-based Key Agreement (OPFKA)* in this paper. OPFKA employs secret features computed from the physiological signal measured at different parts of the human body to enable sensors agree on a symmetric cryptographic key in an authenticated and plug-n-play manner for securing the inter-sensor communications, i.e, no initialization is required. OPFKA does not require any key pre-distribution. It exploits the dynamic and complex nature of the human body. OPFKA works as follows: 1) the features generated by each sensor are ordered to form a feature vector and only the sensor collecting the data knows the order of the features; 2) the sender sends the secret features along

978-1-4673-5946-7/13/$31.00 ©2013 IEEE

with a large number of noisy data to the receiver; 3) the receiver generates a key according to the common features, and then returns the indexes of the matching features; and 4) the sender identifies the common features in its own feature vector and computes the key accordingly. OPFKA meets the design goals suggested by [14] for physiological signals to be a basis for key agreement, namely, the keys are long and random to prevent brute force attacks; they are efficient in terms of computational, communication, and storage overhead; and they possess the properties of time variance and distinctiveness. The main contributions of the paper are outlined as follows.

1) We propose OPFKA, a secure and efficient scheme for an authenticated key agreement between two sensors in a BAN. The scheme has the properties of transparency and plug-n-play to easily support network reconfiguration without sacrificing the already-achieved security.
2) We prove the reliability of OPFKA and analyze its efficiency and feasibility. In particular, we focus on the following security aspects: resistance against brute force attacks, message exchange security, randomness, distinctiveness, and time variance. We also discuss two methods to process physiological signals.
3) We compare OPFKA with PSKA [13] in terms of the security level, the resistance against brute force attacks, and other aforementioned design goals, and our results demonstrate the superiority of OPFKA over PSKA.
4) We estimate the performance of OPFKA in terms of the computational, communication, and storage overhead.

The rest of the paper is organized as follows. Section II overviews the related work. We present the system model in Section III, and develop the main idea of OPFKA in Section IV. Section V analyzes the security of OPFKA, and Section VI presents the performance analysis, followed by the conclusion drawn in Section VII.

II. Related Work

Most pervious work on BAN security focused on issues such as encryption [15] [16] [17], key management [18]–[21], and access control [15], [22], [23].

In order to secure the inter-sensor communications, the idea of employing physiological signals was first introduced in [1], [24], in which the features derived from a physiological signal simultaneously measured at different parts of the body are used to generate the actual key shared between the sensors. To establish a common set of features, simple error correction can be employed to correct the differences between the physiological features generated at different sensors. Based on this idea, [25] proposed to employ the Inter-Pulse-Interval (IPI) to generate cryptographic keys by encoding the IPIs into a 128-bit binary key. IPI refers to the time interval between the R-wave of the Electrocardiograph (ECG) and the foot of the Photoplethysmogram (PPG) pulse. In order for this approach to be applicable, the Hamming distance of the keys generated between two sensors belonging to the same human body should be remarkably lower than that generated by sensors at different human bodies. However, the results from a real world experimental study [13] indicated that the Hamming distances of two IPIs obtained from the same subject and different subjects are 60 and 65, respectively. Though [25] suggested that error correction can be used to improve the matches of the features derived from the same human body, the scheme is still not practical since the Hamming distance of the IPIs for the same person after error correction still varies from 0 to 40. The primary reason of this hardness lies in that the translational and rotational errors can produce drastically different values when IPIs are naively encoded into binary.

To solve the problem mentioned above, fuzzy vault based schemes [7], [13], [26] were proposed to deal with the fact that physiological signals have similar trends but are not completely identical due to the dynamic nature of a human body. The fuzzy-vault scheme has been primarily applied to biometric-based authentication such as fingerprints [27] and iris images [28]. It was argued that an adversary has a high probability to guess the legitimate points in a fuzzy vault according to the analysis in [29]. As a result, the adversary has a high probability to reduce the complexity of identifying the polynomial used by the vault. PSKA [13] and Plethysmogram [7], which are based on the the same fuzzy vault scheme but focus on different physiological signals to secure the inter-sensor communications, claimed that they had a high security level. Nevertheless, the security strength of PSKA and Plethysmogram heavily depends on the vault size, which means that the complexity of breaking the vault increases if the number of chaff points increases. However, the increase of the vault size can cause collisions between the features generated by one sensor and the chaff point generated by another sensor, which leads to a false rejection. Besides, a recent work [30] proposed a secret-key generation mechanism that exploits the signal strength fluctuations caused by the incidental motions of body-worn devices to construct shared keys with a near-perfect agreement, thereby avoiding the reconciliation cost. However, the secret bit generation rate is very limited and the cost is very high. As a result, the scheme in [30] is infeasible for a practical BAN.

Enlightened by the fuzzy vault scheme, instead of using error correction codes or reconstructing polynomials, we leverage the fact that the secret features generated by a sensor are ordered and only the sensor itself is aware of the order of the features, and propose an efficient and secure key agreement scheme termed OPFKA. OPFKA employs simple noisy data as chaff points to provide enhanced security. Our analysis indicates that OPFKA overcomes all the problems mentioned above while meeting the suggested design objectives of physiological signal based key agreement [14] for BANs.

