Plethysmogram-based Secure Inter-Sensor Communication in Body Area Networks

Krishna K. Venkatasubramanian, Ayan Banerjee and Sandeep K. S. Gupta
IMPACT Lab (http://impact.asu.edu)
Department of Computer Science and Engineering
Arizona State University
Tempe, Arizona 85287
{kkv,abanerj3,sandeep.gupta}@asu.edu

***Abstract*—Body Area Networks (BAN) can play a major role in monitoring the health of soldiers in a battlefield. Securing BANs is essential to ensure safety of the soldiers. This paper presents a novel key agreement protocol called Photoplethysmogram PPG-based based Key Agreement (PKA) which allows sensors in a BAN to agree to a common key using PPG values obtained from the subject (soldier) they are deployed on. Using the stimuli which the sensors are designed to monitor directly for cryptographic purposes, enables administrators to provide security for BANs with minimal initial setup. The principal contributions of this paper are: 1) demonstration of the viability of the PPG signals for agreeing upon common symmetric cryptographic keys between two nodes in BAN, and 2) analysis of the security, performance and quality of the keys produced by PKA.**

I. Introduction

Monitoring soldiers' vital signs during combat and training can play an important role in aiding commanders to better assess the state of their troops, plan operations and logistics (http://www.usariem.army.mil/wpsm/index.html). Recent developments in low-powered electronics have lead to wearable or implantable health monitoring **sensors**. Sensors are battery-powered nodes consisting of physiological monitoring, actuation, computation, storage and wireless communication capabilities. These sensors usually form a multi-hop wireless network over the soldier's body, called *Body Area Networks* (BAN). BANs provide the ability to pervasively collect, process, store and forward health information from soldiers on the field to appropriate personnel such as squad medic or medical personnel in the field headquarters. In the rest of the paper we use the term sensors or nodes, interchangeably.

Modern wars are fought in both physical and cyber-space. The sensitive nature of the data collected makes BANs a target for malicious entities to exploit. Lack of adequate security features may not only lead to a breach of soldier's privacy, but may enable a malicious entity to modify data from the BAN to mislead the field commanders and medical staff. Therefore, the ability to ensure the confidentiality and integrity of personally identifiable health information and to prevent any unauthorized access to it (as stated in the Health Insurance Portability and Accountability Act (HIPAA) (http://www.hhs.gov/ocr/hipaa/), is important in military scenarios, as well. One of the most vulnerable aspect of BANs is the use of wireless communication. This allows adversaries to remotely monitor the communication and potentially inject malicious messages, which give wrong status of a soldier's health or trigger inappropriate and even injurious medical actions. Securing inter-sensor communication therefore is one of the most important aspects of securing the BAN.

Sensors rely on cryptographic keys to secure their communication. Keys are usually made available to sensors through explicit key distribution protocols. All classes of key distribution in sensor networks have required some form of pre-deployment. Examples include probabilistic key distribution schemes [2], master key based key distribution schemes [17] and even asymmetric crypto-systems [7]. However, given the progressively increasing size of BANs (networks of size 190-255 nodes have already been proposed [4]), these approaches may potentially involve considerable latency (due to node programming) during network initialization/setup or any subsequent adjustments, due to their need for pre-deployment. We believe that for BANs to be useful in a military setting, they should be plug-n-play. For example, soldiers (or their medics) should be able to add, remove and adjust the sensors on their BAN as and when required without reconfiguring the parts of the network (a very important requirement in time-critical environment such as during active combat) and still have secure communication.

In this paper we present a novel key agreement scheme called *PPG based Key Agreement (PKA)*, which utilizes photoplethysmogram (PPG) signals for enabling sensors to agree upon a symmetric cryptographic key for securing unicast communication between them. The idea of using physiological value 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 [16]. Broadly speaking, PKA works as follows: 1) the sensors which want to securely communicate, measure the PPG 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 PPG 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 un-hiding process are based on the *fuzzy vault* cryptographic primitive, first suggested in [5].

