PEES: Physiology-based End-to-End Security for mHealth

Ayan Banerjee, Sandeep K. S. Gupta[*]
Impact Lab, CIDSE, http://impact.asu.edu
Arizona State University
{abanerj3, sandeep.gupta}@asu.edu

Krishna K. Venkatasubramanian
Department of Computer Science
Worcester Polytechnic Institute
{kven}@wpi.edu

ABSTRACT

Ensuring security of private health data over the communication channel from the sensors to the back-end medical cloud is crucial in a mHealth system. This *end-to-end (E2E)* security is enabled by distributing cryptographic keys between a sensor and the cloud so that the data can be encrypted and its integrity protected. Further, the key can also be used for mutually authenticating the communication. The distribution of keys is one of the biggest overheads in enabling secure communication and needs to be done is a *transparent* way that minimizes the cognitive load on the users (patients). Traditional approaches for providing E2E security for mHealth systems are based on asymmetric cryptosystems that require extensive security infrastructure. In this paper, we propose a novel protocol, **Physiology-based End-to-End Security (PEES)**, which provides a secure communication channel between the sensors and the back-end medical cloud in a transparent way. PEES uses: (1) physiological signal features to hide a secret key, and (2) synthetically generated physiological signals from generative models parameterized with patient's physiological information, to unhide the key. Moreover, in PEES authentication comes for free since only sensors on the user's body has access to physiological features and can therefore gain access to the protected information in the cloud. The analysis of the approach using electrocardiogram (ECG) and phototplethysmogram (PPG) signals and their associated models demonstrate the feasibility of PEES. The protocol is light-weight for sensors and has no pre-deployment or storage requirements and can provide strong and random keys ($\approx$ 90 bits long). We have also started clinical studies to establish its efficacy in practice.

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and Protection; C.2.1 [**Network Architecture and Design**]: Wireless Communication

1. INTRODUCTION

Lifelong monitoring of health has been recently prescribed as an effective remedy to potentially life threatening diseases that have congenital roots, such as congenital heart diseases [1]. Mobile healthcare (mHealth) is a technological oasis that promises the feasibility of lifelong monitoring. In mHealth systems, a network of wireless medical *sensors* and actuators are deployed on a person (also referred to as the *user*), for enabling pervasive, individualized, and real-time health data collection, diagnosis, and critical actuation. The storage, computation, and visualization of the huge amount of data collected by the system is enabled by the massive computation resource of a *medical cloud* (referred to as the cloud from now on). The sensors may forward data to the cloud either directly or through an intermediate base-station. Caregivers and the user can view the collected health information directly from the cloud using a smart-phone app or over the web in real-time and act on it as required. As mHealth systems deal with personal health data, ensuring information security, especially over the communication channel from a sensor to the cloud, is very critical. Lack of adequate security capabilities may not only lead to a breach of patient privacy, but also potentially allow attackers to compromise patient safety by modifying actual physiological data, resulting in wrong diagnosis and treatment [6, 18]. Protecting personally identifiable health data is also a legal requirement as per the Health Insurance Portability and Accountability Act (HIPAA) (http://www.hhs.gov/ocr/hipaa/). Thus, with the possibility of lifelong monitoring comes the requirement of lifelong security.

It is understood that the properties of confidentiality, integrity and authenticity need to be preserved as the health data in a mHealth system is transmitted from the sensors that measure them to the medical cloud, which stores and processes the data. One way of enabling this end to end security is to distribute cryptographic keys between a sensor sender and the cloud receiver. The data can now be encrypted and integrity protected, while the presence of the appropriate key proves the authenticity of the communicating entities. In the rest of the paper, we assume that the keys used for communication in our system model are symmetric cryptographic keys. Although asymmetric cryptography based on Elliptic Curves have been used for communication in a sensor network domain [7, 8], it is still much more expensive to use them for regular data exchange. Additionally, they are prone to man-in-the-middle attacks and need additional authentication mechanisms to be useful. This distribution of symmetric cryptographic keys is one of the biggest overhead in communication security.

Our approach to establishing a secure communication channel from the sensors to the cloud relies on the *end-to-end argument* [15]. In many traditional approaches, secure communication in mHealth settings requires securing two hops individually. The first one is from a sensor to the base-station. If the the sensors form a multi-hop network, then we have one additional step — securing inter-

[*]The works of Ayan Banerjee and Sandeep K.S. Gupta are supported by the NSF grants CNS-0831544 and IIS-1116385.

sensor communication. Once the data reaches the base-station, it then has to be securely transmitted to the cloud. This is usually done with some form of asymmetric cryptosystem. The problem with this hop-by-hop approach is that it is too cumbersome to manage. We need to secure at least two (three if one considers inter-sensor links) individual links each of which has different properties and involve heterogeneous devices with different capabilities. Further, any solution that requires the base-station to play a role in secure transfer of user health data to the cloud is fraught with problems especially because such base-station, usually external to the user, can be compromised. This is not to say that the base-station should be eliminated, because that would mean the sensors would need the capability to directly communicate with the cloud, which might not be ideal in all situations. What we argue for, is that secure transfer of data from a sensor to the cloud should not depend on the base-station. Hence, we need a security solution that establishes an end-to-end (E2E) communication channel between a sensor and the cloud. That way, even if the intermediate communication channel or nodes are compromised, there is minimal loss of sensitive medical data.