III. System Model

A BAN is a network that interconnects physiological and environmental monitoring sensors worn on or implanted inside a human body. These sensing devices collect physiological and contextual information of a human body at a regular interval and transmit it to a highly capable sink node for further processing over multi-hop wireless communications. We assume that all sensors, worn on or implanted, are able to measure the appropriate physiological signals. We also assume that an entity that does not have a physical contact with a human body can't collect any physiological signal, and that only legitimate sensors are in contact with the human body. Thus attackers are mainly able to passively monitor the traffic as the wireless medium is not secure. Furthermore, we assume that malicious entities cannot compromise the sensors in a BAN without being detected, as the sensors are mostly under the supervision of the host and/or the caretaker.

The threats faced by a BAN are primarily from adversaries that can eavesdrop on the traffic of the BAN, replay old messages, inject messages to compromise the confidentiality of the BAN communications, or spoof the BAN sensors' identities. Adversaries may also break the key distribution process by using the physiological signal data obtained from another person if the scheme does not have sufficient distinctiveness. In this paper, we focus solely on designing a secure and efficient scheme to ensure the security of inter-sensor communications within a BAN. Communications from the sink onwards can utilize conventional security schemes such as Secure Socket Layer (SSL), given the considerable capabilities of the entities involved. Note that we do not consider denial of service (DoS) attacks such as jamming, electromagnetic interference, and battery depletion, in this paper.

IV. Key Agreement

The purpose of OPFKA is to promote secure inter-sensor communications by enabling two sensors to agree on a pairwise symmetric key based on the common physiological signal collected by the two sensors at different parts of the body. The key agreement process between two sensors works as follows. First, both sensors simultaneously and independently collect and process a certain physiological signal based on which some secret features are computed. These features are organized into an ordered set called a *feature vector* for each sensor, and the order is only known to the sensor generating the data. But a common ordering policy, which could come from the same feature generation algorithm, is adopted by all sensors. Then one of the sensors, say the sender, generates noises to hide its secret feature vector. Second, the secret features and the noisy data are sent to the other sensor, say the receiver, who can use its own version of the feature vector to identify common features as the receiver's feature vector partially overlaps with that of the sender. Thus the receiver can generate a key K based on the matched (overlapped) part of the feature vector. Finally, the receiver puts the positions (or indexes) of the matching features into a set I, and sends it along with the MAC (Message Authentication Code) of the key K to the sender, which can generate the same key K after identifying the common secret features according to I.

In [13], the authors proposed to use secret sharing scheme [31]–[33], to hide the secret key in the coefficients of a polynomial. For this scheme, the computational cost in the reconstruction process is high. We address this drawback by leveraging the fact that secret features generated by the two sensors are ordered according to the same policy and only the sensors themselves know the indexes (order) of the features in the feature vectors. Table I summarizes the utilized notations and their semantic meanings, and Fig. 1 demonstrates the OPFKA protocol, whose key procedures are detailed in the following subsections.

TABLE I
Table of notations.

Notation	Definition
H	A standard cryptographic hash function, e.g., SHA-1
$Index$	The position of a matching secret feature
IDs, IDr	The ids of the sender and the receiver
Q	The set of common features
FFT	Fast Fourier Transform
IPI	Inter-Pulse Interval signal
N	The number of the features
M	The number of the chaff points (noises)
MAC	Message Authentication Code

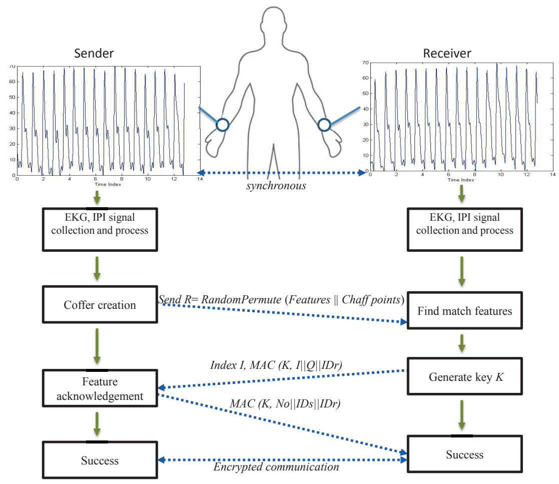

Fig. 1. The OPFKA protocol.

A. Concealing the Features

OPFKA is designed to conceal the set A of secret features generated by the sender in a construct called a *Coffer*, denoted by R. Once the set A has been concealed in the Coffer, no one can distinguish the secret features from the chaff points. As illustrated in Fig. 1, the sender constructs the Coffer by i) generating a set of random chaff points C; ii) adding A and C into the Coffer R; and iii) performing a random permutation in R. Once the receiver obtains the Coffer R, it can utilize its own set of secret features B to find the overlapping ones. We denote the set of common features by Q, i.e., $Q = \{u|u \in R \cap B\}$. Next the receiver generates a key K using the set Q, e.g., $K = H(Q)$, and then sends back the MAC of Q along with the index set I of the overlapping features. After the sender receives the index set I, it can figure out the corresponding common feature set Q. If $|Q|$ is greater than a threshold, the sender generates a key K', and then verifies the MAC of Q. If success, which implies that $K' = K$, the sender sends back an acknowledgement to the receiver. This procedure guarantees that if both sensors are aware of the set of common features, they can generate the same secret key K.