PKA is being implemented as a part of the securing Ayushman health monitoring system [3] being developed at the IMPACT labs at Arizona State University. The use of

978-1-4244-2677-5/08/$25.00 ©2008 IEEE

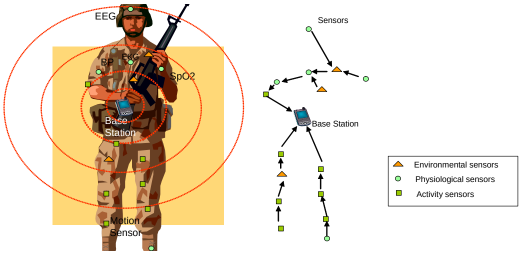
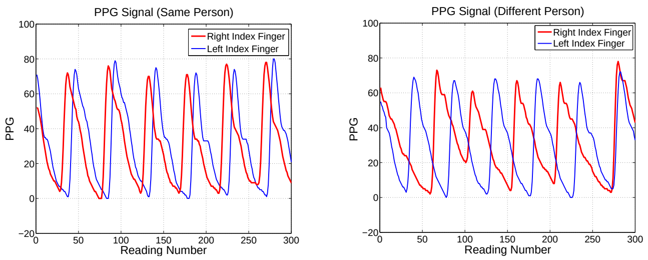


Fig. 1. Body Area Network



(a)

(b)

Fig. 2. PPG signals measured from - a) same individual synchronously (Duration: 5 sec) b) different individuals synchronously (Duration: 5 sec)

PKA can eliminate the need for explicit key distribution in the BAN. Sensors can agree upon keys as and when needed. Additionally, the key agreement technique used by PKA meets the **design goals** suggested for physiological value based keys in [9], namely - 1) The keys agreed upon are *long* and *random*; 2) Knowing the physiological signals at any time will not provide significant advantage in knowing the keys agreed upon in future executions of the scheme, i.e. *time variance*. This is an important property which *differentiates the proposed technique from traditional biometrics based techniques* where once a template is created it is never changed [12]. In our case we want the values to change with time, and as we shall show physiological values such as the PPG meet this criteria; 3) the physiological stimuli used for the agreement is *universally measurable* (PPG); 4) Knowing the physiological value (PPG values) of one individual will not provide significant advantage in guessing the keys being agreed by sensors on another individual, i.e. *distinctiveness*. The **contribution** of this paper is two fold: 1) a scheme for agreeing upon common cryptographic keys between two nodes in BANs using PPG (Sections III, IV and VII) and 2) to show that the key agreement scheme meets the aforementioned design-goals based on data from real subjects (Sections V and VI).

II. System Model

We assume a Body Area Network (BAN) to be a network of physiological and environmental monitoring sensors which are worn and/or implanted on a *subject* or *individual*. The sensors 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. We assume that the sensors *communicate through the wireless medium*, as wires running between sensors in a BAN will make it obtrusive. All sensors are assumed to be able to measure the PPG signals. Already physiological monitoring sensors are becoming multi-modal and are able to sense multiple types of stimuli [8]. For example, a blood pressure can be used to sense systolic/diastolic pressure, heart rate, mean arterial pressure. Here we assume PPG can be measured by the nodes trying to agree on a key.

The threats faced by a BAN is primarily from adversaries who can eavesdrop on all the traffic within the BAN, inject messages, replay old messages, spoof sensor identities. The wireless medium is therefore not trusted by the sensors. Any entity not in contact with the subject cannot measure PPG signals from the subject. 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 SSL given the considerable capabilities of the entities involved.

The purpose of *PPG-based Key Agreement (PKA)* is to enable two sensors to obtain a common (shared) symmetric key, using Photoplethysmogram (PPG) signal, which they can use to secure the communication between themselves. The PPG signal is a measure of the volumetric change in the distention of arteries due to the perfusion of blood through them during a cardiac cycle. It is measured using a pulse oximeter which can be attached to the subject's fingers or ear-lobes. The PPG signal is thus a representation of a subject's cardiac cycle just as electrocardiogram signal, which is generated by the electrical activity of the heart. PPG signal can be used for key agreement, through a two steps process: 1) *physiological feature generation*, and 2) *PPG-based key agreement*, which we describe in detail Sections III and IV, respectively.

III. Physiological Feature Generation

One of the first things needed for key agreement between two sensors in a BAN is to be able to identify something common between them which only they know. Traditionally, some form of deployed value is used for this purpose, requiring human intervention in the network's functionality. Here, we propose a more plug-n-play approach, by utilizing features derived from specific physiological values that both sensors can measure - Photoplethysmogram (PPG).

