PSKA: Usable and Secure Key Agreement Scheme for Body Area Networks*

Krishna K. Venkatasubramanian[1], Ayan Banerjee[2], and Sandeep K.S. Gupta[2], *Senior Member, IEEE.*

***Abstract*—A Body Area Network (BAN) is a wireless network of health monitoring sensors designed to deliver personalized health-care. Securing inter-sensor communications within BANs is essential for preserving not only the privacy of health data but also for ensuring safety of healthcare-delivery. This paper presents *Physiological Signal based Key Agreement (PSKA), a scheme for enabling secure inter-sensor communication within a BAN in a* usable *(plug-n-play, transparent) manner*. PSKA allows neighboring nodes in a BAN to agree to a symmetric (shared) cryptographic key, in an authenticated manner, using physiological signals obtained from the subject. No initialization or pre-deployment is required; simply deploying sensors in BAN is enough to make them communicate securely. *Contributions* of the paper are as follows: 1) description of PSKA key agreement protocol, 2) analysis of its security characteristics, 3) validation of PSKA utilizing two of the most commonly collected physiological signals *photoplethysmogram (PPG)* and *electrocardiogram (EKG)*, and 4) cost analysis of executing PSKA, based on a prototype VHDL implementation, and its comparison with Diffie-Hellman key agreement protocol. Our analysis and prototyping shows that PSKA is a viable inter-sensor key agreement protocol for BANs.**

***Index Terms*—Body Area Networks, secure communication, physiological signals based key agreement, usable security**

I. Introduction

Body Area Networks (BAN) are a network of wireless medical sensors, deployed on a person, for enabling *pervasive, individualized, and real-time* health management. As BANs deal with personal health data, securing them, especially their communication over the wireless link, is very critical. Lack of adequate security features may not only lead to a breach of patient privacy but also potentially allow adversaries to compromise patient safety by modifying actual data resulting in wrong diagnosis and treatment [28]. Indeed, protecting health data is a legal requirement as per the Health Insurance Portability and Accountability Act (HIPAA) (http://www.hhs.gov/ocr/hipaa/) which mandates that all personally identifiable information in electronic form be protected.

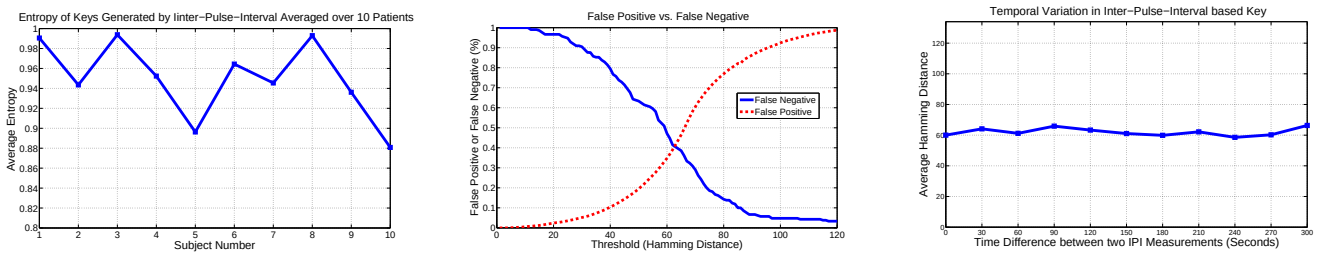
Sensors rely on cryptographic keys to secure their communication. Keys are usually made available to sensors through explicit key distribution protocols. Well known classes of symmetric key distribution techniques such as probabilistic key distribution [8] master-key based distribution [30], require some form of pre-deployment. However, given the progressively increasing size of BANs (networks of size 190-255 nodes have already been proposed [9]); these approaches may potentially involve considerable latency during network initialization/setup or any subsequent adjustments, due to their need for pre-deployment. We believe that for BANs to be useful, they should provide **usable security** — one that is plug-n-play and largely transparent. For example, one should be able to add, remove and adjust the sensors on their BAN as and when required without reconfiguring parts of the network (a very important requirement in mission critical environments) and still have secure communication. In some instances asymmetric cryptosystems such as Diffie-Hellman and its variants have been used to avoid pre-deployment. However, they are prone to man-in-the-middle attacks and need additional authentication mechanisms to be useful.

In this paper we present a novel scheme called *Physiological Signal based Key Agreement (PSKA)*, which utilizes physiological signals for enabling sensors to agree upon a pair-wise symmetric cryptographic key, in an authenticated manner. It requires no *a priori* distribution of keying material, simply deploying the sensors on a subject is enough, thus facilitating secure BAN communication that is **usable**. PSKA is being designed as a part of securing the Ayushman health monitoring system [29] being developed at the IMPACT lab at Arizona State University. The use of PSKA can eliminate the need for explicit key distribution in the Ayushman BAN, allowing constituent sensors to agree upon keys, as needed. The idea of using physiological signals based features for key agreement comes from the observation that the human body is dynamic and complex and the physiological state of a subject is quite unique at a given time. Generally speaking, PSKA works as follows: 1) the sensors which want to securely communicate, measure the physiological signal for a predefined duration of time, 2) one of the two sensors (sender) generates an arbitrary key, 3) it then hides this key using features derived from measured physiological signal, 4) the hidden key is then communicated to the other sensor (receiver), which uses its own features to un-hide the key. The key hiding and unhiding processes are based on the *fuzzy vault* cryptographic primitive, first suggested in [11]. PSKA meets the **design goals** suggested in [20] [26] when physiological signals are used as a basis for key agreement, namely: 1) *Length and Randomness*: The keys agreed upon are long and random to prevent brute-forcing; 2) *Low Latency*: The duration of physiological signal capture required is minimal; 3) *Distinctiveness*: Knowing the feature derived from the current value of the physiological signal of one subject will not provide significant advantage

*To appear in IEEE Transactions on Information Technology in BioMedicine . All rights reserved.

[1]Author is affiliated with University of Pennsylvania, Philadelphia, PA 19104, USA. Email: vkris@cis.upenn.edu

[2] The authors are affiliated with the IMPACT Lab (`impact.asu.edu`), Arizona State University, Tempe, AZ 85287, USA. Email: {abanerj3,sandeep.gupta}@asu.edu.



(a) (b) (c)

Fig. 1. **a) Average Entropy of Keys generated from IPI, b) False Positive and False Negative while using IPI for Authentication, c) Time Variance of IPI-based Authentication Keys.**

in guessing the keys being agreed by sensors on another subject. An important characteristic of distinctiveness is that it authenticates the communicating sensors by ensuring that only sensors on the same BAN can agree on a shared key; and 4) *Temporal Variance*: Knowing the physiological signals at any time will not provide significant advantage in knowing the keys agreed upon in future executions of the scheme.

At this point, we would like to *differentiate the proposed technique from biometrics systems* such as [18] [21] [25]. PSKA is fundamentally different from traditional biometric systems because the physiological signal based features used by it show both distinctiveness and temporal variance properties (see Section VI). Biometric systems only posses the former, and actually depend on not possessing the latter. A biometric template created using a person's fingerprint, for example, is used for all subsequent authentication attempts made by the subject. The success of the authentication is based on the closeness of the match between fingerprint sample, presented by the subject, and the original template. This model of operation allows arbitrary time to pass between two biometric measurements, as the biometric template itself is invariant. However, in the case of PSKA physiological signal features are time variant. Key un-hiding process at the receiver therefore requires a (loosely) synchronized measurement of physiological signals with the sender, in order to be successful. An attacker knowing the physiological signal based features at any given time, has no advantage in compromising any past or future key agreements between sensors, a property that biometric templates do not posses. The **contributions** of this paper are threefold: 1) a scheme for authenticated pair-wise key agreement between two nodes in BANs, i.e. PSKA (Section IV), 2) analysis of PSKA security properties (Section V), 3) validation of PSKA, using actual data from two of the most commonly collected physiological signals - Photoplethysmogram (PPG) and Electrocardiogram (EKG), based on the aforementioned design goals (Section VI), and 4) prototype implementation of PSKA and its comparison to Diffie-Hellman (DH) key agreement. We choose DH as it does not require pre-deployment and also to get a perspective on PSKA's potential costs (Section VII).

II. Preliminaries

In this section we present some preliminaries regarding the use of physiological signals for key agreement, and in the process motivate our approach.

