Secret from Muscle: Enabling Secure Pairing with Electromyography

Lin Yang[†], Wei Wang[‡†], Qian Zhang[†]
[†]Department of Computer Science & Engineering, Hong Kong University of Science and Technology,
[‡]School of Electronic Information and Communications, Huazhong University of Science and Technology
{lyangab, gswwang, qianzh}@cse.ust.hk

Abstract

Forming secure pairing between wearable devices has become an important problem in many scenarios, such as mobile payments and private data transmission. This paper presents EMG-KEY, a system that can securely pair wearable devices by leveraging the electrical activity caused by human muscle contraction, that is, Electromyogram (EMG), to generate a secret key. Such a key can then be used by devices to authenticate each other's physical proximity and communicate confidentially. Extensive evaluation on 10 volunteers under different scenarios demonstrates that our system can achieve a competitive bit generation rate of 5.51 bit/s while maintaining a matching probability of 88.84%. Also, the evaluation results with the presence of adversaries demonstrate our system is secure to strong attackers who can eavesdrop on proximate wireless communication, capture and imitate legitimate pairing process with the help of camera.

1 Introduction

Nowadays we are witnessing the fast development of wearable devices. Such rapid growth has led to a prevalence of direct communications between devices in proximity and innovated many promising applications. This includes, mobile payments, which enable users to make a purchase by interacting their mobile devices or smart watches with an electronic payment device [1]; Private data transfer implemented on many commercial off-the-shelf smart wristbands, such as, fitbit [5], can directly transmit user's biological data to an authenticated mobile device or data collection hub in proximity. Along with the wide adoption of these applications are not only the convenience and excellent user experience, but also an increasing concern about privacy and security, as the data transmitted is often highly sensitive and private. As a result, establishing a secure pairing becomes an important problem for wearable devices.

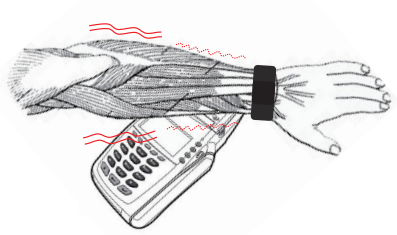


Figure 1. Example application of EMG-KEY on Mobile Payment. The user attaches his arm to the payment device and performs a simple gesture. The EMG signal caused by this gesture can be used to generate secret key to secure further communication.

Since wearable devices often lack convenient input methods and have limited resources, researchers have proposed many novel systems to serve as alternatives to traditional PIN-code-based and cryptographic-based approaches. In these works, the vital part of creating a secure pairing between devices is to ensure both devices obtain consistent and confidential observations from an information source, which allows them to reach an agreement on the same secret key. Such a secret source can be the wireless channel measurement [13, 28, 33, 36, 43], human movements (gesture [10], gait [49], shaking trajectory [38]), ambient environment, *e.g.*, ratio [37], sound [46], or vibration [9].

However, since the characteristics and randomness of the secret source directly determine the robustness of secure pairing schemes, existing works are still exposed to some disadvantages when facing strong attackers. Due to the sharing nature of wireless medium, secure pairing schemes based on wireless channel measurements [13, 28, 33, 36, 43] are vulnerable to predictable channel attacks, in which various hacking techniques can be employed by a malicious adversary, *e.g.*, blocking the Line-of-Sight (LOS) radio propagation between devices, to cause predictable variations in the wireless channel measurement [33]. Also, the secret key generated by movement-based approaches [10, 38, 49] might be attacked if the movement is captured by a camera with motion analysis. Meanwhile, the ambient-environment-based works [9, 46] are threatened by an eavesdropper or ac-

tive attacker who can intentionally controls the ambient environment by making predefined noises or vibrations.

The security limitations of the aforementioned techniques have motivated us to design a more secure pairing system using intrinsic signals which reside in the human body, *i.e.*, the electric activity caused by human muscle contractions. The key insight is that, to perform any human body movement, our central nerve system sends electrical signals to trigger corresponding muscle contractions. Such an electrical signal propagates along with the muscle fibers and can be captured by electrodes placed on the skin. The recorded signal is termed the *Electromyogram* (EMG), which has several promising characteristics. (i) Medical studies [22, 39] have proven that the EMG signal is a quasi-random process. This means the average value of EMG will be statistically larger if we intend to generate stronger force, but the amplitude variation of EMG under a given force value is stochastic in nature. As a result, even if a gesture is imitated and the corresponding output force is estimated, the variation of EMG amplitude is still indeterminable. (ii) The current volume and propagation area of EMG are quite subtle, only physical contact in proximity can sense the signal [22], which means the eavesdropping without physical contact would be extremely difficult, if not impossible. (iii) Fueled by the developments in new human-machine interaction technologies, EMG sensor is being increasingly adopted by many commercial wearable devices, *e.g.*, Myo armband [7], Athos gear [3], and Leo smart band [6]. These facts suggest that EMG signals can be leveraged as a secure source to generate secret key. Such a key can be used by wearable devices to authenticate each other's physical proximity and then to communicate confidentially.