We perform a frequency domain analysis of PPG signals for generating the features. This is because, frequency components of physiological signals, at any given time, have similar values irrespective of where they are measured on the body. A time domain analysis showed that the values of two PPG signals measured at different parts of the body (at two different fingers) have similar trend but diverse values. Another advantage of using frequency domain features is that the required level of synchronization during measurement of the PPG signal at both sensors for feature generation is not very strict. Our experiments show good results even with a 1 second difference in the start-times of the PPG measurement at the sensors. If PPG signals were to be analyzed in time domain, the level of synchronization would be about $1/f$, where f is the sampling frequency, which in this case is 1/60Hz = 16.6 msec [13]. Figure 2 shows synchronously collected PPG signals from two sensors located on the same subject and different subjects.

The feature generation process is executed by both the sensors in the following manner - 1) Both sensors sampling

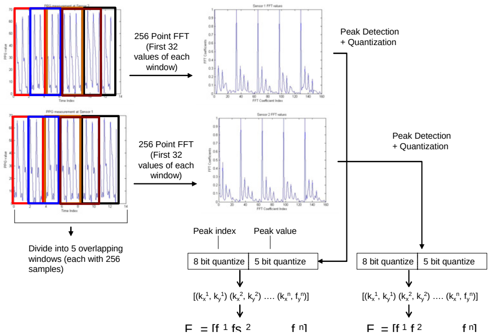

256 Point FFT (First 32 values of each window)
Peak Detection + Quantization
256 Point FFT (First 32 values of each window)
Peak Detection + Quantization
Divide into 5 overlapping windows (each with 256 samples)
Peak index
Peak value
8 bit quantize
5 bit quantize
8 bit quantize
5 bit quantize
[(kx1, ky1) (kx2, ky2) …. (kxn, fyn)]
[(kx1, ky1) (kx2, ky2) …. (kxn, fyn)]
F = [f 1 fs 2 f n]
F = [f 1 f 2 f n]
Sensor 1 FFT values
Sensor 2 FFT values
PPG measurement at Sensor 1
Time Index
FFT Coefficient Index

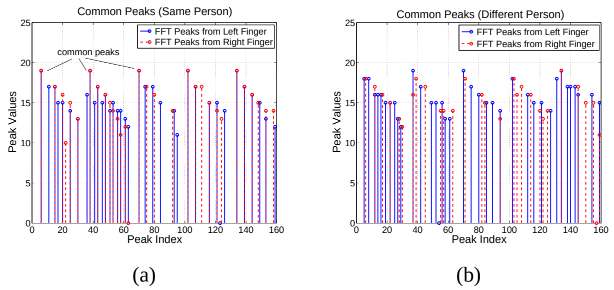
Fig. 3. Peak based Feature Generation

the PPG signal in a loosely synchronized manner, at a specific sampling rate for a fixed duration of time (60Hz and 12.8 seconds, respectively in our case producing 60 × 12.8 = 768 samples); 2) The samples are divided into five overlapping windows of 256 samples each. A 256 point Fast Fourier Transform (FFT) is then performed on each of these parts; 3) The first 32 FFT coefficients of each of the five windows are then passed through a peak detection function (simple local maxima detector) which returns a tuple of the form $< k_x^i, k_y^i >$ where k_x^i is the index of the peaks, k_y^i is its corresponding FFT coefficient values, and i is the peak id, the maximum value of which is the total number of peaks observed. The number of peaks observed by a sensor vary with situation, but on an average it is around 30; 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*. Figure 3 shows the feature generation process. Each individual feature obtained from a single measurement is 13 bits long (8 bits for k_x and 5 bits for k_y) and is concatenated 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 (which is same as the number of peaks observed, i.e. $N = 30$).

Figure 4 shows peak values versus peak index graph for PPG 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 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 4, sensors on the same person have a higher number of super-imposed lines (highly correlated) compared to sensors on two different people. The lack of correlation between the FFTs of sensors on two different people is due to the difference in physiological signature of each person at any given time. The FFT peaks can thus clearly be used to distinguish between sensors which are in the same BAN or different BANs providing an efficient authentication mechanism and a basis of key agreement (as we shall see in the next section). This was the primary reason for choosing FFT peaks as features.