A. Using Inter-Pulse-Interval

One of the earliest works which suggested the use of physiological signals for securing inter-sensor communication was presented in [4] [28]. The papers however do not suggest any particular physiological signals which can be used. Building upon this initial idea, the authors in [20] proposed the use of Inter-Pulse-Interval (IPI) to generate cryptographic keys. The advantage of using IPI is that it can be derived from multiple sources namely Photoplethysmogram (PPG) and Electrocardiogram (EKG) time series by measuring the time difference between the peaks in the EKG/PPG signal. One can envision two sensors in a BAN designed to measure EKG and PPG respectively being able to communicate securely with one another using IPI based keys. The IPI-based key generation process works as follows: 1) the sensors first measure EKG and PPG signals in a synchronized manner; 2) they then generate a series of IPI values from their respective data; and 3) they take 67 contiguous IPI values from a particular start-point and encode them into 128 bit to form the key_{ekg} and key_{ppg} at each of the sensor which can then be used for secure communication between them. We conducted our own experiments of IPI based key generation using EKG and PPG data of ten subjects collected from the PhysioBank database (http://www.physionet.org/physiobank/). We found that even though the keys are long and random (see Figure 1(a)), the average hamming distance between key_{ekg} and key_{ppg} for the same subject is $\sim$60 bits, and $\sim$65 bits for two different subjects. The scheme proposed in [20] is therefore at best suited for *authentication purposes only* (as identical keys would be needed if the key is to be used for encryption and message authentication code). This can be easily done by performing the following check $Hamm(key_{ekg}, key_{ppg}) < Threshold$, where $Hamm()$ is the function for computing the Hamming distance, and $Threshold$ is the Hamming distance threshold between the two keys. Figure 1(b) shows the false positive[1] and false negative[2] rates of IPI based authentication, and we can see the results are not encouraging. We believe the primary reason for this difference is the *topographic specificity* of the human body — physiological signals measured from different areas of the body appear to have similar trends (high correlation) but not the exact same values. As a result, the information symbols in the keys get re-ordered leading to translational

[1]when key_{ekg} and key_{ppg} of two different people are less than $Threshold$

[2]key_{ekg} and key_{ppg} for the same person are above $Threshold$

and rotational errors [11] which produce drastically different values. Additionally, IPI requires quantizing 67 IPI values for generating a key, which means about 30 seconds of EKG/PPG measurements are required (as an EKG/PPG peak is generate every 300-500 msec) before keys can be generated, which is not adequate for real-time requirements of the BAN. Also, the keys generated from IPI lack temporal variance. This is important, as the knowledge of the key_{ekg} or key_{ppg} at any given time should not allow adversaries to authenticate successfully at a later time. It can be seen from Figure 1(c), the average Hamming distance between the keys generated for a person over time remains around 60 bits. This means that if the current value of a key_{ekg} or key_{ppg} is known and the threshold used is 60 bits (based on the Figure 1(b)) or above, the probability of using it to authenticate during any subsequent authentication attempts is close to one.

B. Our Approach

We therefore take a different approach: instead of trying to generate keys from physiological signal measurements, we use them to facilitate key agreement. This is because given the dynamic nature of the human body the chances of physiological signal features being exactly identical is low. Further, we do this by processing physiological signals in frequency-domain, instead of time-domain. Frequency domain processing has many advantages: 1) Frequency components of physiological signals, at any given time, have many more common values, compared to time-domain values of physiological signals, irrespective of where they are measured on the body; 2) The sample of physiological signal required for key agreement is much smaller, and 3) the level of synchronization required for measuring the physiological signals at the sensors is not very strict[3]. The aim of this paper is to show that physiological signals can be used for establishing symmetric keys between sensors in the BAN, and validate the results.

III. System Model

We assume a Body Area Network (BAN) to be a network of physiological and environmental monitoring *sensing nodes* which are worn and/or implanted on a *subject* or *individual*. The sensing nodes collect health and contextual data at regular intervals and forward it over a multi-hop network to a highly capable *sink* node for further processing. A typical sensing node (or *sensor*, as we shall call them from now on) consists of a sensing element, analog-to-digital converter, wireless communication stack, processor and memory. We assume that the sensors *communicate wirelessly*, as wires running between sensors in a BAN will make it obtrusive. The wireless medium is however not trusted. All sensors are assumed to be able to measure the appropriate physiological signals. Already,

[3]Our experiments show successful key agreement even with a 1 second difference in the start-times of the measurement of physiological signals using frequency domain features (see Section VI for more on this). As a sensor needs some form of synchronization to communicate in the first place, we can use the same primitives to achieve the synchronization required for PSKA. Some of the prominent solutions proposed for time synchronization for sensor networks are [6] [7] [24], all of which achieve microsecond-level synchronization, which is more than sufficient for PSKA.

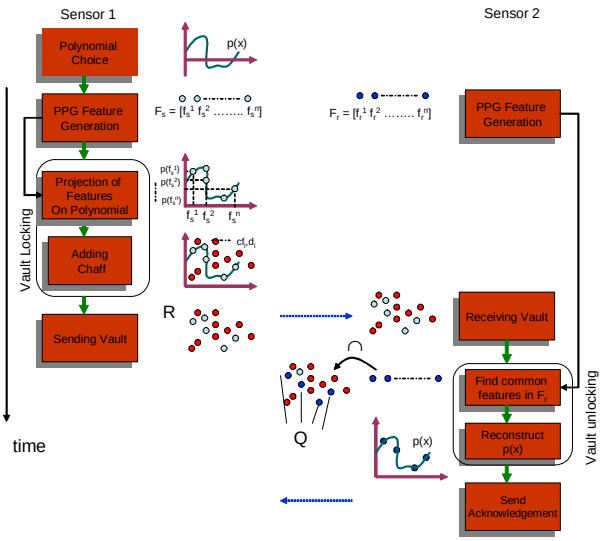


Fig. 2. **The agreement of keys between sensors in a BAN using Physiological Signal based Key Agreement.**

physiological monitoring sensors are becoming multi-modal and are able to sense multiple types of stimuli [17]. Any entity not in contact with the subject cannot measure physiological signals from the subject. We assume that only legitimate sensors are in contact with the body. Further, we assume that malicious entities cannot introduce nor compromise sensors within the BAN without being detected as anything worn is mostly under supervision of the host (if the host is not capable then under the supervision of the caretaker). Therefore, the threats faced by a BAN are primarily from adversaries who can eavesdrop on all the traffic within the BAN, inject messages, replay old messages, and spoof sensor identities. Adversaries may also try to use the physiological signal data obtained from other people to break the key distribution process. Note that, in this work, we focus solely on securing inter-sensor communication within the BAN. Communication from the sink onwards can utilize conventional security schemes such as Secure Socket Layer (SSL) given the considerable capabilities of the entities involved. Finally, we do not consider Denial of Service (DoS) attacks such as jamming, electromagnetic interference or battery depletion attacks in this work.

IV. Physiological Signal based Key Agreement

The purpose of *Physiological Signal based Key Agreement (PSKA)* is to *facilitate secure inter-sensor communication*s between two sensors by enabling them to agree upon a pair-wise symmetric key, using physiological signal based features. The key agreement process works as follows: one of the two sensors (sender) generates a random symmetric key which it then hides using a feature vector obtained from the physiological signal. This hidden key is sent over to the other sensor (receiver) which uses its own version of the feature vector and obtains the random key after compensating for the differences between its feature vector and the one used by the sender. In our previous work [4], we proposed the use of simple error correction scheme such as majority decoding [12], as the compensatory mechanism, to arrive at a common

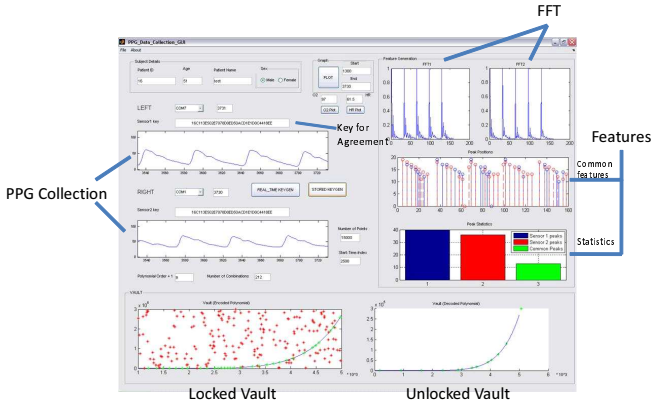


Fig. 3. **Screen Shot of execution of PPG based PSKA for data collected from same person.**