Inspired by this idea, we propose EMG-KEY, a system that securely pairs two wearable devices by using the EMG variation caused by human body movement, *e.g.*, hand gestures, as the secret source to generate cryptographic key. Our system comprises a smart wristband and a smart device equipped with EMG sensors. Through physically attaching these devices to the human body and performing an arbitrary gesture, EMG-KEY can generate secret keys from the captured EMG signals and use them to create a secure communication channel between devices. A typical application of EMG-KEY is the mobile payment[1], in which the transaction data is very sensitive and requires a high security level. As shown in Figure 1, a user touches a payment device with his arm while wearing a smart wristband. He then makes an arbitrary gesture, such as clenching the fist. The EMG signal caused by this gesture will be recorded by the EMG sensors embedded in the smart wristband and payment device. Then, both devices use the captured EMG signal to generate a secret key. As both of their measurements are from the same source, they can reach a consensus on the same secret key with a high success rate while attackers have no clue about this secret key.

To realize such a system, there are several challenges. First, it is not clear whether the randomness of EMG variation is sufficient to generate a robust secret key. To answer this question, we formulate the generation of EMG as a random process model and gain several insights from theoretical study and empirical experiments on volunteers. Another challenge stems from the design of secret key extraction: although both devices involved in the pairing measure EMG from the same source, there are still some inconsistencies in the captured signals due to the different installation locations, electrode attenuation, and hardware imperfections. To address these issues, we design a secret key generation algorithm based on the temporal variation shapes of EMG signals and leverage error correction coding [17] to alleviate the discrepancy. Extensive experimental results have confirmed the effectiveness and efficiency of our algorithm.

Our contributions in this work lay in the following aspects:

- As far as we know, we are the first to explore the possibility of using EMG to enable secure pairing for wearable devices. We have demonstrated that EMG is a good information source to build a secure pairing system due to its physical characteristics and stochastic nature.
- We propose EMG-KEY, a secure pairing system for wearable devices, that can defend against many strong attackers and provide high security. In this system, we design and implement a secret key generation algorithm based on the temporal shape variations of EMG signal and alleviate the inconsistency via error correcting coding.
- We comprehensively evaluate the performance of our system under different scenarios with 10 volunteers. The results indicate that our system can archive a high bit rate of 5.51 bit/s while maintaining a successful pairing rate of 88.84%. Also, the evaluation results, in the presence of adversaries, demonstrate that our system is secure against strong attackers who can eavesdrop proximate wireless communication, capture and imitate the users' pairing process with the help of a camera.

The rest of paper is organized as follows. We first briefly introduce the preliminary theory of EMG generation and investigate its feasibility as a secret source, then define the threat model in Section 2. The system design and detailed implementation are discussed in Section 3. In Section 4, we describe our experimental methodology and evaluation metrics. Then, we present the performance of our secret key generation, impact of confounding factors, and resistance to attacks in Section 5, Section 6, and Section 7, respectively. The discussion and related work are provided in Section 8 and Section 9, followed by a conclusion in Section 10.

2 Feasibility & Threat Model

In this section, we start with a brief introduction to EMG, and then formulate its generation as a random process model. From this model, we can theoretically verify that the randomness of EMG is sufficient for secure pairing. Aside from this, we also conduct empirical experiments on volunteers to demonstrate the feasibility of our system. After that, we discuss our target scenario and define the attack model.

[1]Note that the mobile payment involves many steps, including secure pairing, user authentication and so on. Our system only focus on the pairing part.

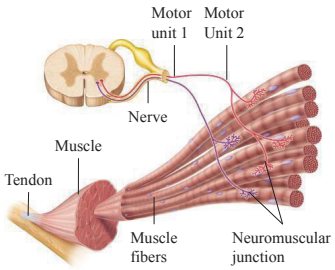


Figure 2. Anatomy of Muscle [35]. Skeletal muscles comprise dozens of muscle fibers, which are innervated by motor neurons.

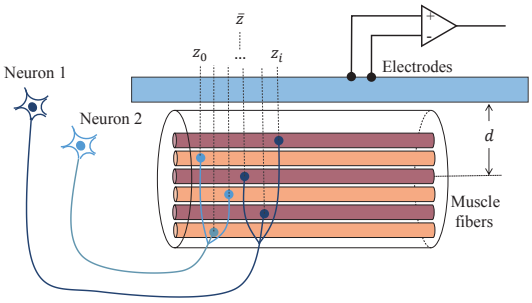


Figure 3. EMG modeling. Two motor units, each of which innervates three muscle fibers at different end-plates z, are presented in this example.

2.1 Preliminary

The generation of physical movement in the human body involves the activation of skeletal muscles [39]. As shown in Figure 2, skeletal muscles comprise dozens of elongated, cylindrical cells known as *muscle fibers*, which are attached to the bones of skeletons via tendons. Each muscle fiber is innervated by a motor neuron and the contact region is termed the *neuromuscular junction*, in which each axon lies in a groove on the surface of the muscle fiber called *motor end-plate*. The motor neuron and the set of muscle fibers it innervates compose the basic functioning unit of a muscle, *i.e.*, *motor unit* (MU).

It is through the contraction of muscle fibers that we form the movement. It starts with an electrical excitation sent from our nerve system to the muscle fibers which activates the acetylcholine-gated channel in the end-plate and allows large amounts of positive sodium to flow into the muscle fiber [14]. This positive influx causes a local depolarization of the fiber membrane and initiates the *muscle fiber action potential*. Such action potential spreads along the muscle fibers innervated by this motor neuron and results in their contraction. The frequency at which the muscle fibers are stimulated by their innervating axon is called the *motor unit firing rate* and multiple motor units are recruited during a movement to meet the requirement of output force.

Through placing electrodes on the skin around the contracting muscle, the electrical activity during a muscle contraction can be captured and the recorded data is termed the *surface EMG signal*.