B. Coffer Packing and Unpacking in OPFKA

OPFKA requires that the secret features in the sets A and B overlap to a certain degree. The presence of the chaff points adds security to the Coffer and conceals the original secret features. Since no one can tell apart the chaff points and the features, an attacker has to perform a brute force attack in order to figure out the common features. In this subsection, we demonstrate how to employ the proposed scheme for key agreement in a BAN.

1) Feature Generation: The generation of the features has a great influence on the effectiveness of OPFKA. Since the physiological signal collected at both sides are not exactly the same, methods can be used to help improve the success rate of the key agreement and reduce the rates of false acceptance and false rejection [25]. Below we propose two methods for feature generation.

The Enhanced FFT Method: PSKA [13] proposed the following simple feature generation scheme. Two sensors that intend to establish a shared key sample an EKG or a PPG signal at a certain frequency simultaneously for a short period of time: 12.8 seconds for PPG at the frequency of 60Hz and 4 seconds for EKG at the frequency of 125Hz. Then the samples are partitioned into windows of size 256 and a 256-point Fast Fourier Transform (FFT) is performed on each window. Next the FFT coefficients of each window are passed through a peak-detection function that returns tuples of the form $\langle k_x^i, k_y^i \rangle$. Here k_x^i and k_y^i are the index and the corresponding value of the ith peak in the FFT coefficient sequence and are termed as *peak index* and *peak value*, respectively. A 13-bit feature ($[k_x^i | k_y^i]$) is generated by first quantizing a k_x^i to a 8-bit binary and the corresponding k_y^i to a 5 bit-binary and then concatenating them. For a PPG signal, about 30 features can be generated and the number of common features for two sensors in the same BAN is about 12. In contrast, the number of common features for two sensors monitoring different human bodies is about 2. PSKA creates a vault with a size varying from 1000 to 5000 and embeds the features into the vault. Our analysis (Section V) indicates that the secret features of the receiver might match with some chaff points. Theoretically the maximum size of a vault is 2^{13}, which is about 8000. If the real vault size is 3000, the probability that the receiver has features matching with the chaff points is quite high.

There is no chance for the receiver to observe that it has features matching with the chaff points, thus the recovery of the session key can be disrupted. To reduce the chance of collisions, we take one step further by expanding the 13-bit features to 20-bit ones via a one-way function (e.g., employing a salted hash function and taking the first 20 bits of its output). By feature expansion, the chance of collision is reduced significantly – all the common features at the receiver should be generated by the sender. Moreover, we record the order of the generated features in this enhanced FFT method.

The generation of the features by our enhanced FFT method takes only seconds for a PPG signal in our study; but the FFT transform and peak detection consume high computational power. Next we present our second feature generation method, which takes a longer time (1-1.5 minutes) but consumes less computational power.

The IPI Method: It has been shown in [25] that the IPIs have a high level of randomness and can be obtained with different types of sensors from different physiological signals (e.g., the ECG, PPG or blood pressure) at different parts of the body (e.g., chest, fingertips, or limbs). The collection of IPIs is relatively easy. The experimental study reported in [22] indicates that the last 4 digits of an IPI's binary representation are almost completely random, which means that we can extract 4 bits from each IPI.

The sender and the receiver collect IPIs simultaneously for about 1 to 1.5 minutes in order to obtain 90 IPIs (collecting an IPI takes about 850 ms in average). Each IPI is first quantified into a 4-bit binary representation and three adjacent IPIs are concatenated to form a 12-bit secret feature, which is then expanded and concealed in a Coffer. An experimental study reported in [22] claimed that about 75% IPI pairs collected at two sensors match, which indicates that the probability of the secret features from the sender and the receiver matching with each other is about 42%. This implies that there exist about 12 matching features computed from the 90 IPIs. To reduce the chance of collision, we expand the 12-bit features to 20-bit ones and record the order of the features.

Note that the IPI method only needs to collect IPIs and quantify them. No complex operations such as FFT transform and peak-detection are needed. Thus the IPI method consumes less computational power at the cost of a longer sampling time compared to the enhanced FFT method. Therefore the IPI method can be used to generate a session key at an hourly basis since the emerging IEEE 802.15 standard [34] for body area networks suggests that the key needs to be renewed once every hour. We denote the feature vectors of length N by $F_s = \{f_s^1, f_s^2, ..., f_s^N\}$ and $F_r = \{f_r^1, f_r^2, ..., f_r^N\}$ for the sender and the receiver, respectively. Algorithm 1 details the procedure of feature generation for both the IPI method and the enhanced FFT method.

2) Coffer Creation: After the feature vectors are computed, the sender can create a Coffer, which contains the set $F_s = \{f_s^1, f_s^2, ..., f_s^N\}$ and a larger set of M random chaff points of the form $F_s' = \{{f'}_s^j\}$, with ${f'}_s^j \notin F_s$, $1 \leq j \leq M$. Each chaff point ${f'}_s^j$ is within the same range as the features in F_s. A random permutation on the values in the Coffer is then performed, i.e., R = RandPermute($F_s \cup F_s'$), to ensure that the chaff points and the legitimate feature points are indistinguishable. The cardinality of the set F_s' can vary with respect to the level of the security requirement. The larger the set F_s', the more difficult to break the Coffer. The Coffer size R is equal to $|N| + |M|$. Algorithm 2 details the process of the Coffer creation. Section V discusses the relationship between the Coffer size and its security strength in more detail.