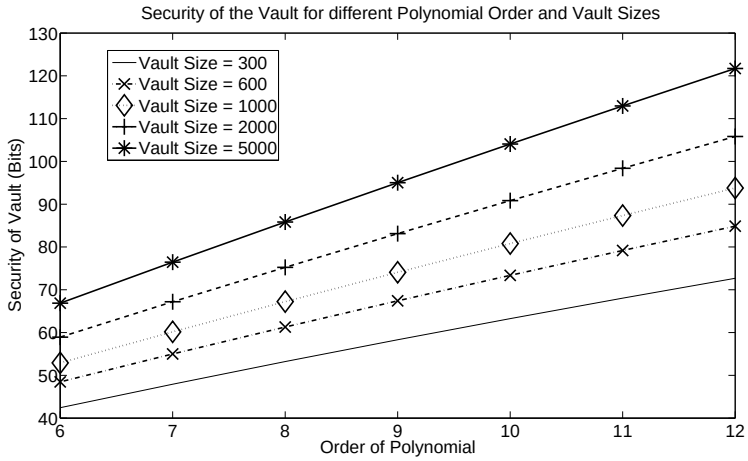


Fig. 4. **Vault Strength w.r.t. Polynomial Orders for Different Vault Sizes.**

key at the receiver. The inspiration behind the idea was the observation that each measurement of a physiological signal was independent of others; any difference in their measured values could be modeled as communication error. An inherent problem with using this approach is although it can correct the presence of a few differences in feature vectors, it cannot handle re-ordering of, or presence of additional features (in one of the sensors) in the feature vector [11]. We address this drawback by proposing the use of a cryptographic construct called *Fuzzy Vault* [11].

A. Fuzzy Vault

The fuzzy vault scheme first proposed in [11] is designed to lock (hide) a secret (S) in a construct called a *vault* using a set of values A. Once the vault has been locked, it can be unlocked only with another set of values B which has a *significant* number of values in common with set A. The construction and locking of the vault is accomplished by: 1) generating a v^{th} order polynomial p over the variable x that encodes the secret S, 2) computing the value of the polynomial at different values of x from set A and creating a set $R = \{a_i, p(a_i)\}$, where $1 \leq i \leq |A|$, and 3) adding randomly generated set of points called *chaff* to R which do not lie on the polynomial. Once the vault is constructed, unlocking it based on the set B is done by constructing a set $Q = \{(u, v)|(u, v) \in R, u \in B\}$. The unlock process is possible only if Q has a significant number of legitimate (non chaff) points which are on the polynomial [11]. We can map this scheme onto PSKA by setting the features obtained at the sender to set A, those obtained at the receiver to set B, and generating a polynomial whose coefficients form the secret key that needs to be agreed upon.

Consider the following example which illustrates the operation of the fuzzy vault. Let the polynomial be $p(x) = x + 1$, set $A = \{1, 2, 3\}$ and $B = \{1, 3, 4\}$, then the vault R created by computing the polynomial's value at each point in A is $R = \{(1, 2)(2, 3)(3, 4)(4, 7)(6, 9)(7, 12)(8, 5)\}$. The last four points are the chaff points which do not fall on the polynomial. To unlock the vault the set Q is constructed, where $Q = \{(1, 2)(3, 4)(4, 7)\}$. As the set Q has two points on the polynomial, we can use it to easily reconstruct the first order polynomial and thus unlock the secret.

B. Vault Locking & Unlocking in PSKA

The use of polynomials ensures that the sets A and B need not have any order to them as long as they have a significant number of common values. The presence of the chaff points adds security to the vault, hides legitimate points and the actual polynomial. Unless the adversary knows a large number of points on the polynomial, it cannot reconstruct the polynomial. In this section, we show how fuzzy vault scheme for key agreement with PSKA.

We use the term *sender* for the sensor which creates the vault and locks it, and the *receiver* for the sensor that unlocks the vault to access the secret key. The key agreement occurs as follows:

Feature Generation: First, both the sender and the receiver obtain physiological signal based features. This is a four step process: 1) Both the sensors sample the physiological signal in a loosely synchronized manner, at a specific sampling rate for a fixed duration; 2) The samples are divided into windows and a Fast Fourier Transform (FFT) is then performed on each of these parts; 3) The FFT coefficients of each of the overlapping windows (a pre-defined number of contiguous signal time-series points) are then passed through a peak detection function (a simple local maxima detector) which returns a tuple of the form $< k_x^i, k_y^i >$, where k_x^i is the FFT point at which peak is observed (its the peak location on x-axis, also called *peak-index*), k_y^i is its corresponding FFT coefficient values (magnitude of the peak, or *peak-value*), and i is the index of the peaks. The number of peaks observed by a sensor vary upon situation; 4) Each of these peak-index (k_x) and peak-value (k_y) pairs are quantized and converted into a binary string and concatenated ($[k_x|k_y]$) to form a *feature*. Individual features obtained from a single measurement are grouped together to form a feature vector $F_D = \{f_D^1, f_D^2, ...f_D^N\}$, where $f_D^l = [k_x^l|k_y^l]$, D is either the sender (s) or receiver (r) node, and N is the size of the feature vector, i.e., number of indexes where peaks were observed. The values of the different parameters used for feature generation are dependent upon physiological signal used and they need to

TABLE I
PSKA Feature Generation Parameters

Signal	Parameters	Values
PPG	Sampling	60 Hz
	Sampling Duration	12.8 secs
	FFT	256 points, 5 windows[a]
EKG	Sampling	125 Hz
	Sampling Duration	4 secs
	FFT	256 points, 2 windows[b]
PPG/EKG	Peak Value Quantization	5 bits
	Peak Index Quantization	8 bits
	Feature Length	13 bits

[a] First 32 points per window were concatenated together for peak based feature generation
[b] First 128 points per window were concatenated together for peak based feature generation

TABLE II
PPG/EKG Feature Statistics

Signal	Parameters	Values
PPG	Number of Iterations	113, 1.6 sec apart
	Avg. Feature Vector Length	∼30
	Avg. # Common Features (Same Subject)	12
	Mode # of Features (Same Subject)	14.8
	Avg. # Common Features (Different Subjects)	2
	Mode # of Features (Different Subjects)	0.8
EKG	Number of Iterations	180, 4 sec apart
	Avg. Feature Vector Length	∼87
	Avg. # Common Features (Same Subject)	24.7
	Mode # of Features (Same Subject)	25.2
	Avg. # Common Features (Different Subjects)	9.5
	Mode # of Features (Different Subjects)	8.1

be tuned during deployment. We chose FFT peaks as features for two reasons -1) they are simple to detect, and 2) they characterize a subject's physiology very well. They are ideal for distinguishing between sensors which are in the same BAN or different BANs, thus providing an efficient authentication mechanism and a basis for key agreement (see Section VI for more details). At the end of the feature generation process, the sender and receiver possess feature vectors of the form $F_s = f_s^1, f_s^2, ...f_s^N$ and $F_r = f_r^1, f_r^2, ...f_r^N$, respectively.

Polynomial Choice: Once the features are generated, the sender generates a v^{th} order polynomial of the form $p(x) = c_v x^v + c_{v-1}x^{v-1} + ... + c_0$, where the value of the coefficients (c_is) are selected randomly (using a pseudo random number generator, for example). The order of the polynomial (v) used within the BAN is not a secret and known to all sensors in the network. The coefficients, concatenated together, form the secret key that the sender wants to communicate to the receiver ($Key = c_v|c_{v-1}|...|c_0$). We have set the length of this Key to be 128 bits (longer keys can easily be used), and depending upon the order of the polynomial used, the coefficients are obtained by dividing the Key accordingly.

Vault Creation: With the polynomial and feature vector available, the sender now creates the fuzzy vault, by computing the set $P = \{f_s^i, p(f_s^i)\}$, where $f_s^i \in F_s$, and $1 \leq i \leq N$. It also computes a much larger set of M random chaff points of the form $C = \{cf_j, d_j\}$, where $cf_j \notin F_s$, $d_j \neq p(cf_j)$, and $1 \leq j \leq M$. Each chaff point, cf_i, is within the same range (0-2^{13}) as that of the features. Therefore, 2^{13} is the bound for the total number of points in the vault ($|R|$), which is equal to $|M| + |N|$. We refer to the cardinality of the set R as the *vault size*.

Vault Locking: The sender then randomly permutes the values in the vault $R = RandPermute(P \cup C)$, to ensure that the chaff points and the legitimate points are indistinguishable. The cardinality of the set C can vary with respect to the level of security needed. The larger the set C, the more difficult it is to break the vault. Section V discusses the relationship between the vault's size and its security in more detail.