2.2 EMG Modeling

As a complicated biological process, EMG begins with the nerve impulse sent from motor neuron, which spreads over end-plates and yields the muscle fiber action potential. The action potential propagates along fibers and tissues, and eventually captured by electrodes on the skin. To quantify this process, consider an example shown in Figure 3, in which a set of muscle fibers are innervated by two motor neurons. The contact regions where the axons of neurons meet muscle fiber are labeled as $z_0, \ldots, z_i$, and the mean is z_m. Let d be the average distance between the muscle and skin, and w indicate the spacing between electrodes.

When a motor unit is recruited, the motor neuron sends excitation impulse to initiate the muscle fiber action potential. It is evidenced [39] that the firing pattern of motor neuron is quasi-random, *i.e.*, the average firing rate grows with the increasing force requirement, but the occurrence of each impulse is stochastic in nature; Moreover, the firing patterns of different motor units are essentially independent [22]. Let random function $R_q(t)$ describe the firing pattern of the q-th motor unit. Then, the overall firing pattern of motor units recruited is:

$$R(t) = \sum_{q=1}^{Q} R_q(t) \tag{1}$$

When the nerve impulse arrives at the muscle fiber, it causes the depolarization of the fiber membrane and generates the muscle fiber action potential. This action potential propagates from end-plates to electrodes at a conduction velocity u and can be described as:

$$p(t) = Aut(2-ut)e^{-ut}, \tag{2}$$

where A is a scale factor and u is the conduction velocity, both of which are determined by fiber membrane properties.

However, one may notice that the geographic distribution of end-plates, *i.e.*, the starting points of the action potential propagations, are quite different. This can be viewed as a time shift from z_m and described by the convolution of the delta shift function:

$$D(t) = \sum_{m=1}^{M} \delta(t-\tau_m), \tag{3}$$

where $\tau_m = \frac{z_m - \bar{z}}{u}$ is the time shift caused by the distance between z_m and $\bar{z}$.

Combining these factors, we can quantify the EMG generation using the following model:

$$\begin{aligned} EMG(t) &= \sum_{q=1}^{Q} \Big\{ R_q(t) * D_q(t) * p(t) * e(t) \Big\} \\ &= \sum_{q=1}^{Q} \Big\{ R_q(t) * \Big[\sum_{m=1}^{M_q} \delta(t-\tau_m) * p(t) \Big] * e(t) \Big\}, \end{aligned} \tag{4}$$

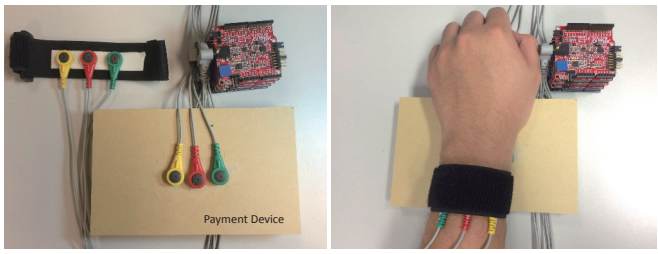


Figure 4. Prototype of EMG-KEY, which consists of a wristband and payment device, both of which are equipped with Olimex EMG sensors and controlled by Arduino UNO board.

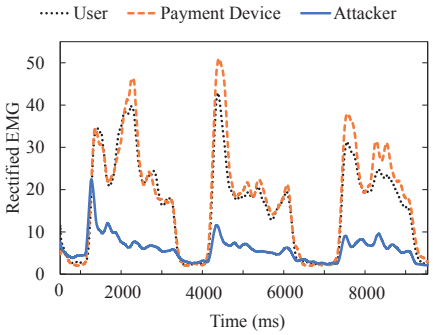


Figure 5. The rectified EMG measurements from the user device and payment device present a high correlation, but they are significantly different from the attacker's signal.

where Q is the number of motor units which are recruited in the contraction, M_q is the number of muscle fibers innervated by the q-th motor unit. The $e(t)$ is the transfer function of electrodes, which is defined by the electronic properties of electrodes and its relative location with respect to the muscle.

From this model, we can gain several useful insights:

- To generate a movement, it often requires multiple motor units to be involved. However, the number of recruited motor units Q is determined by the force requirement. Thus, even under the same movement, the number of recruited motor units can be different.
- Even when the gesture can be captured on camera and the output force might be inferred, the attacker is still agnostic about the user's EMG signal due to the stochastic nature of the firing patterns of motor units.
- The personal difference in the end-plate distribution, conduction velocity of muscle fiber membrane and even muscle fatigue level also introduce additional discrepancies between the EMG signal generated by the legitimate users and attackers.

Apart from these observations, we also find the current volume of EMG signal is quite small (around $\pm 1.5mv$), and propagation area is limited to the skin above the contracting muscles, which implies eavesdropping without close-proximity physical contact is extremely difficult. All of these observations suggest that EMG could be a good randomness source to generate secret keys.

2.3 EMG as Secret Source

To validate the feasibility of using EMG to generate secret key, we build a prototype based on Arduino UNO development board [2] and Olimex EMG shield [8]. As shown in Figure 4, the prototype consists of a wristband and a payment device, both of which are equipped with Olimex EMG sensors.

Similar to the mobile payment scenario, we ask volunteer A to wear our wristband, and put his hand on the payment machine. Meanwhile, volunteer E acts as the attacker, who is also wearing the same type of wristband and can observe every gesture made by user A. To simulate the worst case, both user and attacker are required to perform an easy-to-imitate gesture, that is, slowly clenching the fist and releasing it, which is repeated three times.