Algorithm 1 Feature Generation.

```
The sender and the receiver collect physiological signal;
if employing FFT to quantify the signal then
    Partition samples into windows;
    for each window of samples do
        Perform FFT and peak-detection;
        Return tuples of the form $\langle k_x^i, k_y^i \rangle$;
        Quantize $k_x^i$ and $k_y^i$ to a 5-bit binary number and an 8-bit binary number, respectively;
        Concatenate $k_x^i$ and $k_y^i$ to form a 13-bit feature ($[k_x^i|k_y^i]$);
        Compute H($[k_x^i|k_y^i]$) and take the first 20 bits of the output.
    end for
else
    Collect IPIs;
    Quantify each IPI;
    for each of three adjacent IPIs do
        Concatenate their last 4 binary digits to form a 12-bit feature;
        Compute H($[k_x^i|k_y^i]$) and take the first 20 bits of the output.
    end for
end if
Output the feature vectors: $F_s = \{f_s^1, f_s^2, ..., f_s^N\}$ (for the sender) or $F_r = \{f_r^1, f_r^2, ..., f_r^N\}$ (for the receiver).
```

Algorithm 2 Coffer Creation.

```
Compute $M$ random chaff points: $F'_s = \{{f'}_s^j\}$, where ${f'}_s^j \notin F_s$;
Randomly permute the points to obtain $R$ = RandPermute($F_s \cup F'_s$);
Output the Coffer $R$.
```

3) Feature Exchange: There are two steps in this process: a) The sender communicates the Coffer R to the receiver using the following message: Sender→ Receiver: $\{IDs, IDr, R, No\}$. Here, IDs and IDr are the ids of the sender and the receiver, respectively, and No is a nonce (unique random number) for transaction freshness. b) After the receiver obtains the Coffer R, it compares R with its own features to find the matching ones from the Coffer and records the indexes/positions of these matching features (the set Q) in its own feature vector. Denote the positions of the matching features by the index set $I = \{i\}$, where $i \in N$, and then generate the secret key $K = H(Q)$ using the matching features. The receiver feedbacks to the sender using the following message: Receiver → Sender: $\{IDs, IDr, I, MAC(K, I|Q|IDr)\}$. Algorithm 3 illustrates the process of feature exchange.

Algorithm 3 Feature Exchange.

```
The sender sends the message $\{IDs, IDr, R, No\}$ to the receiver;
The receiver identifies the matching features $Q = R \cap B$.
The receiver labels the positions of the matching features in its feature set $B$, which are denoted by $I = \{i|i \in N\}$;
The receiver generates a secret key $K$ using the matching features $Q$: $K = H(Q)$;
The receiver feedbacks the message to the sender: $\{IDs, IDr, I, MAC(K, I|Q|IDr)\}$.
```

4) Feature Acknowledgement: Upon receiving the message $\{IDs, IDr, I, MAC(K, I|Q|IDr)\}$, the sender first identifies the common features according to the set I, which contains the positions of the matching features. If $|Q|$ is greater than a threshold, the sender generates a key $K' = H(Q)$ in the same way as the receiver. If the sender successfully generates the key K' and confirms the MAC, which implies that K' equals K, it sends back an acknowledgement to the receiver using the following message: Sender → Receiver: $MAC(K, No|IDs|IDr)$. For the sender to generate K successfully, the indexed features should be exactly the same as those in the receiver. This process not only confirms the correctness of the generated key K, but also authenticates the sender to the receiver. This is because the distinctiveness and temporal variance property of the physiological signal features ensure that i) the features generated from a physiological signal for OPFKA are drastically different for two different persons and ii) old Coffers cannot be replayed as the features would have changed by that time such that the sender can't successfully verify the MAC (see Section VI for more details.). Algorithm 4 illustrates the procedure of feature acknowledgement.

Algorithm 4 Feature Acknowledgement.

```
The sender receives the message: $\{IDs, IDr, I, MAC(K, I|Q|IDr)\}$;
Identifies the common features (placed in the set $Q$) according to the position set $I$;
if $|Q| \geq Threshold$ then
    Generates the key $K'$ using the common features;
    Verifies the MAC;
    if $MAC(K', I|Q|IDr) = MAC(K, I|Q|IDr)$ then
        Return $MAC(K, No|IDs|IDr)$ to the receiver;
    end if
else
    The sender sends *None* to the receiver.
end if
```

Fig. 1 illustrates the OPFKA protocol. The secret key K generated in Algorithm 3 and Algorithm 4 is used to enable confidential, authenticated, and integrity-protected communications between two sensors in a plug-n-play manner, which is not considered in traditional key distribution schemes [9]

[16] and the physiological-signal-based approaches [25]. Furthermore, with OPFKA, no key and no physiological feature is ever reused. This ensures that any knowledge of the past keys or past physiological features of a subject can not be reused for subverting the Coffer, due to the temporal variance property, as seen in Section VI.

V. Security of OPFKA

In this section, we discuss the security implications of the two principal aspects of OPFKA: the Coffer and the message exchange.