Vault Exchange: The sender communicates the vault R to the receiver using the following message: $Sender \rightarrow Receiver$: IDs, IDr, R, No, $MAC(Key, R|No|IDs)$. Here, IDs and IDr are the ids of the sender and receiver, respectively, No is a nonce (unique random number) for transaction freshness; MAC is a message authentication code (e.g. HMAC-SHA1); and, the key (Key) used is the one that is being locked in the vault.

Vault Unlocking: The receiver upon receiving the vault R first computes the set Q, where $Q = \{(b, c)|(b, c) \in R, b \in F_r\}$. It then tries to reconstruct the polynomial p based on the points in Q using the Lagrangian interpolation (as suggested in [25]), according to which, the knowledge of $v + 1$ points $\{(x_0, y_0)(x_1, y_1), ..., (x_v, y_v)\}$ on a polynomial allows the reconstruction of a v^{th} order polynomial by performing the following linear combination: $p'(x) = \sum_{j=0}^{v} y_j d_j(x)$, where $d_j(x) = \prod_{i \neq j, i=0}^{i=v} (x - x_i)/(x_j - x_i)$. For the receiver to be successfully able to unlock the vault, the condition $|Q| > v$ should hold. It then takes $v + 1$ points (from Q) at a time and tries to unlock the vault. The coefficients of the resulting polynomial are then used to verify the MAC. This not only confirms the correctness of the unlocking process, but also authenticates the sender to the receiver (confirms that the sender is on the same BAN as the receiver). This is because of the distinctiveness and temporal variance property of the physiological signal features which ensures: 1) the features generated from physiological signals for PSKA are drastically different for two different people, and 2) the old vaults cannot be replayed as the features would have changed by that time, and cannot be unlocked (see Section VI for more details).

Vault Acknowledgment: If unlocking was successful, the receiver then sends a reply back to the sender to inform it of its correct unlocking of the vault using the following message: $Receiver \rightarrow Sender : MAC(Key, No|IDs|IDr)$. The symbols have the same meaning as above. The successful verification of the acknowledgment authenticates the receiver to the sender. This is because only a node on the BAN (receiver), which measured the same physiological signal at the same time as itself, could have unlocked the vault given the distinctiveness and temporal variance properties of the physiological signal based features.

C. Discussion

Figure 2 shows the feature generation process. We refer to the execution of these seven steps as an *iteration* of PSKA. The random key (Key) generated in the first step can be used to enable confidential, authenticated and integrity protected communication between sensors in a *plug-n-play*

manner making BANs more usable. None of the traditional key distribution schemes [8] [14] [30] nor physiological signal based approaches [20] consider this property. Further, with PSKA, no random key or physiological features are ever re-used. This ensures any knowledge of past keys or physiological features (due to their temporal variance property as we see in Section VI) of a subject cannot be used for subverting the vault.

Once the PSKA is successfully executed, both the sender and receiver have the common random Key. In order to use this key for future data communication we derive new data communication keys, as suggested in as in [19], one at the sender $K's = f(Key, 1)$ and another at the receiver $K'r = f(Key, 2)$ to minimize the effects of cryptographic attacks on secured data communication, here f is a pesudorandom function. Finally, the one-hop security provided by PSKA can be easily extended to multi-hop end-to-end communication. As soon as the sensors are deployed, they can execute PSKA with each of their neighbors and use the appropriate key for forwarding messages depending upon which node is the next hop. Figure 3 shows the screen shot of our implementation of PPG-based PSKA in Matlab which can perform key agreement based on data collected in real-time from a subject.

V. Security of PSKA

Security issues in PSKA primarily arise due to its vault exchange requirement and reply it to authenticate oneself and join the BAN. An eavesdropper can record the vault and try to construct the hidden polynomial (key) from it. In this section we discuss the security implications of the two principal aspects of PSKA: the vault and its exchange.

A. Vault Security

The use of the fuzzy vault construct in PSKA 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 the PSKA scheme is based on the difficulty of polynomial reconstruction. The hiding of the legitimate feature points among much larger number of the bogus chaff points, whose values are in the same range, makes the job of identifying the legitimate points very difficult. An adversary who does not know any legitimate points (as it cannot measure the relevant physiological signals from the host's body) has to try out each of the $v + 1$ points in set R to be able to arrive at the correct polynomial. By the same account, the more the number of features an entity is aware of, the easier it is to re-construct the hidden polynomial. An example is the receiver who has to try out $v + 1$ points from the set Q to arrive at the correct polynomial [4].

Figure 4 shows the strength of the vault for different values of polynomial order used for different number of chaff points. The strength of the vault is determined by the number of combinations an adversary has to try in order to find $v + 1$ legitimate points in the vault. For ease of understanding, we represent this computation requirement in terms of its equivalence to brute-forcing a key of a particular length (bits). As expected, increasing the number of chaff points increases the security provided by the vault. Higher the order of the polynomial, the more common features we need to find and therefore higher the security. Note that PSKA guarantees successful unlocking of the vault as long as the number of common features in Q are greater than v. By choosing the order of the polynomial to a value $|F_s \cap F'_r| < v < |F_s \cap F_r|$, where $|F_s \cap F'_r|$ are the number of common features between feature vectors of two different individuals and $|F_s \cap F_r|$ are the number of common features between feature vectors for the same individual, we can ensure successful vault unlocking for the receiver but not for the adversary. Of course, this will work only if there is a discernible difference in the number of features between physiological signals collected from the same individual and different individuals.

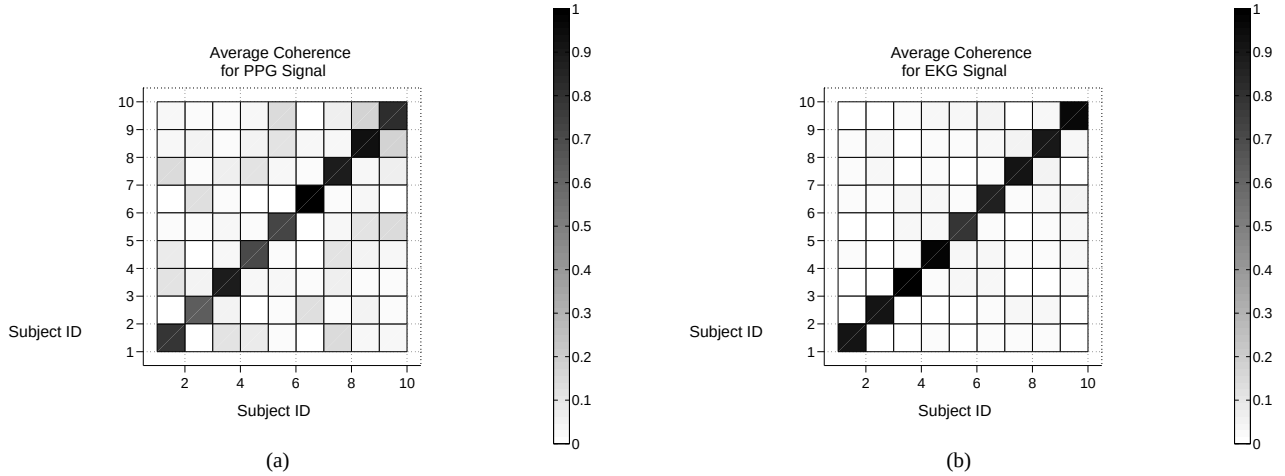
From Figure 4, it can be seen that choosing vault parameters is a balancing act between multiple criteria: number of chaff points, feature length and number of common features. We recommend that the number of chaff points should be as large as possible compared to the total feature length as can be tolerated by the system. A vault with an overwhelming number of chaff points is difficult to compromise by pure brute-force. This along with an appropriate polynomial order depending upon the common feature statistics will ensure that vault is secure.

B. Exchange Security

The vault exchange and acknowledgment phases make it very difficult for adversaries to know the key being agreed upon, because of the following reasons:

- The presence of IDr in the vault exchange message tells the sensors in the vicinity of the sender, who the intended receiver is. 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.
- If a malicious entity sends a vault exchange message (by replaying previous exchanges or creating its own vault using old physiological features) it will be discarded by any receiver as the MAC would not match due to difference in the Key used given the temporal variation of the physiological features.
- The vault has a many orders of magnitude more number of chaff points compared to the legitimate points (for example, 1000 chaff points to 30 legitimate feature points) which makes it difficult for adversaries to know which points are legitimate and which are not (as discussed above). A malicious entity which cannot unlock the vault cannot send a valid acknowledgement as it would have to generate a valid MAC without the Key.
- A malicious entity trying to mount a man-in-the-middle attack has to be aware of the physiological signal features

[4]It needs to be pointed out that recent years have seen the development of remote physiological monitoring technologies [5] [10]. Such technologies in their current state are designed to work in a controlled environment and are susceptible to interferences by objects in the environment, directionality, and motion artifacts. The use of such a technology to measure the physiological signals in open environments and compromise PSKA is an open issue and merits further investigation.