In this paper, we propose **Physiology-based End-to-End Security** (PEES), which provides E2E key distribution in a mHealth setting between a sensor and the cloud with minimal user/administrator involvement. It requires no *a priori* distribution of keying material. Simply deploying the sensors on a user is enough, thus facilitating secure E2E communication that is transparent to the user itself. In PEES sensors use physiological signal based features to hide the keying material through a cryptographic primitive called the *vault*. At the cloud, the vault is opened with a *diagnostically equivalent* physiological signal time-series generated using a generative-model that has been parameterized with the user's physiological information [11]. The idea of using physiological-signal-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 [20]. Any sensor without access to the vital signs of the user or a model of the signals will be unable to update or access the user's data in the cloud. The successful execution of PEES automatically authenticates the communicating entities (i.e., sensors and the cloud). In our previous work, we proposed a secure inter-sensor key agreement approach based on physiological signals [18]. However, the technique only enabled two sensors sensing the same physiological signals to communicate securely. In this paper, we propose a technique for establishing a secure channel between a sensor and the cloud, which is not privy to the user's physiological data, but has access to a trained model. In designing PEES, we aspire to meet the following design goals:

- *Cryptographically Strong keys:* distribution of keys that are random and long.

- *Secure Key Distribution:* distribution of keys between a sensor and the cloud such that there is no leakage of keying information.

- *Long term security:* maintaining freshness of keys between a sensor and the cloud for a long term and providing the ability to add and remove sensors without interruption in monitoring.

- *Minimal user involvement:* execution of the key distribution with minimal user involvement (i.e., *transparently*) as the users of this system are not expected to be tech-savvy.

The *contributions* of this paper are three fold: (1) a scheme, PEES, for E2E key distribution between sensors and the cloud that is secure and transparent to the users, (2) analysis of PEES' feasibility and security properties and (3) validation of PEES, using actual data from two of the most commonly collected physiological signals: photoplethysmogram (PPG) and electrocardiogram (ECG).

2. SYSTEM MODEL

The system model for providing mHealth services considered in this paper is shown in Figure 1. At the core of the system is a set of wireless *sensors* that are either worn on or implanted in the user. The sensors may be invasive e.g., glucose meters, contact-based (therefore less invasive) e.g., ECG or PPG, or environmental such as temperature and humidity monitors. Actuating devices such as infusion pumps, can also be used in mHealth. However, we do not consider them explicitly for this work to keep the discussion simple. The sensors sense physiological as well as environmental signals at a given sampling rate. The goal of the system is to collect data from the sensors and forward them to a *medical cloud*. In general, mHealth systems may have two configurations:

- *Configuration 1:* The mHealth sensors are equipped with a WiFi or cellular radio so that they can have direct communication with the medical cloud. This configuration can be used in monitoring mobile patients in a hospital or in a home environment, where relatively capable sensors are used for monitoring and the patients are not particularly ambulatory.

- *Configuration 2:* This configuration includes an extra device in between the sensors and cloud called the *base-station*. The base-station can be implemented on a variety of devices from generic smart phones to customized dongles [2]. The second configuration is useful for monitoring or in rehabilitation for patients who are not confined to their homes or a care facility. The base-station in an mHealth system, can be used to perform one or more of the two following tasks: (a) forward the data collected from the sensors to the cloud for storage and processing, and (b) visualize the health data in a smart phone based base-station in a meaningful manner.

In both configurations, a caregiver has to download the data from the cloud for reference, diagnosis and treatment.

2.1 Trust and Threat Model

We now present our trust assumptions along with assumptions regarding the *attackers* i.e. the threat model:

- *Sensors:* All the sensors in our mHealth system are assumed to be trustworthy. That is, it is not possible for attackers to compromise an existing sensor within the system without the patient noticing.

- *Communication Links:* The communication links within our system are not trusted. We assume attackers can passively eavesdrop (sniff) on all communication and can perform complex signal processing on physiological signals. However, any brute force attack is still time consuming for the attacker. Further, the attacker can actively introduce bogus data into the network. However, we assume that there are no jamming and denial-of-service attacks, where legitimate devices cannot communicate with each other. We make this assumption because such an attack would be detected quickly and we assume our attackers would employ stealthier techniques.

- *Base-station:* Even though the attacker may not be able to physically compromise the sensors we assume they can compromise the base-station itself. If the base-station is a smart-phone then the attackers can compromise the apps on it as well.

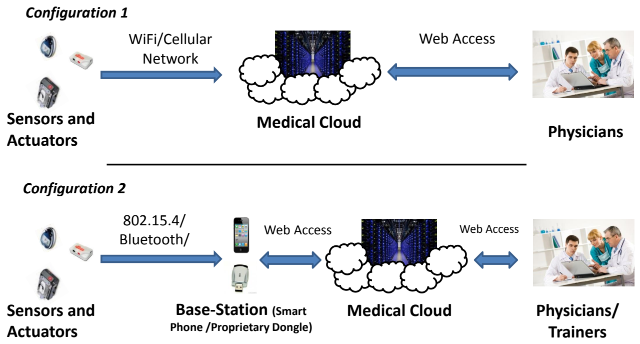

Figure 1: System model for mHealth security.

- *Cloud:* The medical cloud is assumed to be trustworthy. Caregivers are provided information about the patients from the cloud only upon their successful authentication.

- *Caregivers/ Patients:* They are assumed to the trustworthy not influenced by the attacker.

- *Patient's Body:* We assume the attacker can come physically close to the mHealth user and can also have physical contact (e.g., shake hands with the patient) so that the electrical signals of the user can get coupled with those of the attacker [3]. However, the attacker cannot introduce malicious sensors into our system. Further, we assume that *no patient health data* from the past or features derived from it are known to the attacker.