A. *Coffer Security*

The use of OPFKA ensures that even though the two sensors may not have all the features in common, they can still agree upon a common key in a secure manner. The security of OPFKA can be understood as a trapdoor one-way function. The hiding of the legitimate feature points among a much larger number of the bogus chaff points, whose values are in the same range, makes identifying the legitimate points very difficult. An adversary, which does not know any legitimate points (as it cannot collect the relevant physiological signals from the host's body), has to try out each of the features in the set R in order to generate the secret key K. Fig. 2 demonstrates the strength of the Coffer for different Coffer sizes. The strength of the Coffer is determined by the number of combinations an adversary attempts to examine in order to find out the legitimate points and their indexes in the Coffer. For ease of understanding, we represent this computational requirement in terms of its equivalence to brute-forcing a key of a particular length (bits). As expected, increasing the Coffer size automatically increases the security provided by the Coffer. Note that OPFKA guarantees a successful feature exchanges, as long as the number of common features $|Q|$ is greater than a threshold.

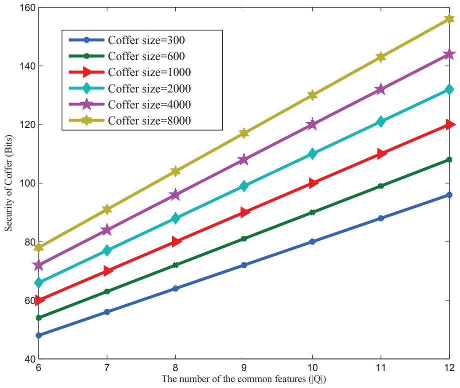

Fig. 2. Coffer strength.

We also compare the security of OPFKA with that of PSKA [13], and find out that the security strength of OPFKA is higher than that of PSKA at every security level. Note that PSKA [13] and Plethysmogram [7] employ the same fuzzy vault scheme but Plethysmogram [7] focuses on the PPG signal while PSKA [13] considers both PPG and ECG. Therefore in this study we do not compare OPFKA with Plethysmogram [7] as the latter employs the same technical approach as that of PSKA [13]. Table II reports the security strength of our scheme and that of PSKA. According to our experimental study, a 4000-point Coffer is better than a 5000-point vault in terms of the security strength.

B. *Message Exchange and Acknowledgement*

The feature exchange and acknowledgement phases make it very difficult for adversaries to detect the key being agreed on due to the following reasons.

1) In the process of feature exchange, the presence of IDr in the message from the sender to the receiver tells the sensors in the vicinity of the sender who is the intended receiver. The nonce No is used to maintain the freshness of the protocol, i.e., to ensure that the acknowledgement received is in response to its latest transmission.
2) If a malicious entity sends a feature exchange message (by replaying previous exchanges or creating its own Coffer using old physiological features), it will be discarded by any receiver, as the MAC would not match due to the temporal variation of the physiological features.
3) If an adversary obtains the set of legitimate features from the Coffer, it's still hard for the adversary to generate the key K because the features are out of order due to the random permutation in the Coffer creation step but the key K is generated by the ordered features.

VI. Evaluation of OPFKA

In this section, we provide an evaluation of OPFKA by performing a series of experiments on our scheme. We will assess the generated key according to the algorithms proposed in Section IV. The EKG signals are obtained from the PhysioBank database (http://www.physionet.org/physiobank). We will address the following important characteristics of a key: 1) long and random keys; 2) memory storage; 3) communication overhead; 4) energy consumption on communications; 5) distinctiveness; and 6) temporal variance.

1) Long and random keys: The keys to be agreed upon are generated by the sender and the receiver according to the ordered matching features using a hash function. The length and randomness of the agreed keys can therefore be ensured.

2) Memory storage: In our proposed scheme, IDs and IDr take 16 bytes each, the features and the chaff points are 20 bits (2.5 Bytes) each, the index is at most 1 byte, the nonce No is 16 bytes, and the MAC is 16 bytes. Thus the estimated memory cost is

$$2(|IDs| + |IDr|) + 2.5|R| + |I| + |No| + 2|MAC| \quad (1)$$

Fig. 3 illustrates the relationship between the memory storage and the Coffer size. We can see that the major fraction of the memory is taken by the Coffer. A chaff point in the Coffer takes 20 bits. A chaff point in PSKA's vault takes 36 bits [13] in comparison. Therefore we conclude that our scheme has an advantage over PSKA in memory storage.

TABLE II
The Security Strength of OPFKA and PSKA.