IV. PPG based Key Agreement

Once the feature vector has been generated, it can then be used by the sensors to agree upon a common key. Broadly speaking, the scheme 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 PPG 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 correcting the differences between its feature vector and the one used by the sender. The idea of using error correction to arrive at a common key for any physiological value (the paper itself did not focus on any particular physiological value) was first argued in our preliminary work in [1]. The inspiration behind the idea was the observation that each measurement of physiological values was independent of others, any difference is their measured values could be modeled as communication error. We proposed the use of simple error correction schemes such as majority decoding as proposed in [6]. A inherent problem with using this approach is that though 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 [5]. We address this by proposing the use of a cryptographic construct called *Fuzzy Vault* [5].



(a)

(b)

Fig. 4. FFT Peaks (peak index vs. peak values) for - a) same individual (Total: 12) b) different individuals (Total: 2)

A. Fuzzy Vault

The fuzzy vault scheme first proposed in [5] 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 “significant” number of values in common with set A. The construction and locking of the vault is done 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 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 [5]. We can map this scheme onto PKA 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.

Example 1: Let the polynomial be $p(x) = x + 1$, set A be $A = \{1, 2, 3\}$ and B be $B = \{1, 3, 4\}$, 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

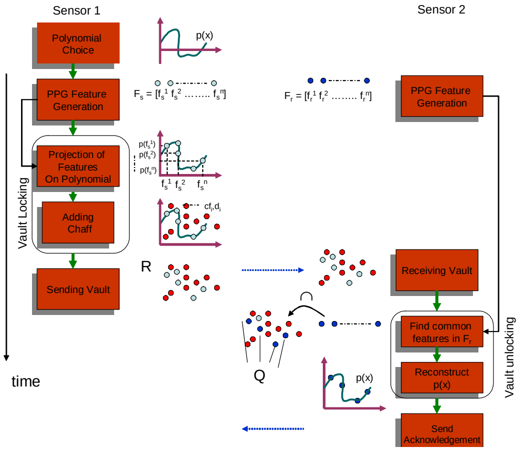


Fig. 5. Key Agreement in PKA

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 first order polynomial and thus unlock the secret.

B. Vault Locking & Unlocking

The use of polynomials ensures that the sets A and B need not have any order to them as long as they have significant number of common values. The presence of the chaff points adds security to the vault and hides 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 we use fuzzy vault for PKA. We use the term *sender* for the sensor which creates the vault and locks it, and the *receiver* for the sensor which unlocks the vault to access the secret key. The key agreement occurs as follows:

Polynomial Choice: 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 values of coefficients are randomly selected. The order of the polynomial (v) is not a secret and is known to all sensors in the BAN. 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, and depending upon the order of the polynomial used the coefficients are obtained by dividing the Key accordingly.

PPG Feature Generation: The sender and the receiver then measure the PPG signal and generate feature vectors $F_s = f_s^1, f_s^2, ...f_s^N$ and $F_r = f_r^1, f_r^2, ...f_r^N$, respectively. As the features f_s^i and f_r^i are represented by 13 bits, the maximum number of features possible is 2^{13}.

Vault Creation: The sender then computes 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 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 $1 \leq j \leq M$. The value cf_i of each of the chaff points, is within the same range as that of the features (i.e 2^{13}). 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 set R as *Vault Size*.

Vault Locking: The sender constructs the vault $R = P \cup C$, randomly mixes the values, i.e. $R = mix(R)$, to ensure the chaff points and the legitimate points are indistinguishable, which otherwise will follow one another.

Vault Exchange: The sender then sends the vault R to the receiver using the following message: $Sender \rightarrow Receiver : IDs, Nonce, R, MAC(Key, R|Nonce|IDs)$. Here IDs is the id of the sender and $Nonce$ is a unique random number for transaction freshness, MAC is the message authentication code, the key (Key) used for computing the MAC 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 [12]) 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 us to reconstruct 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 $|Q| > v$ 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 to check the validity of the unlocking.

Vault Acknowledgement: 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: $Receiver \rightarrow Sender : IDr, MAC(Key, Nonce|IDs|IDr)$. The symbols have the same meaning as above.

Figure 5 shows the feature generation process. We refer to the execution of these seven steps as an *iteration* of PKA. The key agreement protocol need not be executed in isolation as shown above and can be combined with secure data exchange. 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 [2] [17] [7] nor physiological value based approaches [9] can achieve this property. Finally, the one-hop security provided by physiological values can be easily extended to multi-hop end-to-end communication, where a physiological value based key is generated between each link on the path to the base station. This might increase the latency within the network which might be a problem if BAN is being used in emergency situations. To minimize this, sensors can execute PKA once with their neighbors, arrive at a common key, derive keys from this common key, and use them for securing more than one communication.