3. PROBLEM STATEMENT

In this paper we consider the problem of assuring E2E security of physiological data transfer for mHealth systems. That is, we want to ensure the confidentiality and integrity as the patient data gets transmitted from a sensor to the cloud. The approach we take in this regard is to establish a pair-wise symmetric key between the sensors and the medical cloud. Once the key distribution has happened, then E2E security can be enabled by encrypting data at the sensors and decrypting them at the cloud. One of the simplest approach is to explicitly program the sensors and the cloud with appropriate cryptographic keys. This can be problematic as it requires considerable configuration of the sensors as they are introduced within the system. Further, we envision a future, where the sensors are purchased over the counter and added to the mHealth system on the go and still be able to perform secure data collection. As mHealth systems will typically be used by people without technical or security training, we want to develop security solutions that are *transparent* to the users and require minimal configuration.

Approach: We approach this problem (Section 5.1) based on two observation (1) features derived from certain physiological signals are complex, dynamic and unique enough to be useful for hiding the keying material, (2) certain physiological signals can be synthetically generated using generative models when appropriately parameterized with user health information based features. The model parameters have to be initially transferred securely to the cloud (discussed in Section 5.2). The model in the cloud however, is not static and hence requires regular updates, which also needs to be tackled in a secure manner (Section 5.3). We instantiate the proposed protocol using two types of signals, ECG and PPG and validate our basic hypothesis of using models and physiological signals to achieve E2E security (Section 6).

4. PRELIMINARIES

In this section we focus on some important concepts that our end-to-end security solution leverages - physiological signal-based key agreement and generative models of physiological signals.

4.1 Key agreement using physiological signals

The variability in the human physiology can be used to derive fresh cryptographic keys for secure communication between two sensors [4, 18, 19]. Sensors sensing the same physiological signal e.g., PPG sensors on the left and right arms or different leads of ECG sensor, can use common *physiological signatures* to hide and unhide a secret key. In this protocol one sensor, the sender, generates a random key and hides it using frequency-domain features generated from recently measured physiological signals with cryptographic construct called the *vault*. The vault is then transferred to the other sensor, called the receiver, which then uses its own set of frequency-domain features generated from concurrently measured (with the sender) physiological signals to un-hide the random key.

The key hiding using physiological features is a light-weight signal processing algorithm that executes at the sender [18]. The sender senses physiological signals for a given time and derives frequency domain features. The sender then generates a random 128 bit key and splits it into $n + 1$ coefficients of a nth order polynomial. The features are then transformed using the polynomial to form a set of ordered pairs (x, y) of feature values and their polynomial evaluations. This set is then obfuscated with random pairs (x', y') called chaff points, such that y' is not the polynomial evaluation of x'. The ordered pairs and the chaff points together form the *vault*. This vault is then sent to the receiver, which has its own set of 16-bit features generated from concurrently measured physiological signals. As long as the receiver has more than $n+1$ features in common with the sender, it can use Lagrangian interpolation to reconstruct the polynomial and obtain the secret key from its coefficients. Since there is a high degree of commonality between the physiological features derived by the two sensors that measure a physiological signal concurrently, the receiver is successful in deriving the secret key from the vault. However, if this vault is received by an attacker who does not have access to the patient data, it has to go through all possible combination of $n + 1$ points out of total number of points in the vault which is combinatorial in order. For example with a 9th order polynomial and 4000 point vault, the complexity for the attacker to break the vault is equivalent to brute-forcing a 95-bit key. In this paper, we use this result to propose an E2E security scheme using generative physiological models.

4.2 Generative physiological models

Generative models of physiological signals are mathematical functions, which take personalized temporal and morphological parameters as input and output synthetic physiological signals, *diagnostically equivalent* to actual physiological signals [11, 12]. A generative model requires two types of parameters - temporal and morphological. The *temporal parameters* change frequently over time. They may include physiological parameters such as the heart rate and the standard deviation of the heart rate. Despite the considerable dynamics of the human body, an important characteristic of human physiology is the periodicity of the waveform of its various physiological signals. The waveform shape within a period is called the *morphology* of the signal. Typically, a generative model expresses the morphology by using a set of mathematical functions. The parameters of this function are called *morphology parameters*. It has been observed that for the ECG and the PPG signals the morphology parameters change very slowly over the lifetime of a person and hence is a physiological signature [10].

To use a generative model for synthesizing physiological signals the morphology parameters have to be learned from a sample of the actual physiological signal. The temporal properties too have to be obtained from the actual physiological signals, but obviously in real time. Finally, though generative models produce *diagnostically equivalent* signals, the synthesized and actual physiological signals may not match sample for sample. They only match in certain features deemed useful for diagnosis of critical health problems as suggested by a physician. In this section, we will briefly discuss generative models for both the ECG and the PPG signals developed in our previous work.

Numerous generative models for various physiological signals have been proposed. In this paper we use two models, one for the ECG and another for the PPG signals. For the ECG we use the well accepted ECGSYN model proposed by McSherry et al [10] while for the PPG we use the DE-PPG model [12].

ECGSYN uses *inter-beat temporal variability parameters*, which includes the mean heart rate, standard deviation of heart rate and LF/HF ratio as temporal parameters. For morphological parameters, ECGSYN represents each of the P, Q, R,S, and T waves of ECG by a Gaussian curve. Each curve has three parameters and hence, there are a total of 15 morphological parameters (a_P, a_Q, a_R, a_S, a_T, b_P, b_Q, b_R, b_S, b_T, θ_P, θ_Q, θ_R, θ_S, θ_T). The ECG curve is expressed using Equation 1.

$$\frac{dECG(t)}{dt} = -\sum_{i \in P,Q,R,S,T} a_i(2\pi hr_{mean}t - \theta_i)e^{(\frac{-(2\pi hr_{mean}t-\theta_i)^2}{2b_i^2})}, \qquad (1)$$

where hr_{mean} is the mean heart rate of the person. To obtain the parameters of ECGSYN for a given user, a set of 256 inter-beat interval values are obtained from the given ECG data. To calculate the LF/HF ratio, the Power Spectral Density (PSD) of this set is computed. The Low Frequency (LF) and High Frequency (HF) components is then obtained by integrating the PSD over the ranges (0.04Hz - 0.15Hz) and (0.15Hz - 0.4Hz) respectively. The ratio between these components gives the value of the *lfhfratio* parameter. The *hrmean* and *hrstd* values are obtained by averaging and computing the standard deviation on the set of R-R interval values, respectively. Among the morphology parameters, ($\theta_P, \theta_Q, \theta_R$, θ_S, and θ_T) are calculated by detecting the relative locations of the P, Q, R, S and T peaks respectively. The remaining parameters are calculated through curve fitting using a mean squared error minimization approach.