(a) (b)

Fig. 5. **Average Coherence computed over all time stamps for all subjects a) PPG , b) EKG.**

being used. Without them, any modification of the vault during exchange would be caught as none of the $\binom{|Q|}{v+1}$ keys unlocked by the receiver will verify the MAC.

VI. Performance of PSKA

We validated the PSKA approach using two of the most common physiological signals that can be collected from a person — photoplethysmogram (PPG) and electrocardiogram (EKG). The former is a measure of the volumetric change in the distension of arteries due to the perfusion of blood through them during a cardiac cycle, while the latter is the representation of a subject's cardiac cycle generated by the electrical activity of the heart. The basis for validation was the meeting of our design goals set forth in Section I. We begin by discussing the data collection procedure for our experiments, followed by the analysis of the performance the two physiological signals when they are used with PSKA.

A. Data Collection and Feature Extraction

The PPG data utilized for our analysis was collected from ten volunteer subjects in the IMPACT Lab. We used Smith Medical pulse oximeter boards (specifications can be found at http://www.smithsoem.com/applications/oxiboards.htm) to collect the data from the volunteers. The volunteers were asked to sit upright with their hands firmly placed on a desk; an oximeter sensor was placed on the index finger of each hand. We assume, for the purposes of our experiment, that the two communicating sensors utilize signals measured from each finger. Data was collected for about 5 minutes from each subject at a sampling rate of 60Hz. The EKG data (for ten subjects) on the other hand was obtained from the PhysioBank database (http://www.physionet.org/physiobank). EKG data was obtained from two leads on every person. We assume that the two communicating sensors utilize signals from each lead for key agreement. About 15 minutes of data was downloaded for our analysis, with the signals sampled at 125Hz. The PSKA implementation and analysis was done using Matlab. Table I shows the parameters used in the feature generation process.

B. Results

In this section we look at the results obtained for PSKA when used with PPG and EKG signals as the physiological signal of choice. Our aims are two fold: 1) to show that the results follow the design goals set forth earlier, and 2) to perform a comparative analysis of PPG-based and EKG-based PSKA. However, we first provide an empirical argument for why PSKA works by evaluating the correlation between the two EKG and PPG signals in frequency domain.

1) Coherence: PSKA depends upon the high degree of correlation between loosely synchronized physiological (EKG/PPG) signals of the same subject to enable key agreement between sensors deployed on a subject. In order to validate this property, we performed a rigorous study on the frequency domain correlation of EKG and PPG signals. In this regard we computed the *coherence* of time synchronized EKG and PPG signals. Coherence is a measure of the cross correlation between the FFT coefficients of two signals. Formally, coherence of two time series signals x and y can be calculated as $\frac{Cxy^2}{(Cxx \times Cyy)}$, where Cxy is the cross-correlation of the FFT coefficients of x and y, Cxx is the autocorrelation of the FFT coefficients of the signal x and Cyy is the autocorrelation of the FFT coefficients of signal y. We used coherence because it provides us with a measure of commonality in the frequency domain characteristics of the two EKG and PPG signals. We computed the coherence between the PPG and EKG signals for all combination of subjects, and derived a metric from it to represent the overall trend called *average coherence*. Average coherence is the normalized area under the coherence curve of two EKG and PPG signals, averaged over all possible time-stamps ((over 100 time stamps for both EKG and PPG)). Figure 5(a) and (b) shows the average coherence results for PPG and EKG signals. It can be seen that, for the same subject (diagonal values), the metric has a very high value indicating that EKG and PPG signals have high frequency domain commonality. However, for different subject the metric has very low values indicating low commonality. The results thus confirms that frequency domain components of PPG and EKG signals are highly correlated and ideal for executing PSKA.

2) Design Goal Compliance: In this section, we show how PSKA meets our design goals set forth earlier.

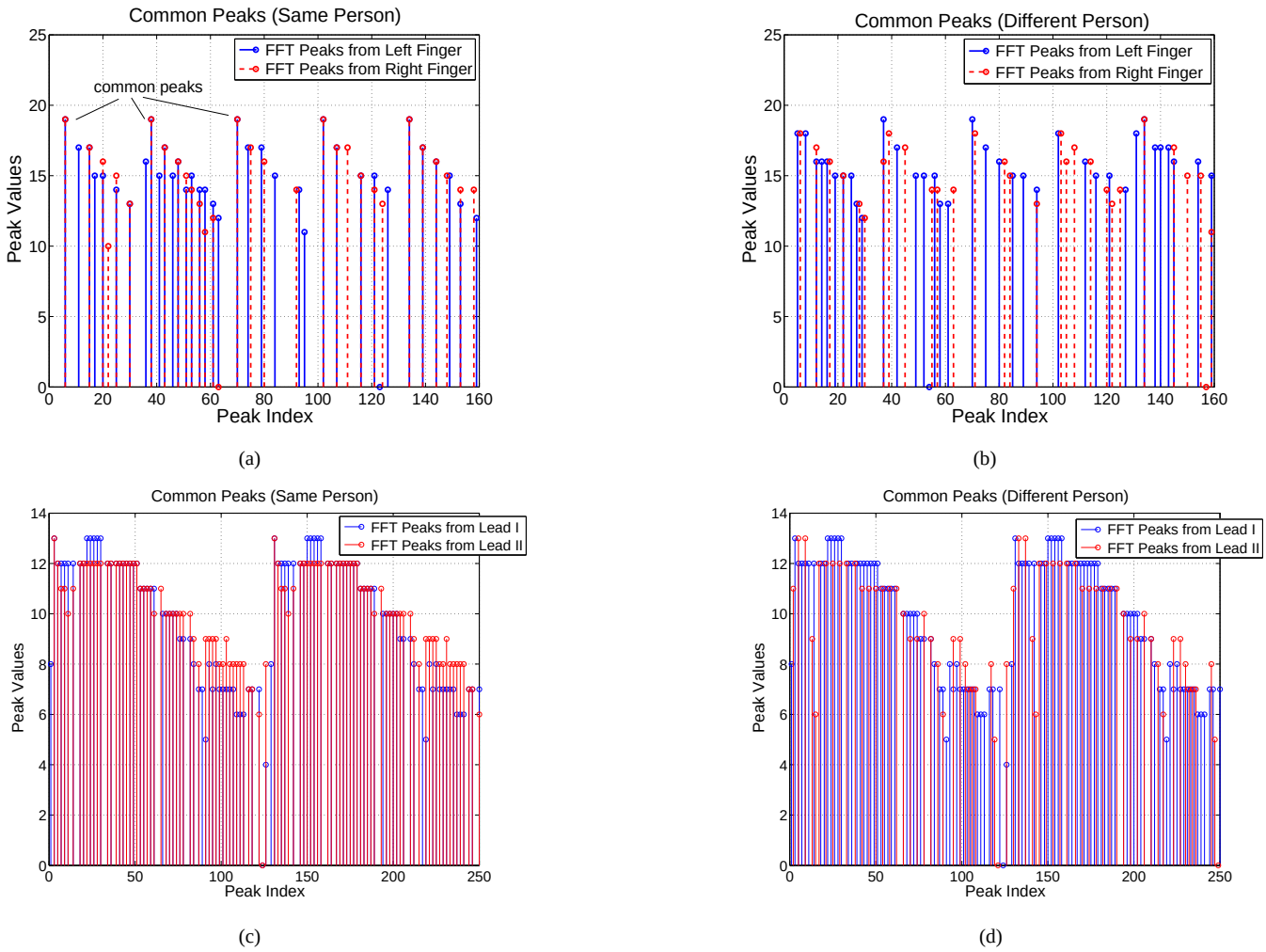


Fig. 6. **FFT Peaks (peak index vs. peak values) for: a) PPG - same subject (Total Common Peaks: 12), b) PPG - different subjects (Total Common Peaks: 2), c) EKG - same subject (Total Common Peaks: 40), d) EKG - different subject (Total Common Peaks: 8).**

Long and Random Keys: The keys to be agreed upon are generated by the sender in the form of polynomial coefficients using a pseudorandom number generator. The length and randomness of the keys agreed can therefore be ensured.

Low Latency: The duration of sampling needed for secure key agreement depends upon the physiological signal used. With PPG (sampled that 60Hz) our best results were obtained with 12.8 seconds of data. While for EKG (sampled at 125Hz) this dropped down to 4 seconds of data. In general we observe that the more detailed the data available, the lower the latency. Both of these signals outperform IPI, which requires about 30 seconds of data.