Scheme	OPFKA (Bits)	PSKA (Bits)	Scheme	OPFKA (Bits)	PSKA (Bits)
Coffer Size=300, $\|Q\|$=6	48	42	Coffer Size=1000, $\|Q\|$=10	100	80
Coffer Size=300, $\|Q\|$=7	56	47	Coffer Size=1000, $\|Q\|$=11	110	87
Coffer Size=300, $\|Q\|$=8	64	52	Coffer Size=1000, $\|Q\|$=12	120	94
Coffer Size=300, $\|Q\|$=9	72	57	Coffer Size=2000, $\|Q\|$=6	66	59
Coffer Size=300, $\|Q\|$=10	80	62	Coffer Size=2000, $\|Q\|$=7	77	67
Coffer Size=300, $\|Q\|$=11	88	67	Coffer Size=2000, $\|Q\|$=8	88	75
Coffer Size=300, $\|Q\|$=12	96	72	Coffer Size=2000, $\|Q\|$=9	99	83
Coffer Size=600, $\|Q\|$=6	54	48	Coffer Size=2000, $\|Q\|$=10	110	91
Coffer Size=600, $\|Q\|$=7	63	54	Coffer Size=2000, $\|Q\|$=11	121	99
Coffer Size=600, $\|Q\|$=8	72	60	Coffer Size=2000, $\|Q\|$=12	132	107
Coffer Size=600, $\|Q\|$=9	81	66	Coffer Size=5000, $\|Q\|$=6	74	67
Coffer Size=600, $\|Q\|$=10	90	72	Coffer Size=5000, $\|Q\|$=7	86	76
Coffer Size=600, $\|Q\|$=11	99	78	Coffer Size=5000, $\|Q\|$=8	98	85
Coffer Size=600, $\|Q\|$=12	108	84	Coffer Size=5000, $\|Q\|$=9	110	94
Coffer Size=1000, $\|Q\|$=6	60	52	Coffer Size=5000, $\|Q\|$=10	122	103
Coffer Size=1000, $\|Q\|$=7	70	59	Coffer Size=5000, $\|Q\|$=11	134	112
Coffer Size=1000, $\|Q\|$=8	80	66	Coffer Size=5000, $\|Q\|$=12	146	121
Coffer Size=1000, $\|Q\|$=9	90	73			

TABLE III
The Communication overhead of OPFKA and PSKA.

	Coffer Exchange	Feature Ack	Total
OPFKA	$4\|ID\|+2.5\|R\|+\|I\|+\|No\|+\|MAC\|$	$\|MAC\|$	$4\|ID\|+2.5\|R\|+\|I\|+\|No\|+2\|MAC\|$
PSKA	$4\|ID\|+4.5\|R\|+\|No\|+\|MAC\|$	$\|MAC\|$	$4\|ID\|+4.5\|R\|+\|No\|+2\|MAC\|$

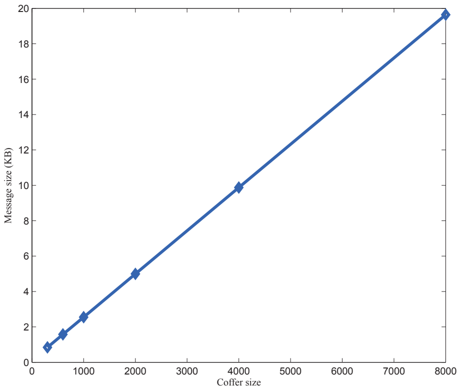

Fig. 3. The relationship between memory storage and Coffer size.

3) Communication overhead: The processes of feature exchange and feature acknowledgement contribute the most to the communication overhead, which is mainly associated with the message size in Algorithm 3 and Algorithm 4. Table III reports the communication overhead of OPFKA and PSKA [13]. Fig. 4 illustrates the relationship between the communication overhead and the Coffer size. We observe that the communication overhead increases along with the Coffer size.

4) Energy consumption due to communications: In this subsection, we take the method proposed in [35] to evaluate the energy consumption resulted from message exchanges. As presented in [36], a Chipcon CC1000 radio used in Crossbow MICA2DOT motes consumes 28.6 μJ and 59.2 μJ to receive and transmit one byte, respectively.

For our OPFKA scheme, the total message size is $2.5*|R|+113$ bytes; hence the energy consumption on transmitting and receiving the messages equals to $(2.5*|R|+113)*(28.6+59.2)\mu J=(0.2195|R|+9.9214)\ mJ$. For PSKA, the total message size is $4.5|R|+112$ bytes; thus the total energy consumed by transmitting and receiving the messages equals to $(4.5|R|+112)*(28.6+59.2)\mu J=(0.3951|R|+9.8336)$ mJ. We summarize the results of energy consumption for both OPFKA and PSKA in Table IV. Fig.5 illustrates the

TABLE IV
Energy Consumption Due to Message Exchange

The schemes	Energy consumption (mJ)
OPFKA	$0.2195\|R\|+9.9214$
PSKA	$0.3951\|R\|+9.8336$

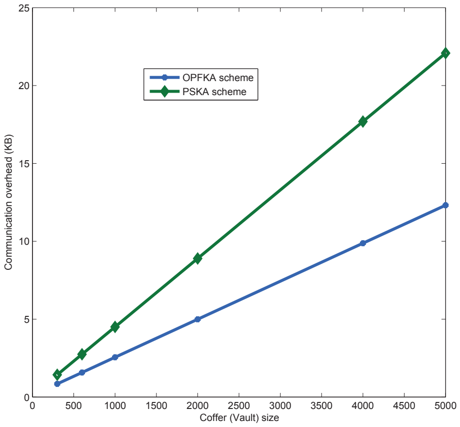

Fig. 4. Comparison between OPFKA and PSKA in terms of the communication overhead.

energy consumption due to communications as a function of the Coffer size $|R|$. Obviously, OPFKA offers a lower energy consumption compared to PSKA.

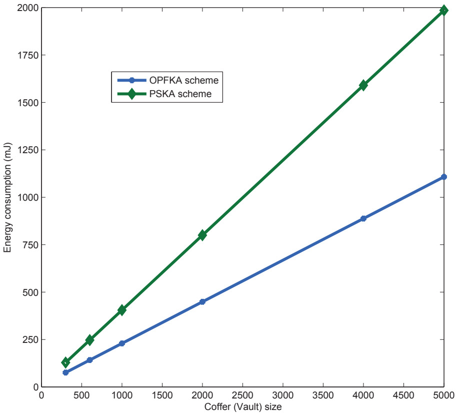

Fig. 5. Energy consumption *vs.* the Coffer (vault) size.