Table 1. Pearson correlation coefficient among user A, payment device B and attacker E

corr(A, B)	corr(A, E)	corr(B, E)
0.98	0.69	0.66

Figure 5 gives an example of the rectified EMG signal (for the details of rectification, see Section 3) obtained from the wristbands of user A and attacker E, and the payment device B. The pairwise Pearson correlation coefficients are also present in Table 1. We can notice some interesting observations: (i) Even for the same person, making the same gesture, the EMG measurement can be different each time; (ii) Although some slight differences do exist, the EMG signals recorded from user A's wristband and the payment device are highly similar in their variation shapes and are strongly correlated, evidenced by a correlation coefficient of 0.98. (iii) The correlation between attacker and legitimate devices are not minor (around 0.69). Such a correlation derives from the fact that the attack can clearly observe the gesture and easily imitate it. As the EMG amplitude is a quasi-random process with respect to output force, the general rise and drop trend at the beginning and end of a gesture can easily be imitated, but fail in the matching of the small scale variations of the gesture.

These observations correspond to our insights from EMG modeling in Section 2.2, which provides additional support for the feasibility of using the EMG signal as a secret source.

2.4 Threat Model

In our scenario, two legitimate devices, neither of which have priori knowledge about the other, would like to communicate confidentially. We assume both devices are equipped with EMG sensors. To associate them successfully, the user needs to put them in close proximity (around 4 cm) above the acting muscle and have physical contact with the skin.

For the threat model, we assume there exists a powerful attacker, who knows the exact details of our system and can observe all the gestures made by the legitimate users, or even use a camera to capture it for further analysis. Besides, he

can imitate the same gesture as the user's. Moreover, all the packets transmitted through the wireless channel can be overheard and unencrypted packet will be correctly decoded by the attacker. We term such an attacker the ***copy attacker***.

In such a threat model, the copy attacker can first record the user's gesture and capture all the packet over wireless channel during this communication. As these packets are encrypted with our secret key, he can imitate the user's gesture and generate his own key with the knowledge of our key generation algorithm. Also, if there is any information about the secret key exchanged over the wireless channel, it can be captured by the attacker and used to help the hacking of real secret key. In such way, the copy attacker poses a serious threat to user's data security and privacy.

3 System Design

In this section, we present the design of EMG-KEY in detail. We start with the rectification process and noise removal of the raw EMG signal, introduce the secret key generation, and then move to the discussion on how to alleviate the discrepancies caused by the electrode transfer function and hardware imperfections. Figure 6 provides an overview of our system.

3.1 Pre-processing

As discussed in Section 2.1, the EMG signal can be modeled as the convolution result of the firing pattern of motor neurons, distribution of end-plates, muscle fiber action potential and electrode transfer function. To magnify the effect of neuron firing pattern, rectification is a common applied approach [22]. The Root-Mean-Square-based rectification of EMG signal $x(t)$, is defined as:

$$EMG_{rect}(t) = \sqrt{\Big[\frac{1}{T}\int_{t-T}^{T} x^2(\tau)d\tau\Big]}, \tag{5}$$

where T is the window size which controls the trade-off between smooth envelope against transient variations of EMG signal. In our system, we set this value to be 0.8 seconds.

Also, during the recording of EMG, there are many sources of noise and interference, such as the electrical noise caused by the friction between the electrodes and the skin, or the power line interference. We notice that the most significant noise is either less that 10 Hz (friction noise) or concentrated around 50 Hz (power line interference, the frequency of which can be different among countries). Besides, the majority of arm EMG is above 20 Hz [14]. Thus, a high-pass filter with cut-off frequency of 15 Hz and a notch filter implemented based on Chebyshev IIR filter are adopted to alleviate the interference of the noise . Figure 7(a) and 7(b) show an example of raw EMG and its corresponding rectified signal.

Through applying the rectification and filtering on a raw EMG measurement, we can obtain the rectified EMG. In what follows, we demonstrate how to generate secret key based on the rectified EMG signal.

3.2 Secret Key Generation

The goal of the secret key generation scheme is to fully explore the randomness of EMG signals and encode them into secret bits at a high rate. A common practice to this end is to divide the signal into segments, and encode each segment via quantizing its amplitude into several levels. Although such a method can preserve most information of the signal, the quantification level is not to defined [43] and may introduce many many mismatched bits in our case: as we can observe in Figure 5, the signal amplitudes of legitimate devices are not exactly coincident due to their hardware difference.

Algorithm 1 Shape-based Secret Key Generation.

Input:
Rectified EMG signal S, coding window size w
Output:
Secret bit list $L = [c_0, c_1, ..., c_n]$

$ind \leftarrow 0$
while $ind + w < size(S)$ **do**
  $s = S[ind : ind + w]$, $range = max(s) - min(s)$
  $rise = [min(s) + i * range/w$ for i in $0 : w]$
  $drop = [max(s) - i * range/w$ for i in $0 : w]$
  $stay = [\frac{range}{2}$ for i in $0 : w]$
  Template list $T \leftarrow [rise, drop, stay]$
  $dis \leftarrow \infty$, $c \leftarrow NULL$, $tid \leftarrow 0$
  while $tid < size(T)$ **do**
    $d = fastDTW(s, T[tid])$.
    if $d < dis$ **then**
      $c = toBinary(tid)$, $dis = d$
    end if
    $++tid$
  end while
  $L.append(c)$
end while
return L

However, we notice that even though the EMG amplitudes of legitimate devices are not well matched, their variation trends are highly correlated. Moreover, the variation in the EMG shapes of attacker is significantly different from the legitimate devices. Therefore, we choose to encode the EMG signal by using their variation shapes.