Distinctiveness: An important requirement of PSKA is that the physiological signals can distinguish people. This is important because we do not want the vault created by a sensor in one BAN to be unlocked by another sensor located on another subject (either accidentally or maliciously) based on features generated from its measurements. We therefore want to make sure the number of common features for sensors on the same subject be "significantly" more than the number of common features for sensors on the different subject. Our definition of significant is dependent upon the polynomial order v used. Figure 6 shows peak values versus peak index graph for PPG and EKG obtained from the same individual and different individuals. The lines which are completely super-imposed are the common features. We found that the higher the correlation between the FFT of PPG and EKG signals at two sensors, the larger is the number of peaks values and indices they have in common in their respective FFTs. It can be seen from Figure 6, sensors on the same person have a higher number of super-imposed lines (less blue bars are visible as they are hidden by the red ones) compared to sensors on two different people. The lack of correlation between the FFTs of the physiological signals measured at two different subjects is due to the difference in physiological signature of each person at any given time. Table II shows the statistics observed for the features when PSKA was executed based on PPG and EKG signals. We can see that, the difference between the number of common features between two sensors on the same subject and two sensors on two different subjects, is significant. Therefore, given the statistic on differences in the number of common features, we can now decide the possible values for v. The polynomial order has to be such that we minimize the *false positives*, i.e. the number of times the common features between two people exceeds it, as well as minimize the *false negatives*, i.e. the number of times the

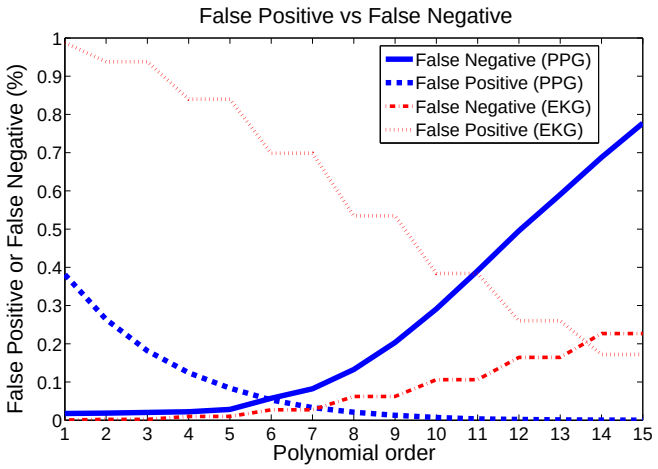


Fig. 7. **False Positive vs. False Negative Rate for EKG and PPG.**

common features for the same subject is below it. Figure 7 shows the percentage of false positives and false negatives when PPG and EKG is used with PSKA for different orders of polynomials. For PPG, the false positive and the false negative rates are minimized when the order of polynomial used is 6, while for EKG it is 14. From these results we can say that using features derived from the PPG and EKG signal to generate a vault, does not give any significant advantage to an adversary who uses features derived from another subject, provided an appropriate polynomial order is chosen.

Temporal Variance: We have shown EKG and PPG signal features are distinctive between two people. We now want to establish their temporal variance. Figures 8(a) and (b) show the temporal variance of the PPG and the EKG signal features, respectively. The x-axis of the graph is the time difference between the PPG and EKG measurement start-times of two iterations of PSKA, the y-axis is the polynomial order used, while the z-axis shows the *average violations*, which is the percentage of times when the number of common features between the first and second iterations of PSKA are greater than the order of the polynomial used[5]. As expected, when the time difference between the two iterations of PSKA for both PPG and EKG is very close violations are very high; since the feature values in both the iterations are very similar. However, as the time difference increases the percentage of violations falls drastically for PPG, reaching almost zero within the first few seconds for polynomials of order 9 and above. For EKG with its higher number of common peaks between two different people, the fall in violations is more gradual and does not fall significantly before 14th order polynomials and about 600 seconds time difference between two PSKA iterations. Finally, as expected for both EKG and PPG the higher the order of polynomial, the more the number of common features needed and therefore lower the chance of getting $v+1$ common features between physiological signal measurements at different time stamps.

We can thus see that PSKA meets all our design goals. The PPG-based PSKA requires a smaller polynomial order and shows more time variance compared to EKG-based PSKA. Interestingly, the temporal variance plots for EKG and PPG also illustrate the level of synchronization required between sensors when either is used as the physiological signal of choice. We see that for both EKG and PPG the violations are highest when the time difference between the iterations is around 1 second, irrespective of the polynomial order used. This means that the features measured one second apart have not changed considerably thereby still allowing successful unlocking of the vault. We can therefore say that even if the communicating sensors measure their physiological signals for PSKA a second apart (even longer for EKG) they will succeed is agreeing on a common key.

[5]We considered over 100 random start-times for PPG and EKG, over all the ten subjects to compute the average violation

VII. Prototype Implementation

In order to estimate the cost and performance of PSKA in hardware we prototyped it using VHDL. The Altera Quartus software tool was used emulating a Stratix II platform (http://www.altera.com). The details of the implementation have not be presented here for space reasons, and can be found in our paper [1]. The metric used for the evaluation are - 1) CPU clock cycles, and 2) memory footprint. We also compare PSKA with the Diffie-Hellman protocol. We choose Diffie-Hellman as - 1) it is well known key distribution protocols which distribute pair-wise keys 2) it is used in the domain of sensor networks, and 3) it requires minimal pre-deployment. For PSKA the physiological signals for both PPG and EKG were provided from a file. Further, in this implementation the fuzzy vault constructed utilized a 9th order and 12th order polynomial for PPG and EKG signals, respectively. For the DH protocol we assumed the values of the generator and modulus are pre-deployed.

Table III shows the average computational cost associated with our implementation of PSKA using PPG and EKG. The cost is expressed in terms of clock cycles required to perform the various tasks of PSKA - 1) polynomial coefficient generation, 2) FFT computation, 3) feature generation, 4) key hiding (polynomial evaluation and chaff point generation for the sender), and 5) key un-hiding (Lagrangian interpolation). The results have been averaged by executing PSKA over 100 iterations at random start-times for each of the ten subjects in the cases of both EKG and PPG. The clock cycles required for feature generation are slightly different for senders and receivers due to the difference in the number of features observed at each end. The receiver requires more clock cycles than the sender because of its need to perform key un-hiding with Lagrangian interpolation using $v+1$ (v is the polynomial order) points at a time from the total number of features observed (Q). Further, the key hiding stage is cheaper even though it requires chaff point generation and polynomial projection of features because it is executed in parallel with the feature generation. If done sequentially it would require about 3000 extra cycles. Another consequence of these results is that the execution of the PSKA does not affect the latency requirements of the design goal. If we assume a 8MHz clock (as the Mote platform developed by Crossbow Inc (http://www.xbow.com)), the time required for executing one

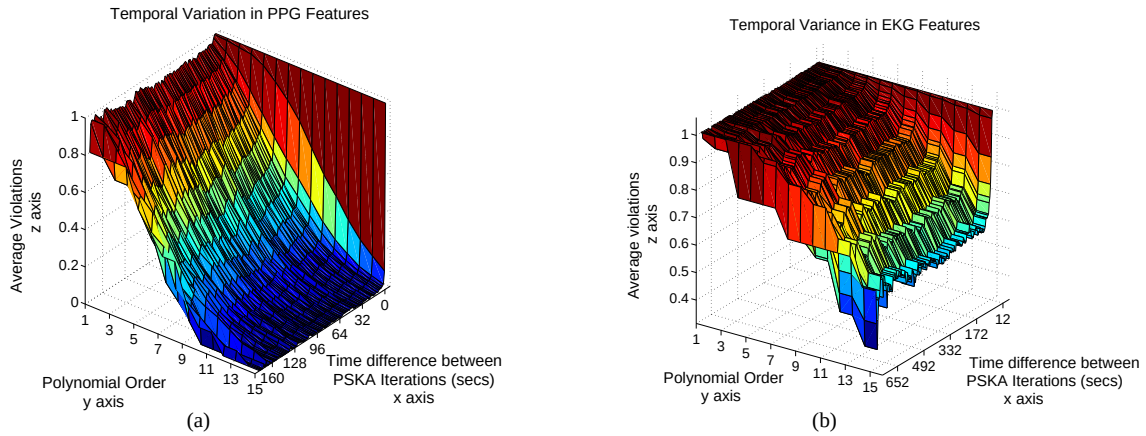


Fig. 8. **a) Temporal Variance in PPG Features, b) Temporal Variance in EKG Features.**

iteration of PSKA would be only a few milliseconds. Table III also shows the memory footprint required for implementing PPG and EKG based PSKA. The primary component of the these footprint values are the chaff points (3000, 13 bit x-values and 23 bit y-values), physiological features (13 bit values, about 30 for PPG and about 85 for EKG) and their polynomial projections (23 bit values). The footprint for EKG based PSKA is greater than PPG because of its larger number of features.