V. PKA Security Analysis

Security issues for PKA arise due to its need to communicate the vault. An eavesdropper can record this message and try to construct the hidden polynomial (key) from it. In this section we discuss the security implications of the two principal aspects of PKA: the vault and its exchange.

Vault Security: The use of the fuzzy vault construct in PKA ensures that even though the two sensors may not have all the

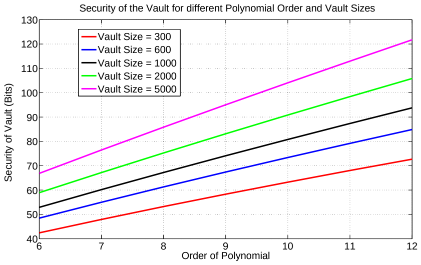


Fig. 6. Vault Strength w.r.t. Polynomial Orders for Different Vault Sizes

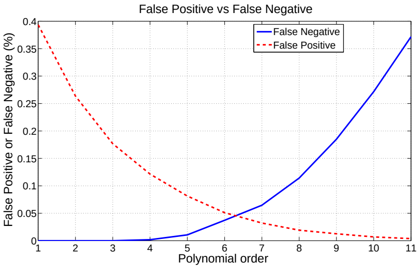


Fig. 7. False Positive vs. False Negative Rate

features in common they can still agree upon a common key in a secure manner. The security of the PKA 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 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 reconstruct the hidden polynomial. An example is the receiver who has to try out $v+1$ points in set Q to arrive at the correct polynomial.

Figure 6 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 to find $v+1$ legitimate points. 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 PKA 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 individuals, we can ensure successful vault unlocking for the receiver but not for the adversary. Of course this will work only if there is a discernable difference in the number of features between PPG collected from the same individual and different individuals. More on this in Section VI.

Exchange Security: The vault exchange and acknowledgement phases makes it very difficult for adversaries to know the key being agreed upon. There are fives reasons for this - a) An external malicious entity cannot spoof the identity of a legitimate node or inject bogus massages into the BAN, as it cannot measure PPG and therefore cannot construct a valid vault; b) The vault has a large number of chaff points which makes it difficult for adversaries to know which points are legitimate and which are not (as discussed above); c) 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; d) Replaying the vault exchange message or the acknowledgement message will not provide the adversary with any advantage as the repetition of nonce values will simply lead to the rejection of the messages; e) Even an adversary with the knowledge of the current legitimate features will not be able to reuse them because as each time the protocol is executed new PPG features are derived, which change with time (see Section VI). With PKA only the sender has the responsibility of generating the key by choosing the polynomial, the receiver only reconstructs it. This may be a problem if there is a difference in the capabilities of the sensors, resulting in poorly chosen keys. One way to overcome it is to perform the key agreement twice with each node generating a key once. Both keys can then be XORed to generate the common key.

VI. Performance Results

In this section we analyze the two important characteristics of PKA which define its security - the distinctiveness in the number of common features generated from PPG measured at two BANs and same BAN, and the temporal variance of the PPG signal to ensure that a knowing the features used in current iteration of PKA will not allow compromise of subsequent PKA iterations. The analysis utilizes actual PPG data from 10 subjects which we collected from volunteers 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 hand firmly placed on a desk, the oximeter sensor was placed on the index finger of each hand. Data was collected for about 5 minutes from each subject at a sampling rate of 60Hz. The PKA implementation and analysis was done using Matlab.

A. Distinctiveness

First, we check to see if PPG features can be used for distinguishing between people. This is important because we do no want the vault created by a sensor in one BAN be unlocked by another sensor located on another subject based on features generated from its measurements. In our case we want to make sure the number of common features for sensors on the same subject be significantly different from the common tuples for sensors on the different subject. Our definition of "significant" is defined based on the polynomial order (v) used.