Our encoding algorithm consists of three steps. First, divide the rectified EMG S into small segments of size w. For each segment, we define three basic shape templates, *i.e.*, *rise*, *drop*, and *stay,* according to their amplitude variations. Then, we use *Fast Dynamic Time Warping* [30] to compute the distance between the segment and these three templates and find the best-matching shape template. After that, we use the binary representation of the corresponding template ID tid as the *secret key*. Algorithm 1 elaborates this process.

Let V be the number of shape templates and w define the coding windows size in seconds. Since we can generate $\frac{1}{w}$ segments per second and use the binary representation of the best-matching shape template ID of each segment as secret bits, the bit generation rate (in units of bit/second) can be computed as:

$$\text{bit generation rate} = \frac{1}{w}\log_2 V, \tag{6}$$

where $V = 3$ in our case.

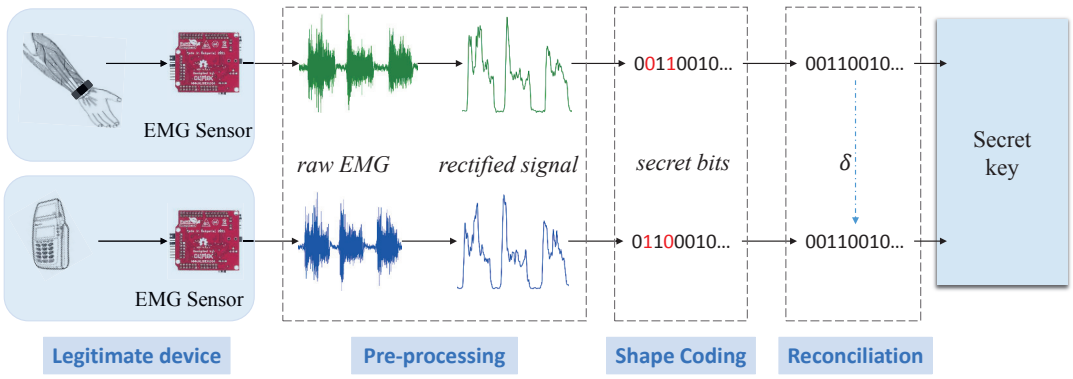
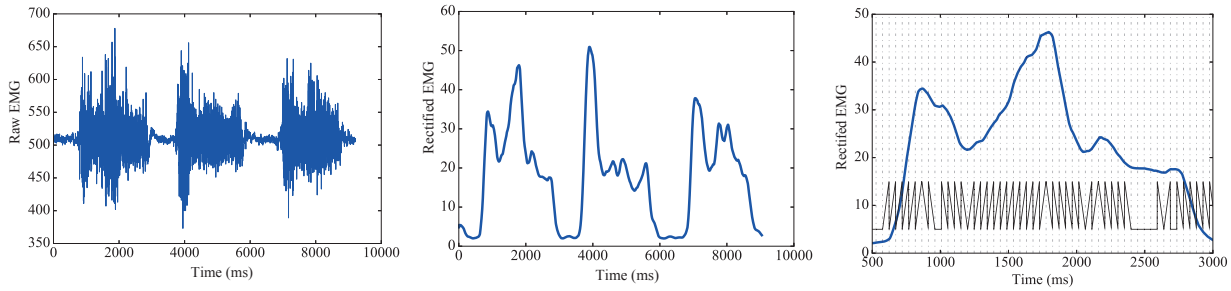


Figure 6. Overview of EMG-KEY.



(a) Raw EMG (b) Rectified EMG (c) Close-up of shape-based encoding

Figure 7. Flow of EMG-KEY.

An example result of the shape-based encoding algorithm is presented in Figure 7(c), in which the blue line is the rectified EMG signal and the black line within each coding window is the approximated shape of the segment.

3.3 Reconciliation

After the secret key generation, each device individually ends up with an n-bit secret key. However, due to the space between the devices and the imperfection in the electrodes' properties, *e.g.*, signal amplification gain and resistance to noise, the transfer function $p(t)$ of each EMG sensor can be different. As a result, there are some discrepancies in the EMG variation shapes, which inevitably lead to mismatching bits among the secret keys.

The purpose of reconciliation is to alleviate the mismatching of the secret keys between legitimate devices. As both of the secret keys of legitimate devices are derived from the same EMG source, they can be viewed as two different distorted versions of the same signal. Through employing the Error Correction Coding (ECC) [17], the number of mismatching bits can be reduced.

Specifically, given two legitimate devices A and B, the secret keys they obtained from secret key generation are k_a and k_b, the mismatching bits between which are defined as ε. Let $C(n,k,r)$ be an error correction code that encodes k-bit message into an n-bit code to resist r-bit random error. Function $f(\cdot)$ and $g(\cdot)$ denote the corresponding encoding function and decoding function. To perform the reconciliation, device A first computes an offset δ between k_a and its corresponding codeword:

$$\delta = k_a \oplus f\big(g(k_a)\big), \tag{7}$$

Then, device A transmits this offset data to device B via a public communication link, *e.g.*, WiFi or Bluetooth. Once device B receives the *delta*, it can deduce k_a as follows:

$$k_a\prime = \delta \oplus f\big(g(k_b \oplus \delta)\big), \tag{8}$$

If the mismatching rate ε can be roughly estimated, an appropriate error correction code C can be leveraged to ensure $k_a\prime$ equals k_a with a high probability.