In order to put the performance of PSKA in perspective, we compared it with the implementation of Diffie-Hellman (DH) and Elliptic Curve Diffie-Hellman (ECDH) key agreement protocols on the same platform using VHDL. For DH we used 1024 bit modulus and 160 bit exponent, which is equivalent to 80 bits of security in the symmetric cryptography, while for ECDH we used a 163 bit public key, as in [14]. Table IV shows the results of our experiments. The computational cost and memory footprint are identical for senders and receivers in both DH and ECDH. We find DH is takes 10 times the number of clock cycles than PSKA, primarily because it requires exponentiation of large numbers. The ECDH protocol is much cheaper than DH, it still utilizes more computational cycles than PSKA due to its elliptic curve multiplications and additions. The memory footprint for DH and ECDH protocols is much lower than PSKA because they do not require the storage of any chaff points or features.

It can be seen that PSKA trades-off memory for computational cycles compared to both Diffie-Hellman protocols. Given the ever increasing amounts of memory in systems, the trade-off is not detrimental. However, the down-side for using PSKA is the amount of communication it requires. For each iteration of PSKA to complete it needs to transmit $\sim$10KB of data (when 3000 chaff points are used) as compared to around 0.5 KB for both Diffie-Hellman protocols in the worst case. This may be taxing on the sensor hardware. One way to amortize this cost is to perform data communication along with key agreement as suggested earlier. We believe, the best use of PSKA is to perform initial key agreement and then use the key agreed upon for subsequent secure communication requirements. However, it should be noted that both Diffie-Hellman protocols do not provide any form of authentication [15]. Therefore, a given sensor can potentially agree on a key with any entity (malicious or otherwise). Therefore any execution of the Diffie-Hellman protocols has to be preceded by an authentication protocol, which will increase its overhead further. With PSKA, authentication is inbuilt due to the distinctiveness property of the scheme. Therefore we contend that it is feasible to implement PSKA to enable usable security in BANs.

VIII. Related Work

Recent years have seen growing interest in using physiological signals for generating cryptographic primitives. In, [27], our preliminary work, we presented PPG based PSKA for key agreement. However the work was limited in scope and did not include execution of PSKA using EKG, comparative analysis between EKG and PPG based implementations of PSKA, and a study of implementation cost of PSKA. The use of EKG directly for key generation was studied by us in [26]. The approach also depended upon the generation of frequency domain features which were then exchanged between the sensors and used for generating common keys. However, it was later found that the way the features were extracted during the process tended to distort the original signal considerably and therefore it was not a sound way for key generation. In [2] the authors present an approach for secure communications in BANs which uses IPI signals and error correction codes to arrive at the common key. However, their choice of time domain features makes them susceptible to synchronization and feature reordering/introduction issues.

In [11], the authors suggest the use of a fuzzy vault which has the ability to tolerate the re-ordering of information symbols. The scheme is ideal for approaches dealing with physiological signal features, as two samples of any physiological signal are never completely identical due to the dynamic nature of the body. The fuzzy vault scheme has so far been primarily applied to biometric based authentication using finger-prints [25] and iris-images [22]. However, the non-variant biometric template opens up the fuzzy vault to attacks involving template modification [23]. Similarly, in [13], the authors present the implementation of an attack on fuzzy vault system used for biometric authentication systems. The attacker intercepts two

TABLE III
PSKA Computational Cost (in Avg. Clock Cycles), and Memory Footprint

Signal	Entity	Key Gen	FFT	Feature Gen	Key Hide	Key Un-Hide	Total Cycles	Memory Footprint
PPG	Sender	9	1280	5112.2	31.95	-	6,433.15	47.35KB
	Recvr.	-	1280	5045.42	-	5454.38	11,779.80	45.3KB
EKG	Sender	12	512	22424.46	87.59	-	23,036.05	48.41KB
	Recvr.	-	512	22221.64	-	9789.28	32,522.92	46.31KB

TABLE IV
Diffie-Hellman Computational Cost (in Avg. Clock Cycles), and Memory Footprint

Type	Clock Cycles	Memory Footprint
Diffie-Hellman (24 bit modulus, 160 bit exponent)	327,680	7KB
Elliptic Curve Diffie-Hellman (163bit public key)	135,456	2.5KB

vaults generated from the same biometric (fingerprint) data with different chaff points, correlates them to reveal the hidden biometric features. The both the attack succeed because the fingerprint features in vaults do not change. These attacks are not possible with PSKA because the feature values in the fuzzy vaults generated in two iteration of PSKA are drastically different due to the temporal variance property of the physiological signals used. Therefore trying to tamper with a vault or correlating two vaults will not give the attacker any information in PSKA.

In [3] the authors propose a method to reduce the effort required by an attacker to find the legitimate points from a fuzzy vault. They propose the notion of free area of a point in the vault which is defined as the positions in the vault where the new point can be placed. The authors argue that for two points $p1$ and $p2$ in the vault, if free area of $p1$ is greater than the free area of $p2$ then the probability of $p1$ being added to the vault earlier than $p2$ is higher than the points being added the other way around. An attacker, by ordering the points in the vault in decreasing order of free area can distinguish between the legitimate and chaff points in a vault (the legitimate points in the vault would be the ones which were added to the vault the earliest and therefore have higher free area compared to chaff points added later). However, by their own admission they could not prove this claim and therefore try to give empirical evidence to its validity. We believe that the attack proposed in this work is more of a mathematical exercise rather than a genuine attack on the vault from the perspective of the attacker. The authors have simply computed the probability of a point having high free area being a genuine point in the vault by trying out thousands of different vaults and state there is a "good" probability associated with this fact. In order to compute this probability they fill the vault keeping track of which points are genuine and which are not, compute the free area of the points, and see in how many instances the genuine points had high free area. Their results show that for vaults with very large set of features points (165, 330 and 495 points) the probability of successfully stating that a point with high free area is genuine approaches 0.5 (Figure 4 in the paper), which is essentially random. With PSKA the number of features is in the range of 30 for PPG and 80 for EKG, the probability of a point having high free area being a legitimate point will be minuscule. The attack proposed in [3], therefore, does not provide an attacker with any distinct advantage in identifying legitimate points from the chaff over a brute-force attack with respect to PSKA.

In [16] the author suggests a brute-force attack on the fuzzy vault which bounds the number of operations required to guess the legitimate points in fuzzy vault, with a high probability, using a polynomial of order v to $8vlog^2v(|R|/|N|)^v$. The consequence of this result is that with a probability close to 1, the complexity of identifying the polynomial used by the vault decreases. Using our parameters R = 5000, N = 30, v = 9 for PPG based PSKA is now only as secure as brute-forcing a 75 bit key rather the original 95 bit key (based on our previous estimate, where we were trying to compute the upper bound of brute-forcing the vault; the complexity of identifying $v + 1$ points on the chosen polynomial with probability equal to 1), still fairly strong. If we increase the polynomial order to 12, the strength goes up to 97 bits. Our EKG results which uses R = 5000, N = 87, v = 12 provides a security of about 80 bits. However, given the larger number of feature points (t) the overall security provided by EKG is lower than PPG, keeping everything else constant. Further, in both cases, if we increase the number of chaff points (our feature length is 13bits which allows for a range of 0 - 2^{13}) we can again increase the complexity of breaking the vault. Of course, increasing the polynomial order or vault size is going to increase the memory usage and communication complexity, but given that the length of the key exchanged using PSKA is arbitrarily long, we do not have to execute PSKA regularly and thus should be affordable.

IX. Conclusions

In this paper, we presented a usable and secure key agreement scheme for BAN called Physiological Signal based Key Agreement (PSKA). It allows two sensors to agree on a shared key, in an authenticated manner, without any form of initialization or setup. Simple deployment of the sensors is enough to allow them to agree upon a common key, in a largely transparent manner. The security analysis of the PSKA protocol showed that physiological signals meets the design goals set forth if physiological signals are used as a basis of key agreement. We also analyze the performance and cost of using the PSKA protocol by prototyping it in VHDL. We then compared its performance with both exponentiation-based and elliptic curve versions of Diffie-Hellman protocols which enable key exchange in open channel, and found that our approach requires considerably less computation, albeit at the cost of memory and communication overhead. PSKA is thus a viable approach to secure key agreement in BANs. Future work includes an expanded in-field study of PSKA to better

understand the distinctiveness and temporal variance properties of the scheme.