The DE-PPG model characterizes the shape of a PPG pulse using differential equations, and is based on a Windkessel model of the human vascular system [5]. The signal is split into two parts - systole and diastole. The diastole is modeled using the equation $PPG_{dias}(t) = a_1 + a_2e^{-a_3t} + \frac{1}{a_4+e^{(-a_5t-a_6)}}cos(a_7t + a_8)$. For the systole, an analytical driving left ventricular pulse waveform is considered, using a single logistical function, as $PPG_{sys}(t) = \frac{1}{a_9+e^{(-a_{10}t-a_{11})}}$. The coefficients [$a_1$, a_2, ... , a_{11}] in the above equations are the morphological parameters. The temporal parameters include the mean heart rate, standard deviation of heart rate and the LF/HF ratio.

5. E2E SECURITY FOR MHEALTH

In this section, we present *Physiology-based End-to-End Security* (PEES), a scheme that establishes a secure communication channel between a sensor and the medical cloud in a transparent manner. The idea is to use the complexity and randomness of the physiological signals form the human body to make sensors agree on a secret cryptographic key with the cloud. In our previous work [18,19], we utilized synchronously measured physiological signal-based features to enable key agreement between two sensors on the patient's body. The entire process of key distribution is transparent, as the user simply needed to deploy the sensors, and the key distribution happens automatically, in a plug-n-play manner. However, both sensors were required to be located on the user's body, so they could measure the same underlying physiological signal and perform key agreement. When it comes to E2E security however, the medical cloud is not privy to the physiological signals. In such a setting our original scheme has to be transformed to provide E2E key distribution while maintaining its transparent nature. This is achieved by use of generative models at the cloud.

5.1 Physiology-based End-to-End Security

PEES works by first measuring the physiological signal of choice, extracting features from it and using the features to create a vault as described in Section 4.1. This vault is then transmitted to the cloud, which tries to open it with physiological features from synthesized physiological time-series obtained using a generative model of the physiological signal. These generative models output synthetic signals that are diagnostically equivalent to the original physiological signals and can be used to generate features that are common enough with the sender to be able to open the vault.

More formally, let p_i represent the time-series of a physiological signal i. Let G_i be the generative model of the signal i. G_i takes as input the time-domain features, f_i, (e.g, heart rate variability for ECG or PPG) and morphological features m_i, (e.g., parameters of equations in Section 4.2) and a time t as input to generate the physiological signal value at time t. Thus, the function $G_i(f_i, m_i, t)$ represents the synthesized signal at time t. The generative model is pre-loaded at the cloud and is parameterized with the user's physiological time-domain and morphological features. We will see in the next section that this can be done relatively easily and securely. Given the physiological signal of choice and its generative model, the following steps are performed by a sensor on the user's body to perform key distribution between itself and the cloud:

1. Sample the physiological signal p_i from time t to $t + \Delta t$ and apply a transformation to obtain current *physiological signature* of user $S_{sender} = Tr(p_i(t \dots t + \Delta t))$. This transformation consists of FFT computation, peak detection of the FFT series and quantization of the peaks.
2. Generates a random key K_s of arbitrary length (128-bits).
3. Divide the key into $q + 1$ equal parts $c_0 \dots c_q$ where q is the order of a polynomial previously agreed, in the open.
4. Compute the polynomial $T(x) = c_0 + c_1x + c_2x^2 \dots c_qx^q$ at each signature point $s^j_{sender} \in S_{sender}$ and obtain a set of ordered pairs $\{s^j_{sender}, T(s^j_{sender})\}$.
5. Obfuscate this set of "legitimate" pairs by adding a large number of "chaff" pairs $\{ch_1, ch_2\}$ such that $ch_1 \neq T(ch_2)$, to create a *vault*.
6. Transfers this *vault* to the cloud, either directly or through the base-station.

The medical cloud, upon receiving the vault, performs the following steps to retrieve the key:

1. Generate a Δt long synthetic signal $G_i(f_i, m_i, t) \dots G_i(f_i, m_i, t+\Delta t)$ with the current time-domain features.

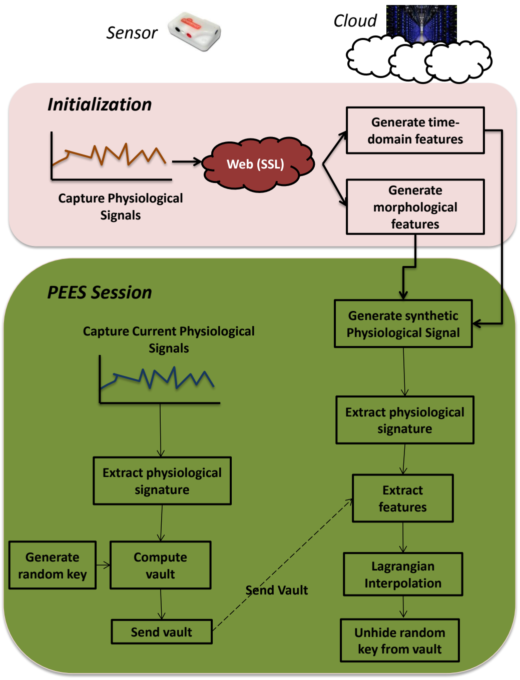
Sensor
Cloud
Initialization
Capture Physiological Signals
Web (SSL)
Generate time-domain features
Generate morphological features
PEES Session
Capture Current Physiological Signals
Generate synthetic Physiological Signal
Extract physiological signature
Extract physiological signature
Extract features
Generate random key
Compute vault
Send Vault
Lagrangian Interpolation
Send vault
Unhide random key from vault

Figure 2: Physiology-based End-to-End Security.