The rationale is that, with an ECC of correction range of r, any encoded message that is within the correction range to a codeword w will be decoded as $g(w)$. Moreover, exchanging the offset information δ can ensure both k_a and k_b are within the correction range of the same codeword if the distance between k_a and k_b is not larger than r. Therefore, reconciliation process can map two different bit sequences, which have at most r-bit mismatching, to the same key.

We understand this process not only reduces the mismatching bits between the secret keys of legitimate devices, but also leaks partial information about the secret key. Since the δ is transmitted over a public communication link, it may be overheard by an attacker and can be used to help the attack of secret key. However, it can be theoretical proved that there are only $(n-k)$ bits of information leakage occurred [37]. Moreover, since the secret key during is derived from the random variation of EMG signal, the offset information δ in each pairing procedure varies independently. Therefore,

an attacker still cannot infer k_a by observing δ. To ensure no partial information leakage, we can further reduce every n-bit secret sequence to a k-bit sequence, for instance, use $g(k_a)$ as the secret key instead of k_a. As a result, after the reconciliation, the valid bit generation rate will be reduced by a factor of $\frac{n-k}{n}$.

In our implementation of EMG-KEY, we employ the binary Golay Code $G(23,12)$ [17] in the reconciliation stage. It is a perfect linear error-correction code, which encodes 12-bit of data into a 23-bit word and can detect any 7-bit errors or correct any 3-bit errors in each 23-bit block.

4 Experimental Methodology

Experiment Setup: In our experiment, we build a prototype of the EMG-KEY as shown in Figure 4. It includes a wristband and a device that acts as the payment device, both of which are embedded with Olimex EMG/EKG sensors [8] with a sampling frequency of 250 Hz controlled by Arduino UNO develop board [2]. Based on this prototype, we have implemented the shape-based secret key generation scheme in Python 2.7 and performed the reconciliation via Golay Code $G_{23}(23,12)$.

Table 2. Details of human subjects

No.	Age	Gender	Wrist Circ.(cm)	BMI
1	29	M	17.8	34.7
2	26	M	15.5	20.7
3	23	M	17.5	24.9
4	28	M	16.2	25.2
5	23	F	15.8	21.8
6	24	F	14.1	17.5
7	23	M	17.5	29.4
8	28	F	14.0	20.8
9	27	M	16.8	26.2
10	25	M	16.3	22.4

Testing Scenario: To conduct a comprehensive evaluation, we have recruited 10 volunteers (7 males and 3 females, details in Table 2) to conduct extensive experiments. Nine of them act as normal users while one simulates the attacker. In each experiment, the user is required to wear the wristband on his/her arm, have physical contact with the electrodes on the payment device in proximity (around 4 cm) as shown in Figure 4, and then perform a gesture to initiate a secure pairing. During this process, an attacker who wears the same type of wristband is standing nearby in such a way that he can clearly observe the gestures, and imitate them exactly. To simulate the worst case in a real application, we intentionally ask users to perform simple gestures which are easy to imitate, *e.g.*, slowly clench then release the fist. We evaluate the information leakage during the reconciliation process by letting the attacker know the exact offset data between legitimate devices during each pairing process. All the EMG signals measured from devices, and corresponding secret keys generated during these experiments are recorded for further analysis. Ten experiments are conducted on each user and there are $30 \times 10 = 300$ records in total.

Performance Metrics: Throughout the evaluation, four metrics are employed to measure the performance of our system.

- **Bit generation rate** is the number of valid secret bits we can generate per second. In our system, this metric is directly determined by the key generation scheme and reconciliation process. Let w be the coding window size in seconds and V indicate the number of predefined shape templates. With the adoption of error correction code $ECC(n,k)$, the final bit generation rate is defined as:
$$BGR = \frac{k}{wn}\log_2 V, \tag{9}$$
where $V = 3$ in our case.
- **Bit Mismatching rate** reflects the level of inconsistency between secret keys. It is defined as the number of mismatched bits divided by the length of secret key:
$$BMR = \frac{bitcount(k_a \neq k_b)}{min(|k_a|,|k_b|)}. \tag{10}$$
A low bit mismatching rate ensures legitimate devices agree on the same secret key and pair successfully with a high possibility. In our system, some factors can obviously affect the bit mismatching rate, *e.g.*, the distance between devices, the choice of error correction code, and even the complexity of gesture.
- *Entropy* is a measurement of information contained in data [19]. Given a random variable $X = [x_0, x_1, ..., x_i]$, its entropy can be computed as:
$$H(X) = -\sum_i Pr[x_i]\log_2 Pr[x_i], \tag{11}$$
where $Pr[x_i]$ is the probability of the i-th value of X. In our case, we use segment-wise entropy, in which $Pr[x_i]$ is the probability of the i-th variation shape template.
- *Mutual information* measures the mutual dependence between two variables [19], which quantifies the amount of information obtained about one random variable X through the other variable Y as:
$$I(X;Y) = \sum_{y\in Y}\sum_{x\in X} p(x,y)\log\frac{p(x,y)}{p(x)p(y)}. \tag{12}$$
The smaller the mutual information between X and Y is, the less information of Y can be gained by only observing X, or *vice versa*. In our evaluation, we use this metric to measure the information leakage between user and attacker.

5 Performance of Secret Key Generation

This section evaluates the performance of our secret key generation scheme.