5) Distinctiveness: Distinctiveness is one of the most important criteria for our scheme since our primary purpose is to distinguish the sensors in one BAN from those in another BAN. A false rejection rate (FRR) refers to the probability that two sensors within the same BAN fail to establish a session key. A false acceptance rate (FAR) represents the probability that two sensors at different BANs successfully establish a common key or two sensors in the same BAN asynchronously establish a session key. Since our enhanced FFT method is similar to the feature generation process in PSKA [13], the distinctiveness, efficiency, and temporal variance of the key generated by OPFKA based on the enhanced FFT method will not be discussed here as the analysis is similar to the one conducted in [13]. Thus we focus on the distinctiveness of our IPI method here. The data were collected from 11 subjects (obtained from the PhysioBank database (http://physionet.org/physiobank/database)). Wavelet-based algorithms mentioned in [37] [38] [22] are performed over the ECG signal to detect a heart beat cycle. IPI is the time interval between adjacent R-waves. After quantization, concatenation, and expansion, we obtain the secret features. The red curve in Fig. 6 shows the FAR for two synchronous sensors at different BANs. We can see that the FAR reduces almost to zero when the threshold is greater than 5, which indicates that our scheme can successfully distinguish two BANs. Normally, the threshold should be greater than 10 if the Coffer size is around 2000. The blue curve in Fig. 6 represents the FRR for two synchronous sensors at different BANs. We discuss the case that two asynchronous sensors in the same BAN in next subsection: temporal variance. We can see that the FRR is almost zero if the threshold is less than 12.

6) Temporal Variance: A higher temporal variance implies that the signal has a better randomness, which can reduce the adversary's ability on replay attacks. We mainly experiment on the temporal variance of our IPI method here. As mentioned in [22], we can confidently extract 4 bits from an IPI and the randomness is sufficiently high, which indicates that asynchronous IPIs should not match each other. But in reality, the probability that asynchronous features match each other is not zero. If two sensors establish a key with asynchronous IPI features, a false acceptance occurs. Fig. 7 illustrates the FAR between two asynchronous sensors in the same BAN. The x-axis is the time difference. For example, the value of 20 means that two sensors have a time difference of 20 IPIs. We can see that the FAR is reduced to almost zero if the time difference is greater than 125 IPIs, which is about 2 minutes. Theoretically, if there is any time difference, the FAR should be close to zero for truly random IPIs. Since we use a fixed quantization level, for a human being who has a relatively placid heart beat, a certain level of precision may be lost. Note that our experiment results are consistent with those in [22]. The distribution of the Hamming distances between synchronous features highly matches with the theoretical binomial distribution. But they are not completely the same. If the sensors can dynamically adjust their quantization levels according to historical IPIs or the patient's healthy status, the result should be improved. We leave this to our future research.

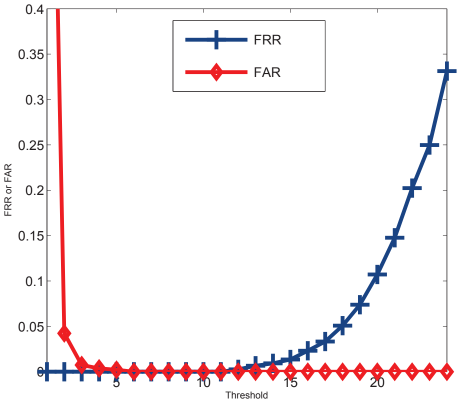

Fig. 6. FAR and FRR between two synchronous sensors in different BAN

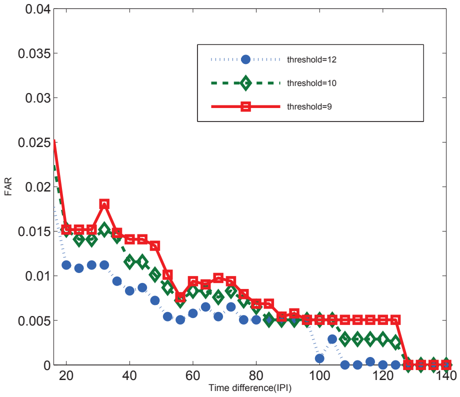

Fig. 7. Time variance.

VII. Conclusion

In this paper, we present a secure and efficient key agreement scheme, namely the Ordered-Physiological-Feature-based Key Agreement (OPFKA), to enable secure inter-sensor communications in a BAN. OPFKA allows two sensors in a BAN to agree upon a symmetric cryptographic key generated from their common features in an authenticated and transparent manner without any keying material pre-distribution or initialization. The security analysis of OPFKA shows that OPFKA meets the design goals of key agreement. We analyze the performance, memory storage, and communication cost of OPFKA and demonstrate that the scheme has low computational cost, low memory storage, and low communication overhead, which indicates that OPFKA is a feasible and efficient approach to secure inter-sensor communications within a BAN.

Acknowledgment

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

References

[1] K. Venkatasubramanian and S. Gupta, "Security for pervasive health monitoring sensor applications," in *the Fourth International Conference on Intelligent Sensing and Information Processing*, 2006, pp. 197–202.