2. Apply the same transformation as the sender to obtain current physiological signature of the mHealth user $S_{receiver} = T(G_i(f_i, m_i, t \dots t + \Delta t))$ from the synthesized signal.
3. Computes the intersection of the sets $S_{receiver}$ and the set $\{S_{sender} \bigcup \{ch_1 \dots ch_N\}$, where N is the number of chaff points.
4. if $\|S_{sender} \bigcap S_{receiver}\| \geq q + 1$, then the receiver has enough number of $\{x, y\}$ pairs to derive the polynomial coefficients $c_0 \dots c_q$ by using Lagrangian interpolation [18].
5. The cloud then concatenates the coefficients of the regenerated polynomial to obtain the key K_s.

5.2 Initializing Generative Models

The most important factor in using generative models for opening the vault is to parameterize them. For example, generative models of ECG and PPG require time-domain and morphological features as suggested in Section 4.2. This can be computed off-line and provided to the cloud when the model is initialized, or one can send a sample of physiological signal timer-series to the cloud, which can then derive the morphological feature values. Either way, when the user purchases a sensor, we require them to use a sensor to collect the physiological signal sample long enough to derive the model inputs and then upload the features to the cloud over a secure web-connection. It is very easy to have a tool-chain available for the user to "initialize" a sensor in this manner. Many monitoring technologies such as runner monitors (`http://www.garmin.com`), use such a setup to upload their running data to the cloud very easily. Contrary to such existing systems, we expect the user to initialize the generative model only once. As this initialization will be done in the confines of the user's home or care facility, we assume that the initial physiological time series is securely transmitted to the cloud. After the initial transmission any future E2E key distribution can be done transparently. To illustrate the initialization process, we consider a scenario where a user goes to a doctor's office for installing an ECG sensor as shown in Figure 3. We assume that the medical practitioner is trusted and has an account in the cloud server at the time of initialization of an electronic health record of the user. The initialization process consists of the following steps:

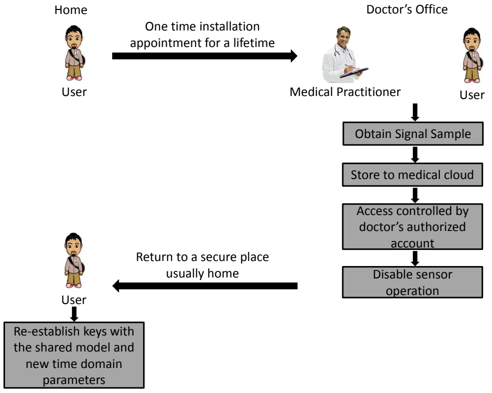
Home
One time installation appointment for a lifetime
Doctor's Office
User
Medical Practitioner
User
Obtain Signal Sample
Store to medical cloud
Access controlled by doctor's authorized account
Return to a secure place usually home
Disable sensor operation
User
Re-establish keys with the shared model and new time domain parameters

Figure 3: PEES initialization process.

1. The medical practitioner samples the physiological signal of the user.
2. The practitioner then uses his authenticated cloud server account to transfer a signal sample to the cloud.
3. The cloud uses this sample signal to automatically learn the model for the user.
4. A sensor and the cloud then automatically perform PEES to establish the first secure key.
5. Once this initialization process is done, the security key can be refreshed by executing PEES as and when needed.

5.3 Changing Model Parameters

PEES needs to store a generative model of the physiological signal at the cloud. The physiological signal sensed by a sensor may however drift from the signal generated by the stored model. Therefore the model parameters for the generative models are not static and tend to change over time. This may happen due to pathological conditions such as arrhythmia [11], or after a surgery. Therefore, for a future re-keying between a sensor and cloud one needs to ensure that the model parameters at the cloud are current. This can be accomplished at run-time for our system. Once the initialization has been done as described in the previous section, the sensors will forward their latest measurements using the secure channel thus established. The measurements will be continually compared by the cloud with the synthetic time-series generated by the cloud. If the actual time-series varies significantly from the generated one, the model parameters are re-learned. This way the model parameters are always in synch with the current state of the patient's physiology and re-keying can be done as needed. Finally, once a model has been loaded onto the cloud, adding or replacing sensor(s) measuring the same physiological signal can be done seamlessly and does not require any change to the cloud.

6. VALIDATION

The proposed approach is based on the hypothesis that physiological signals and their models have enough commonality in order to achieve secure key agreement. We validate this hypothesis for two physiological signals the ECG and the PPG and evaluate the feasibility and strength of PEES. We then move on to analyze the security of PEES.