To find the statistic on the number of common features, we measured the PPG at 113 different start-times which were

1.6 seconds apart for each of the 10 people for whom we have data. For each subject we had two time series of data collected from left and right index fingers. We found that the average number of features observed in each time series was about 30. Out of the total number of features on average only two features were common between PPG time series collected from two different people. While the most commonly observed value was only 0.8. Executing the same experiment for data from the same subject showed an average of 12 features to be common with the most commonly observed value of 14.8. We can thus see that PPG features create a clear distinction between people. 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 number of times the common features between two people does not exceed it (*false positives*) and minimize the number of times the common features for the same subject is below this value (*false negatives*). Figure 7 shows the percentage of false positives and false negatives for different order of polynomials. Note that there is an additional trade-off between the choice for the order of the polynomial and level of security. Even though 7^{th} order polynomial gives the lowest false positive and false negative rates, the security provided by it for a given vault size will be lower than a 9^{th} order polynomial. But higher polynomial order will increase the amount of false negatives. In summary, we can say that using features derived from the PPG signal to generate a vault, does not give any significant advantage to an adversary who uses features derived from another subject.

B. *Temporal Variance*

Now that we have seen PPG signals are distinctive between two people, we want to see the temporal variance in the PPG features. Figure 8 shows the results for different polynomial orders. The x-axis of the graph is the time difference between the PPG measurement start-times of one iteration of PKA and another. The y-axis shows the *average violations* - which is the percentage of times when the number of common features between the first and second iterations of PKA are greater than the order of the polynomial used, averaged over all 10 subjects. We performed curve fitting on the data to show the overall trend. As expected, when the time difference between the two iterations of PKA is very close violations are very high as the feature values in both the iterations are very similar. But as the time difference increases the violations falls drastically and then stabilizes with a slight downward trend. We believe as the time goes by the average violations will decrease further. Finally, as expected 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 PPGs measurements at different time stamps. We can see that PKA meets all our design goals.

VII. Implementation

We have implemented the PKA protocol in Matlab. It works in real-time, in that it collects data from two PPG sensors attached to the left and right index fingers of a subject (where each sensors is equivalent to a sensor located on a subject) and produces identical keys. When the sensors are connected to two different people the keys produced are not identical, as the number of common features are less than the order of the polynomial which we set to 9. We are currently in the process of implementing PKA on actual sensor hardware as a part of our Ayushman health monitoring system [3] (http://impact.asu.edu/Ayushman.html). Here we provide estimates of the memory, computation and communication requirements of a PKA implementation.

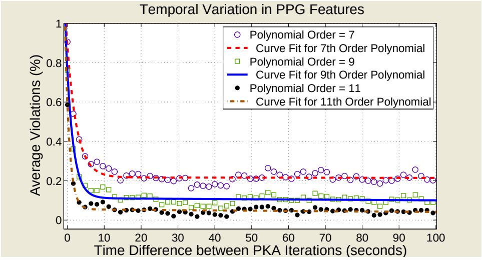


Fig. 8. Effect on Time on the PPG Features

In terms of memory PKA requires the storage of PPG (FFT) features which are represented with 13 bit value. Each sensor generates about 30 features in a PKA iteration which requires $\lceil (13 \times 30)/8 \rceil \sim$ 49 bytes. To store the polynomial projection of each feature we need 23 bits. Thus we need a total of $\lceil (23 \times 30)/8 \rceil \sim$ 87 bytes of memory. Now to store the chaff points we need 36 bits for each chaff point. If we use 1000 chaff points then the total memory storage required will be 36000 bits or approximately 4.5 KB of memory space. So, each iteration of PKA requires about 4.6 KB of memory.

In terms of computation, PKA has two main requirements. FFT computation and polynomial manipulation. For performing FFTs we plan to use customized hardware. In [15], the authors have proposed the use of dedicated FFT processor in the sensor architecture. Their design can perform variable length FFT (128 to 1024 point FFT) on a sensor with an energy dissipation of 155nJ per FFT computation at a supply voltage of 350 mV and a clock frequency of 10 KHz. The processor, implemented using 0.18 μm CMOS technology, will add only a small footprint to sensors. Evaluating a v^{th} order polynomial at a certain point requires $p * (p + 1)/2$ multiplications and p additions. Considering multiplications as the unit computation we need O(p^2) operations to evaluate the polynomial at a single point. Now for each iteration of PKA we need to evaluate the polynomial at N points, where N is the size of the feature vector. So, O(Np^2) operations are performed for each iteration of PKA. If we use p = 9 and N = 30 then there will be 2700 multiplications to perform in each iteration of PKA. This task can be performed easily if we leverage FFT processor boards which have similar capabilities.