[2] L. Schwiebert, S. Gupta, and J. Weinmann, "Research challenges in wireless networks of biomedical sensors," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, 2001, pp. 151–165.

[3] R. Schmidt, T. Norgall, J. Mörsdorf, J. Bernhard, and T. von der Grün, "Body area network (BAN) - a key infrastructure element for patient-centered medical applications," *Biomedizinische Technik/Biomedical Engineering*, vol. 47, no. s1a, pp. 365–368, 2002.

[4] J. Penders, J. vande Molengraft, L. Brown, B. Grundlehner, B. Gyselinckx, and C. V. Hoof, "Potential and challenges of body area networks for personal health," in *Engineering in Medicine and Biology Society*, 2009, pp. 6569–6572.

[5] K. Venkatasubramanian, S. Gupta, R. Jetley, and P. Jones, "Interoperable medical devices," *IEEE Pulse*, vol. 1, no. 2, pp. 16–27, 2010.

[6] M. Chen, S. Gonzalez, A. Vasilakos, H. Cao, and V. Leung, "Body area networks: a survey," *Mobile Networks and Applications*, vol. 16, no. 2, pp. 171–193, 2011.

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

[8] L. Eschenauer and V. Gligor, "A key-management scheme for distributed sensor networks," in *ACM CCS*, 2002, pp. 41–47.

[9] S. Zhu, S. Setia, and S. Jajodia, "LEAP+: Efficient security mechanisms for large-scale distributed sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 2, no. 4, pp. 500–528, 2006.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] K. Venkatasubramanian, A. Banerjee, and S. Gupta, "EKG-based key agreement in body sensor networks," in *In Proceedings of the 2nd Workshop on Mission Critical Networks*, 2008, pp. 1–6.

[15] C. Tan, H. Wang, S. Zhong, and Q. Li, "Body sensor network security: an identity-based cryptography approach," in *ACM Wisec*, 2008, pp. 148–153.

[16] D. Malan, M. Welsh, and M. Smith, "A public-key infrastructure for key distribution in tinyos based on elliptic curve cryptography," in *IEEE SECON*, 2004, pp. 71–80.

[17] C. Tan, H. Wang, S. Zhong, and Q. Li, "IBE-Lite: a lightweight identity-based cryptography for body sensor networks," *IEEE Transactions on Information Technology in Biomedicine,*, vol. 13, no. 6, pp. 926–932, 2009.

[18] S. Keoh, E. Lupu, and M. Sloman, "Securing body sensor networks: Sensor association and key management," in *IEEE PerCom*, 2009, pp. 1–6.

[19] M. Li, S. Yu, W. Lou, and K. Ren, "Group device pairing based secure sensor association and key management for body area networks," in *IEEE INFOCOM*, 2010, pp. 1–9.

[20] Y. Law, G. Moniava, Z. Gong, P. Hartel, and M. Palaniswami, "Kalwen: A new practical and interoperable key management scheme for body sensor networks," *Security and Communication Networks*, vol. 4, no. 11, pp. 1309–1329, 2011.

[21] M. Li, S. Yu, J. Guttman, W. Lou, and K. Ren, "Secure ad-hoc trust initialization and key management in wireless body area networks," *ACM Transactions on Sensor Networks (TOSN), (To Appear)*, 2012.

[22] F. Xu, Z. Qin, C. Tan, B. Wang, and Q. Li, "IMDGuard: Securing implantable medical devices with the external wearable guardian," in *IEEE INFOCOM*, 2011, pp. 1862–1870.

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

[24] 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 *International Conference on Parallel Processing Workshops*, 2003, pp. 432–439.

[25] 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.

[26] A. Juels and M. Sudan, "A fuzzy vault scheme," *Designs, Codes and Cryptography*, vol. 38, no. 2, pp. 237–257, 2006.

[27] U. Uludag, S. Pankanti, and A. Jain, "Fuzzy vault for fingerprints," in *Audio-and Video-Based Biometric Person Authentication*, 2005, pp. 55–71.

[28] E. Reddy and I. Babu, "Authentication using fuzzy vault based on iris textures," in *Proceedings of the 2nd Asia International Conference on Modelling & Simulation (AMS)*, 2008, pp. 361–368.

[29] 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, p. 1901, 2006.

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

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

[32] G. BLAKLEY, "Safeguarding cryptographic keys," in *AFIPS Conference Proceedings*, vol. 48. AFIPS Press, 1979, pp. 313–317.

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

[34] D. Davenport, N. Seidl, J. Moss, M. Patel, A. Batra, J. M. Ho, S. Hosur, J. C. Roh, T. Schmidl, O. Omeni, and A. Wong, "Medwin Mac and Security Proposal - Documentation," IEEE 802.15 WPAN Task Group 6., September 2009.

[35] 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.

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

[37] C. Li, C. Zheng, and C. Tai, "Detection of ecg characteristic points using wavelet transforms," *IEEE Transactions on Biomedical Engineering*, vol. 42, no. 1, pp. 21–28, 1995.

[38] J. Martínez, R. Almeida, S. Olmos, A. Rocha, and P. Laguna, "A wavelet-based ecg delineator: evaluation on standard databases," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 4, pp. 570–581, 2004.