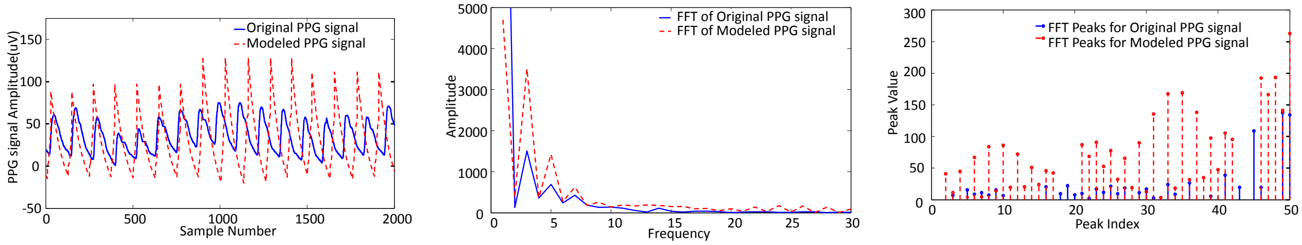

Figure 4: (a) Original and Modeled PPG signal (b) Original and Modeled PPG signals in frequency domain (c) FFT peak features (for vault locking/unlocking) using Original and Modeled PPG signals.

6.1 Feasibility Analysis of PEES

The first step in evaluating PEES is to check if generative models can produce good enough physiological signals such that features derived from them can open the vault. All our evaluations were done using two data-sets: (a) MIT BIH database [11] and (b) IMPACT database [12]. In all there were 20 mHealth users in the study. We have also started testing the proposed E2E security protocol in a realistic use case with patients in an ICU (Section 8).

In our study with ECG data, we found that the average size of the intersection between the physiological signatures obtained from the actual data and the model supplied with current values of time domain features of the same person, $||S_{sender} \bigcap S_{receiver}||_{avg}$ was 8, with a most likely value of 8. This means that key distribution can be performed between a sensor and the cloud using PEES with a 7^{th} order polynomial. However, if the model is supplied with the wrong values of the time domain features f_i, then the average size decreases to 4 and a most likely value of 1. This drastic drop in intersection size can be obtained by a 5% change in time domain feature values. For PPG data the intersection size with correct time domain features was even higher (around 10) with a most likely value of 8. Figures 4 and 5 shows the ECG or PPG data along with their respective synthesized signals, the physiological signatures of the data and the model, and the commonality among the two signatures in sequence. This shows that PEES is feasible and can be used for E2E security.

6.2 Security Analysis of PEES

As discussed in Section 2, we assume that the attacker does not have access to any old data or model parameters. The attacker starts monitoring the network when a sensor is first plugged on to the body. The attacker can successfully retrieve data in three ways:

1. Get access to encrypted data and brute force the secret key that was exchanged using PEES.
2. Get access to the model parameters and use the model to break the vault.
3. Get access to a vault and brute force the entire vault to find out the physiological signature.

We evaluate each attack with respect to the computational complexity of performing the attack. We will quantify computational complexity in terms of the computation required to brute force a secret key. The size of the key, $||key||$, is indicative of the maximum number of combinations, $||key||!$, that the attacker has to try before it gets the correct key. In this regard, brute-forcing the secret key (which we assume is at least 128 bits long) is intractable for the attacker. Even if the attacker is able to brute-force the data, a long enough time would have passed and the attacker will only have a snippet of *old* physiological data.

This brings us to the next mode of attack where model parameters are available to the attacker. To evaluate the feasibility of using stale physiological data for breaking a current vault, we consider performing PEES with physiological data and model generated data with delayed time domain features. Initially the time delay was kept to 0s so that PEES will be successfully executed in close to 100 % of the cases. As the delay is increased the number of cases for which PEES is successfully executed decreased drastically. We found that a 22 second delay in time domain features can cause the PEES success rate to drop from 100 % to 70 % and a three minute delay yields a success rate of 3 %. Figure 6 show the PEES success rates for ECG for different time delays. For PPG the drop in success rate is much more drastic from 50% for a 22 second delay to 0.1% for a 3 minute delay.

Finally, the strength of security comes from the difficulty in breaking the vault. The attacker can brute force the vault to get at least $q + 1$ common points in the physiological signature. In this attack the attacker gets access to a vault and takes $q + 1$ elements from it and performs Lagrangian interpolation to obtain the key. In the worst case the attacker has to perform $\binom{N+||S_{sender}||}{q+1}$ Lagrangian interpolation computations, where N is the number of chaff points. Thus, the computational complexity increases combinatorially with the increase in the number of chaff points and the polynomial order.

For a fixed polynomial order there can be *false negatives*, i.e., models with wrong parameters can have enough common points in the physiological signature leading to a security breach, or *false positives*, i.e., models with correct parameters may not have enough common points leading to a denied access. Figures 7 and 8 show the false positive and negative rates for different polynomial orders. We see that as the polynomial order increases the false negatives decrease but the false positives increase. Ideally we would want to minimize both the false positives and negatives and from both the figures we see that there is a “saddle" point where both gets minimized. However, if we put forth the security of the system as our prime objective rather than accessibility, we can sustain a high false positive rate for a low false negative rate (≈ 0.05). We see that for both ECG and PPG a polynomial order of 9 has very low false positive rates. We also observed that on an average the size of S_{sender} is ≈ 30 for both PPG and ECG. If we consider 4000 chaff points in the vault then in the worst case the attacker has to perform $\binom{4030}{9}$ combinations which is equivalent to brute forcing a 90 bit

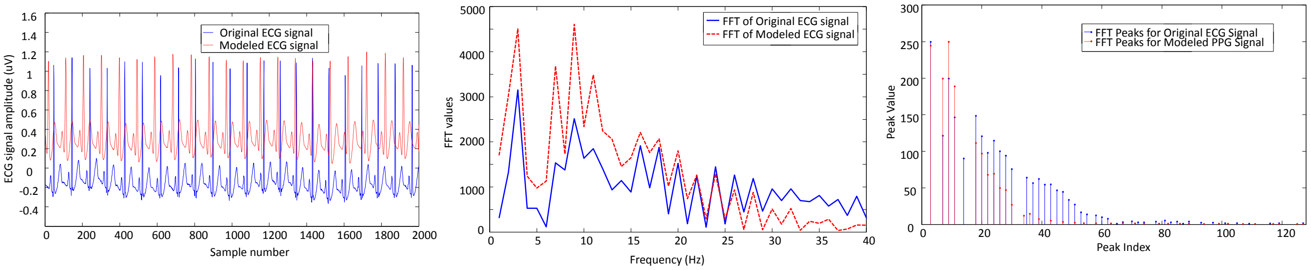

Figure 5: (a) Original and Modeled ECG signal (b) Original and Modeled ECG signals in frequency domain (c) FFT peak features (for vault locking/unlocking) using Original and Modeled ECG signals.