We begin with an examination of the choice of the coding window size and the error correction code, both of which directly determine the bit generation rate and bit mismatching rate. According to the result, our system can generate secret bits at a rate of 5.51 bit/s, while retaining a low bit mismatching rate. After that, we show that the reconciliation process can be further extended to achieve a required key matching rate with a trade-off of bit generation rate and the generated key is random enough to pass the standard randomness test.

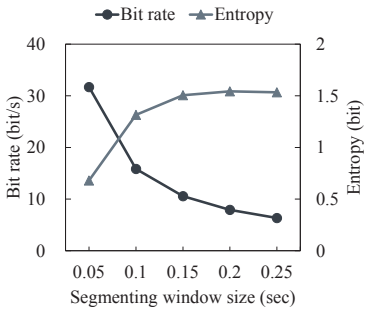


Figure 8. The bit rate before reconciliation. A small-sized coding windows results in a high bit rate, but reduces the information contained in the generated key.

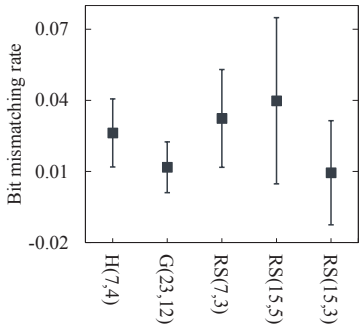


Figure 9. Performance of different error correction coding scheme. Golay Code $G(23,12)$ outperforms the other ECC codes.

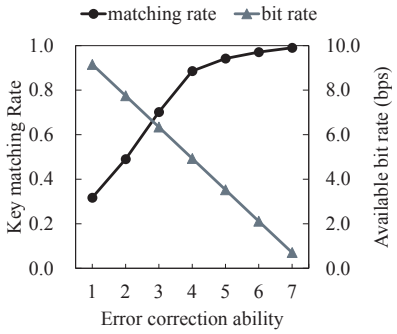


Figure 10. Trade-off between bit rate and key matching rate. Higher error correction ability leads to higher key matching rate but lower bit rate.

5.1 Effect of Parameters

5.1.1 Bit Generation Rate

An important performance indicator for a secret key generation scheme is *how fast it generates secret bits*. For our system, the bit generation rate before reconciliation directly depends on the coding window size w used to segment EMG signals in the shape-based secret key generation. Although a small coding window gives us a high bit generation rate, it also reduces the information contained in the generated secret key as the uncertainty of possible variation shapes within each window becomes smaller. As we can image, if we set the coding window size to an extreme small value, then all the variations in each coding window will be very minor and can be approximated by a horizontal-line, *i.e.*, the "stay" shape.

To find the optimal coding window size, we compute the bit generation rate and segment-wise entropy of generated secret keys with respect to different values of w. As shown in Figure 8, we observe that, with the growth of the coding window size, the bit generation rate drops quickly, but the entropy contained in each segment increases and then converges to 1.54 bits per segment (Theoretically, the maximum entropy $= -\sum_1^3 \frac{1}{3}\log_2\frac{1}{3} \approx 1.58 bit/segment$). To preserve sufficient randomness, we set the coding windows size to 0.15 seconds in our system, which leads to a bit generation rate of 10.57 bit/s and 1.51 bits information per segment.

Note that this is not the final bit rate of our system, because the reconciliation process will sacrifice part of the bit rate to alleviate the mismatching of bits via employing error correction coding. In the next section, we analyze its impact on system performance.

5.1.2 Choice of Error Correction Code

Due to the spacing between devices, differences in the electrodes' properties and the hardware imperfection, there are some discrepancies in the EMG measurements of legitimate devices, which inevitably leads to mismatching bits among the generated secret keys. To alleviate such inconsistency, error correction code is adopted at the reconciliation stage. As a result, the choice of error correction coding algorithm, as well as its setting, *i.e.*, n and k, not only define the bit mismatching rate of our system, but also cause a loss in valid bit rate.

To examine the effectiveness of different ECC codes, three candidate codes are employed: (i) *Hamming Code*, which is a linear perfect error correction code that encodes 4-bit data into 7-bit code by adding 3 parity bits. (ii) *Golay code*, a well-known linear code which translates 12-bit message into 23 bits in such a way that any 3-bit error can be corrected. (iii) *Reed-Solomon code (RS)* is a cyclic code designed to detect and correct multiple errors. By adding check symbols to the raw data, a RS code, *RS(n, k)*, can correct up to $\lfloor \frac{n-k}{2} \rfloor$ symbols of error. Such property make it suitable for burst errors and thus is widely adopted in many data storage applications [17]. Table 3 lists the ECC codes used in our evaluation, plus their parameters and properties, *i.e.*, code word length n, code length k, error-correcting ability r, information leakage and bit loss ratio.

Table 3. Candidates of error correction codes

Code	n	k	r	Leakage	Bit loss
Hamming Code	7	4	1	0.43	0.57
Golay Code	23	12	3	0.48	0.52
RS(7, 3)	7	3	2	0.57	0.43
RS(15, 5)	15	5	5	0.67	0.33
RS(15, 3)	15	3	6	0.8	0.2

Additionally, we collect a data set of raw EMG signals and corresponding secret keys from 10 users as described in Section 4. The average bit mismatching rate before reconciliation of this data set is 0.065 and the standard deviation is 0.029. We feed these data into the reconciliation process with different ECC codes and compare their performances in Figure 9.

From this figure, we find that, although Reed-Solomon Code with $n = 15, k = 3$ has the lowest average bit mismatching rate, Golay code $G(23,12)$ is a better choice as it performs more stably among different data records. Besides, we notice the standard deviation of linear ECC codes, *e.g.*, Hamming Code and Golay Code, are generally smaller than the Reed-Solomon code. This can be explained by the fact that the Reed-Solomon code may introduce more mis-

matching bits if the number of mismatching bits exceeds its correction ability due to its nonlinear nature.