ACKNOWLEDGEMENTS

This research was funded in part by National Science Foundation Grants CNS-0617671 and CT-0831544.

REFERENCES

[1] A. Banerjee, K. Venkatasubramanian, and S. K. S. Gupta. Challenges of Implementing Cyber-Physical Security Solutions in Body Area Networks. April 2009. Proc. of 4th Intl. Conf. on Body Area Networks.

[2] F. M. Bui and D. Hatzinakos. Biometric Methods for Secure Communications in Body Sensor Networks: Resource-Efficient Key Management and Signal-Level Data Scrambling. *Proc. of EURASIP Journal on Advances in Signal Processing*, pages 1–16, 2008.

[3] E.-C. Chang, R. Shen, and F. W. Teo. Finding the original point set hidden among chaff. In *In Proc. of the ACM Symposium on Information, Computer and Communications Security*, pages 182–188, 2006.

[4] S. Cherukuri, K. Venkatasubramanian, and S. K. S. Gupta. BioSec: A Biometric Based Approach for Securing Communication in Wireless Networks of Biosensors Implanted in the Human Body. pages 432–439, Oct 2003. Proc. of Wireless Security & Privacy Workshop.

[5] A. DroitCour. Non-contact measurement of heart and respiration rates with a single-chip microwave doppler radar, 2006. PhD Thesis, Stanford University.

[6] J. Elson, L. Girod, and D. Estrin. Fine-grained network time synchronization using reference broadcasts. pages 147–163. Proc. of 5th symposium on Operating systems design and implementation.

[7] J. Elson and K. Romer. Wireless sensor networks: A new regime for time synchronization. pages 28–29, 2002. In Proc. of the First Workshop on Hot Topics In Networks (HotNets-I).

[8] L. Eschenauer and V. D. Gligor. A Key-Management Scheme for Distributed Sensor Networks. pages 41–47, Nov 2002. In Proc. of the 9th ACM Conf. on Computer and Communication Security.

[9] S. C. et. al. A Low-power Star-topology Body Area Network Controller for Periodic Data Monitoring Around and Inside the Human Body. pages 139–140, 2006. In Proc. of the IEEE International Symposium on Wearable Computers.

[10] E. F. Greneker. Radar sensing of heartbeat and respiration at a distance with applications of the technology. *IEE Radar*, 449:150–154, 1997.

[11] A. Juels and M. Sudan. A Fuzzy Vault Scheme. page 408, 2002. Proc. of IEEE Intl. Symposium on Information Theory.

[12] A. Juels and M. Wattenberg. A Fuzzy Commitment Scheme. pages 28–36, Nov 1999. Proc. of ACM 9th Conf. on Computer and Communication Security.

[13] A. Kholmatov and B. Yanikoglu. Realization of correlation attack against the fuzzy vault scheme. *SPIE Security, Forensics, Steganography, and Watermarking of Multimedia Contents X*, 6819(1), Jan 2008.

[14] D. J. Malan, M. Welsh, and M. D. Smith. A Public-Key Infrastructure for Key Distribution in TinyOS Based on Elliptic Curve Cryptography. pages 71–80, Oct 2004. In Proc. of IEEE 2nd Intl. Conf. on Sensor & Ad Hoc Comm. & Networks.

[15] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC, October 1996.

[16] P. Mihailescu. The Fuzzy Vault For Fingerprints Is Vulnerable To Brute Force Attack. The Computing Research Repository, http://www.citebase.org/abstract?id=oai:arXiv.org:0708.2974, Aug 2007.

[17] K. Ouchi, T. Suzuki, and M. Doi. LifeMinder: A Wearable Healthcare Support System Using User's Context. pages 791–792, July 2002. In Proc. of 22th International Conf. on Dist. Comp. Sys. Workshops.

[18] R. Paranjape, J. Mahovsky, L. Benedicenti, and Z. Koles. The electroencephalogram as a biometric. *Electrical and Computer Engineering, 2001. Canadian Conference on*, 2:1363–1366, 2001.

[19] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and D. Tygar. SPINS: Security Protocol for Sensor Networks. *Wireless Networks*, 8(5):521–534, September 2002.

[20] C. C. Y. Poon, Y.-T. Zhang, and S.-D. Bao. A Novel Biometrics Method To Secure Wireless Body Area Sensor Networks for Telemedicine And M-Health. *IEEE Communications Magazine*, 44(4):73–81, 2006.

[21] M. Poulos, M. Rangoussi, and E. Kafetzopoulos. Person identification via the eeg using computational geometry algorithms. pages 2125–2128, September 1998. In Proc. of the 9th European Signal Processing.

[22] E. S. Reddy and I. R. Babu. Authentication using fuzzy vault based on iris textures. pages 361–368, 2008. In Proc. of the 2nd Asia Intl. Conf. on Modelling & Simulation.

[23] W. J. Scheirer and T. E. Boult. Cracking Fuzzy Vaults and Biometric Encryption. pages 1–6, Sept 2007. Proc. of Biometrics Symposium.

[24] M. Sichitiu and C. Veerarittiphan. Simple, accurate time synchronization for wireless sensor networks. *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*, 2:1266–1273, March 2003.

[25] U. Uludag, S. Pankanti, and A. K. Jain. Fuzzy Vault for Fingerprints. pages 310–319, July 2005. Proc. of Audio- and Video-based Biometric Person Authentication.

[26] K. Venkatasubramanian, A. Banerjee, and S. K. S. Gupta. EKG-based Key Agreement in Body Sensor Networks. pages 1–6, April 2008. Proc. of 2nd Workshop on Mission Critical Networks.

[27] K. Venkatasubramanian, A. Banerjee, and S. K. S. Gupta. Plethysmogram-based Secure Inter-Sensor Communication in Body Area Networks. pages 1–7, November 2008. Proc. of IEEE Military Communications Conference.

[28] K. Venkatasubramanian and S. K. S. Gupta. Security for Pervasive Health Monitoring Sensor Applications. pages 197–202, Dec 2006. In Proc. of the 4th Intl. Conf. on Intelligent Sensing & Information Processing.

[29] K. Venkatasubramanian et. al. Ayushman: A Wireless Sensor Network Based Health Monitoring Infrastructure and Testbed. pages 406–407, June 2005. Proc. of IEEE Intl. Conf. on Distributed Computing in Sensor Systems.

[30] S. Zhu, S. Setia, and S. Jajodia. LEAP+: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks. *ACM Trans. on Sensor Networks (TOSN)*, 2(4):500–528, Nov 2006.

Krishna K. Venkatasubramanian is a post-doctoral researcher at the Computer and Information Science Department at the University of Pennsylvania, Philadelphia, PA. He received his B.S. in Computer Science from Webster University, St. Louis, MO, and M.S. and Ph.D. in Computer Science from Arizona State University, Tempe, AZ. His research focuses on secure cyber-physical systems, body area networks, trust management, and medical device security. His publication list is available at http://www.seas.upenn.edu/~vkris/. Dr. Venkatasubramanian is a member of the ACM and the IEEE.

Ayan Banerjee is a Ph.D. student at the School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ. He received his B.E. in Electronics and Telecommunication Engineering from Jadavpur University, Kolkata, India. Since Fall 2007 he has been with the IMPACT Lab at Arizona State University. His research interests include safety and security of cyber-physical systems.

Sandeep Kumar S. Gupta is a Professor with the School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ. He received the B.Tech degree in Computer Science and Engineering (CSE) from Institute of Technology, Banaras Hindu University, Varanasi, India, M.Tech. degree in CSE from Indian Institute of Technology, Kanpur, and M.S. and Ph.D. degree in Computer and Information Science from Ohio State University, Columbus, OH. His current research focuses on dependable, criticality-aware, adaptive distributed systems with emphasis on wireless sensor networks, thermal and power-aware computing and communication, and pervasive healthcare. He has co-authored the book Fundamentals of Mobile and Pervasive Computing, McGraw Hill, and is currently on the editorial board of IEEE Communication Letters, IEEE Transactions on Parallel & Distributed Systems and Springer Wireless Networks. He is a member of the ACM and a senior member of the IEEE. Dr. Gupta heads the IMPACT (Intelligent Mobile and Pervasive Applications and Computing Technologies) Lab at Arizona State University. For information about his recent research projects and publications please visit http://impact.asu.edu.