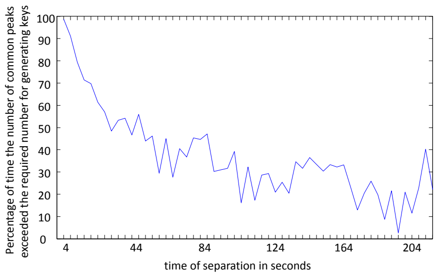

Figure 6: Decrease in number of common peaks with increase in time delay.

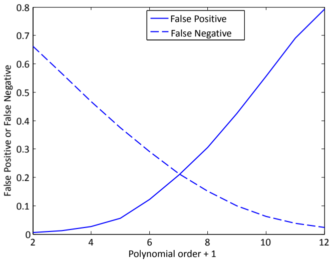

Figure 7: False negatives and positives for PPG signals.

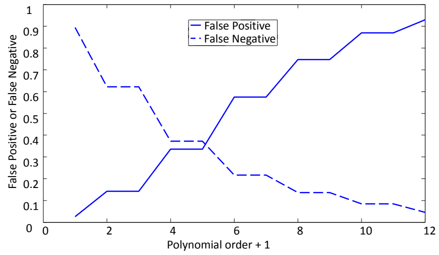

Figure 8: False negatives and positives for ECG signals.

private key. However, since with increasing polynomial order the false positives increase, the cloud has to perform the PEES computation more number of times to retrieve the key. However, this number is of the order of 10 computations as opposed to 2^{90} for the attacker. Given the massive computation capability of the cloud such computations may be considered light-weight. Note that the false positive here simply means the receiver (cloud) has to try more combination of points in the vault, and not a complete shut-down of the protocol itself.

7. RELATED WORK

Typically in mHealth systems researchers focus on securing every communication link separately [14, 17]. Solutions require maintaining dedicated public and private keys for sensor-to-sensor, sensor-to-smartphone, and smartphone-to-cloud [14]. This introduces significant overhead in the deployment of the mHealth system and also blocks storage in already resource constrained sensors. Several solutions have also proposed introducing additional hardware in the mHealth system solely dedicated to achieve security among different entities [16]. Although such an approach can provide security as well as interoperability in a heterogeneous sensor setting, the requirement for an additional device is invasive. To enable non-invasive plug-n-play security, no PKI key storage and no pre-deployment overhead, researchers have proposed physiological signal based key agreement [4, 13, 18, 19]. However, such an approach is only limited to securing inter-sensor communication and cannot provide E2E security. PEES overcomes the limitations of these approaches by providing transparent E2E secure communication channel between a sensor and the cloud. It can also be used to secure individual communication links in a mHealth setting.

8. DISCUSSIONS

In this paper, we have shown that models and physiological data can collaborate to provide plug-n-play E2E security. However, in our approach we have considered that the attacker cannot get access to time domain features. Such assumptions are not often true and there are several non-invasive ways such as electromagnetic coupling to obtain traces of current time domain signals without physically placing a sensor. Further, using the MIT BIH and IMPACT data base we only showed the feasibility of PEES. A thorough clinical study is required to show the effective operation of PEES in practice. We discuss these two issues in this section.

The heart has a strong electromagnetic field which gets coupled to electrical measurements done in close proximity to a human body. For example, ECG artifacts are often observed in electroencephalogram (EEG) measurements [9]. In our previous work [3], we have demonstrated in very limited cases that an attacker can deduce time domain properties of ECG from its own EEG measurements. If the attacker has a generative model, it can then break PEES. However, to be successful the attacker must have access to a generative model and have physical contact with the mHealth user, which can be avoided.

To show practical applicability of PEES, clinical studies in an actual hospital environment are necessary. We have partnered with St. Luke's Hospital in Phoenix, Arizona, to simultaneously deploy medical grade ECG monitors (Holter monitors), to sense sample by sample ECG, and Shimmer sensors, which sense time domain features and synthesize ECG data, on 25 patients for 20 hours each (setup shown in

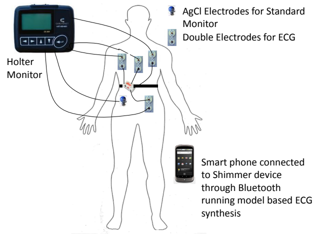
AgCl Electrodes for Standard Monitor
Double Electrodes for ECG
Holter Monitor
Smart phone connected to Shimmer device through Bluetooth running model based ECG synthesis

Figure 9: Clinical study setup.

Figure 9). The configuration shown in Figure 9 is similar to the configuration 2 shown in Figure 1. We have prepared consent documents in both English and Spanish and have also secured Institutional Review Board (IRB) approvals. The data is kept in a secured repository and is only available to the authors and the participating physicians of St. Luke's hospital. After analyzing the data we plan to make the data public in our IMPACT Lab webpage. We were able to successfully execute PEES on a single patient, and the attacks discussed in Section 6.2 were not successful. We are still investigating other patients and the results will be published in future.