The memory and computation requirements for PKA are not very high. A down-side for using PKA is the amount of communication it requires. For each iteration of PKA to complete it needs to transmit about 4.5KB of data. This can 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

PKA is to perform initial key agreement and then use the key agreed upon for further secure communication requirements. Any time in the future if a complete system reset is required PKA can be re-executed to form new keys. PKA is thus ideal for replacing any form of pre-deployment, which essentially performs the same task as PKA but in a plug-n-play manner, i.e. requires sensor programming and user involvement. With PKA we can achieve the same result in a plug-n-play manner.

VIII. Related Work

The idea of using physiological signals for securing inter-sensor communication was first introduced in [1] [14]. The principal idea was to use physiological values for hiding the actual key to be shared between the sensors, and correct the differences in the physiological values using simple error correction scheme. The paper however does not suggest any particular physiological value which can be used. Building upon this initial idea, the authors in [9] propose the use of Inter-Pulse-Interval (IPI) to generate cryptographic keys. They measure IPI from Photoplethysmogram (PPG) and EKG time series by measuring the time difference between the peaks in the EKG/PPG signal. This series of IPI values were then encoded into binary to form a 128 bit cryptographic key. Using IPI for key generation results in keys whose hamming distance was shown to vary considerably (90 bits different) when measured in two different people and considerably less for the same subject. Though the authors suggest the use of error correction codes to make up for the differences, keys for the same subject vary from 0 to 40 bits in particular cases, which is very difficult to correct. The primary reason for this is potential re-ordering of information symbols (due to translational and rotational errors which are common in multiple measurements of physiological and biometric data [5]) which when naively encoded into binary produces drastically different values. The schemes proposed in [9] are therefore ideal for authentication purposes but not necessarily for ensuring confidentiality, though in some very specific values which are only a few bits apart have been reported [9].

In [5] 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 signals because the values produced from measuring the signals are never completely identical. The dynamic nature of the body which makes predicting physiological values so difficult also ensures that any two measurements are not entirely same. The fuzzy vault scheme has so far been primarily applied to biometric based authentication example - finger-prints [12] and iris-image [10]. The main difference between ours and these schemes is that we want our physiological values to vary with time, while they do not. This opens biometric based schemes to different attacks which involve changing the template [11], which we avoid. Ours is the first approach we know of which uses fuzzy vault for securing communication in BANs.

We have previously shown that electrocardiogram (EKG) signals can also be used for generating keys between sensors [13]. The approach was similar to this in that frequency domain features are used as a basis for generating keys. However the primary issue with the approach is the collection of EKG signals using the leads on the arms and chest has many usability issues, an important property that we are striving to achieve in our security schemes for BANs.

IX. Conclusions and Future Work

In this paper we presented a novel means of key agreement in a BAN called PPG based Key Agreement Protocol. It allows two sensors to agree on a common key without any initialization or setup. The security analysis and simulation studies show that PPG is a viable option of key agreement in a BAN. In the future we are planning to implement the scheme on actual hardware, while reducing its overheads.

Acknowledgements

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

References

[1] 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. In Proc. of Wireless Security & Privacy Workshop 2003.

[2] 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 Comp. & Comm. Sec.

[3] K. Venkatasubramanian et al. Ayushman: A Wireless Sensor Network Based Health Monitoring Infrastructure and Testbed. pages 406–407, June/July 2005. In Proc. of the IEEE Intl. Conf. on Dist. Comp. in Sensor Systems.

[4] S. Choi 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. ISWC.

[5] A. Juels and M. Sudan. A Fuzzy Vault Scheme. page 408, 2002. In Proc. of IEEE Intl. Symp. on Inf. Theory.

[6] A. Juels and M. Wattenberg. A Fuzzy Commitment Scheme. pages 28–36, Nov 1999. In Proc. of ACM 9th Conf. on Comp. & Comm. Sec.

[7] 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.

[8] 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.

[9] 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.

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

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

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

[13] K. Venkatasubramanian, A. Banerjee, and S. K. S. Gupta. EKG-based Key Agreement in Body Sensor Networks. April 2008. In Proc. of the 2nd Workshop on Mission Critical Networks.

[14] 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.

[15] A Wang and A.Chandrakasan. A 180-mV Subthreshold FFT Processor Using a Minimum Energy Design Methodology. *IEEE Journal on Solid State Circuits*, 40(1):310–319, January 2005.

[16] B. J. West. *Where Medicine Went Wrong: Rediscovering the Path to Complexity*, volume 11 of *Studies of Non Linear Phenomena in Life Sciences*. World Scientific, 2006.

[17] 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.