9. CONCLUSIONS

In this paper, we proposed Physiology-based End-to-End Security (PEES), a novel protocol that establishes a secure communication channel between a sensor and the medical cloud in a transparent manner. Once the key exchange has happened in this manner, a sensor and cloud can perform secure communication with each other. The idea behind PEES is for the sensors to use physiological signal based features to hide the keying material using a cryptographic primitive called the *vault*. This information is then transferred to the cloud, which then uses a clinically relevant physiological model to unhide the keying material, or open the vault. Although we show the validity of our hypothesis for two signals, we believe that if we have a generative model for a physiological signal then the proposed E2E protocol is generic enough to provide communication security using that signal. PEES' key distribution meets our design goals (Section 1): (1) the keys are long and random, (2) the vault with large enough polynomials and chaff-points is quite secure to prevent information leakage about the key being exchanged, (3) the entire process permits re-keying at anytime, and (4) all this can be done with minimal user involvement. Although the proposed scheme for plug-n-play security bypasses the smartphone, in many cases, however, the smart phone is an important entity for real time applications such as physiological data visualization, or diagnosis. The proposed E2E protocol, can be easily extended to include the smartphone without any extra storage or pre-deployment overhead for the sensors, provided the smartphone is kept secure from physical compromise. We have also started clinical studies to establish its efficient execution in practice.

10. REFERENCES

[1] Monitoring lifelong progress of congenital heart disease, http://arkansasmatters.com/fulltext?nxd_id=641372.

[2] P. Asare, D. Cong, S. G. Vattam, B. Kim, A. King, O. Sokolsky, I. Lee, S. Lin, and M. Mullen-Fortino. The medical device dongle: an open-source standards-based platform for interoperable medical device connectivity. In *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium*, IHI '12, pages 667–672, New York, NY, USA, 2012. ACM.

[3] P. Bagade, A. Banerjee, M. Joseph, and S. K. S. Gupta. Protect your BSN: No Handshakes, just Namaste! In *Intl Conf on Body Sensor Networks*. IEEE, MIT Boston, 2013.

[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, October 2003. In Proc. of Wireless Security and Privacy Workshop.

[5] V. P. Crabtree and P. R. Smith. Physiological models of the human vasculature and photoplethysmography. *Electronic Systems and Control Division Research, Department of Electronic and Electrical Engineering, Loughborough University*, pages 60–63, 2003.

[6] S. K. S. Gupta, T. Mukherjee, and K. Venkatasubramanian. Body area networks: Safety, security, and sustainability. Cambridge University Press, 2013.

[7] A. Liu and P. Ning. Tinyecc: A configurable library for elliptic curve cryptography in wireless sensor networks. In *Information Processing in Sensor Networks, 2008. IPSN '08. International Conference on*, pages 245–256, 2008.

[8] K. Malhotra, S. Gardner, and R. Patz. Implementation of elliptic-curve cryptography on mobile healthcare devices. In *Networking, Sensing and Control, 2007 IEEE International Conference on*, pages 239–244, 2007.

[9] R. McCraty, M. Atkinson, D. Tomasino, and W. A. Tiller. The electricity of touch: Detection and measurement of cardiac energy exchange between people. *Brain and Values: Is a Biological Science of Values Possible. Mahwah, NJ: Lawrence Erlbaum Associates, Publishers*, 1998:359–379, 1998.

[10] P. E. McSharry, G. D. Clifford, L. Tarassenko, and L. A. Smith. A dynamical model for generating synthetic electrocardiogram signals. *Biomedical Engineering, IEEE Transactions on*, 50(3):289–294, 2003.

[11] S. Nabar, A. Banerjee, S. K. S. Gupta, and R. Poovendran. GeM-REM: Generative model-driven resource efficient ecg monitoring in body sensor networks. In *Body Sensor Networks (BSN), 2011 International Conference on*, pages 1–6. IEEE, 2011.

[12] S. Nabar, A. Banerjee, S. K. S. Gupta, and R. Poovendran. Resource-efficient and reliable long term wireless monitoring of the photoplethysmographic signal. In *Wireless Health*, pages 9:1–9:10. ACM, 2011.

[13] 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. *Communications Magazine, IEEE*, 44(4):73 – 81, April 2006.

[14] P. K. Sahoo. Efficient security mechanisms for mhealth applications using wireless body sensor networks. *Sensors*, 12(9):12606–12633, 2012.

[15] J. H. Saltzer, D. P. Reed, and D. D. Clark. End-to-end arguments in system design. *ACM Trans. Comput. Syst.*, 2(4):277–288, nov 1984.

[16] J. Sorber, M. Shin, R. Peterson, C. Cornelius, S. Mare, A. Prasad, Z. Marois, E. Smithayer, and D. Kotz. An amulet for trustworthy wearable mhealth. In *Proceedings of the Twelfth Workshop on Mobile Computing Systems &#38; Applications*, HotMobile '12, pages 7:1–7:6, New York, NY, USA, 2012. ACM.

[17] C. C. Tan, H. Wang, S. Zhong, and Q. Li. IBE-Lite: A lightweight identity-based cryptography for body sensor networks. *IEEE Transactions on Information Technology in Biomedicine*, 13(6):926–932, 2009.

[18] K. K. Venkatasubramanian, A. Banerjee, and S. K. S. Gupta. PSKA: Usable and secure key agreement scheme for body area networks. *Information Technology in Biomedicine, IEEE Transactions on*, 14(1):60 –68, Jan. 2010.

[19] K. K. Venkatasubramanian and S. K. S. Gupta. Physiological value-based efficient usable security solutions for body sensor networks. *ACM Trans. Sen. Netw.*, 6(4):31:1–31:36, jul 2010.

[20] B. J. West. Studies of nonlinear phenomena in life sciences. In *Where Medicine Went Wrong: Rediscovering the Path to Complexity 11*. World Scientific, 2006.