According to this result, we adopt the Golay Code, $G(23,12)$, in our system and the final bit generation rate is $\frac{12}{0.15\times 23}\times \log_2 3 \approx 5.51$ bit/s.

5.2 Extensibility of Reconciliation

Although Golay code provides a good performance, its error correction ability is fixed, but in practice, different applications might pose distinct requirements on the bit generation rate and key matching rate. To further prove that our system can be extended to meet the various requirements, we employ the Reed-Solomon (RS) code in this experiment to demonstrate that the reconciliation process can be extended to achieve a required matching rate with the trade-off of bit generation rate.

In this experiment, we adopt a RS code with $n = 15, m = 3$. It encodes n symbols of m bits into k symbols to handle $r = \lfloor \frac{n-k}{2} \rfloor$ symbol errors, and brings $\frac{k}{n}$ loss to the final bit generation rate [17]. Also, we generate 500 keys for the test, each of which is 60-bit long and equivalent to 18-digit PIN code.

Figure 10 shows a trade-off between final bit generation rate and corresponding key matching rate. For instance, when error correction ability $r = 1$, the bit generation rate is 9.2 bps but the key matching rate is only 0.31. This is because the errors are larger than the error correction ability. With the growth of the error correction ability, the key matching rate is significantly improved, *e.g.*, key matching rate $= 0.9904$ when $r = 7$. However, a higher error correction ability also introduces a larger bit rate loss, which linear degrades the final bit generation rate.

5.3 Randomness of Generated Key

To ensure the randomness of generated key, we employ the standard randomness test suite from NIST [44] to examine the randomness level of secret bits after the reconciliation. This test suite conducts a series of randomness tests with a null hypothesis that the input key is random and computes the corresponding p-value. If the p-value is less than a significance level, e.g., 1% in our case, then the null hypothesis is reject and the key is claimed to be non-random.

Table 4 shows the p-values of our secret keys in the randomness tests. We can find that the p-value of each test are larger than the 1%, which implies that our system can pass the test with sufficient randomness.

Table 4. Randomness Test

Test	p-value
Frequency	0.162606
Block Freq.	0.437274
Approximate Entropy	0.637119
Runs	0.162606
Longest Run	0.025193
Cum. Sum (forward)	0.162606
Cum. Sum (backward)	0.437274
FFT	0.012650
Serial	0.275709

6 Impact of Confounding Factors

This section investigate the impact of confounding factors, namely, the distance between devices, the placement of electrodes and the gesture complexity. The evaluation results shows that, by placing the devices within 4 centimeters of one another, our system can provide a good performance with a simple gesture and is robust to the electrode placement.

6.0.1 *Secure Distance between Devices*

In our system, both legitimate devices need to be placed in close proximity on the skin to ensure a successful pairing. This is because EMG signal is a very subtle electrical activity, which can only be precisely sensed near the contracting muscles. Besides, the signal measured by devices are actually a composition of several individual EMG signals from different muscles. For example, as a complex organ, the human arm consists of 23 muscles, each of which has different functions [35]. Due to these facts, we can image that large distances between legitimate devices could increase their inconsistency in the EMG measurements, which would eventually introduce additional mismatching bits.

To evaluate *how close the devices need to be placed to ensure a successful pairing*, we conduct extensive experiments on the volunteers by placing the wristband and payment device at different distances. Figure 11 shows the corresponding bit mismatching rate between legitimate devices.

From this figure, we observe a growing trend in bit mismatching rate with the increase of distances between legitimate devices, which corresponds to our previous analysis. Also, a distance of within 4 centimeters can still maintain a good performance with the help of reconciliation. However, larger distances will exceed the correction ability of the ECC code and result in a high mismatching rate.

6.0.2 *Placement of Electrodes*

Apart from the distance between devices, another factor deriving from the subtle propagation nature of EMG and complex composition of the human arm muscle is the placement of electrodes. Although the muscles of forearm are elongated and often distributed over the whole of the forearm, we wonder whether there is difference if the electrodes is placed at different locations.

To evaluate the impact of electrode placement, we design three groups of experiments, in each of which the electrodes of the wristband and payment device are placed at different locations as shown in Figure 12. The distances among different placements are 4 centimeters while the spacing between wristband and payment device in each experiment is fixed to 2 centimeters.

We first evaluate the bit mismatching rate under each placement in Figure 13. An immediate observation from this figure is that the mismatching rate at location 2 is lower than at locations 1 and 3. This is because location 1 is relatively far away from the contracting muscles, while location 3 is often covered with more fat and tissue, which is evidenced as being able to hinder the propagation of EMG [39]. Compared with these two locations, the EMG measured at location 2 is much stronger and suffers less interference, which leads to a better performance. However, we also find that,

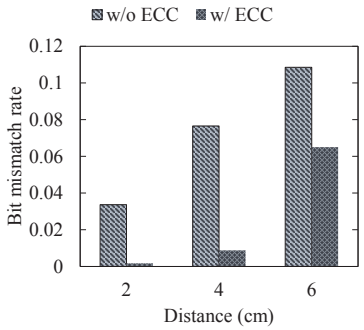


Figure 11. The bit mismatching rate with different distances between legitimate devices. A larger distance boosts the inconsistency in devices' EMG measurements and thus results in more mismatching bits in a generated secure key.

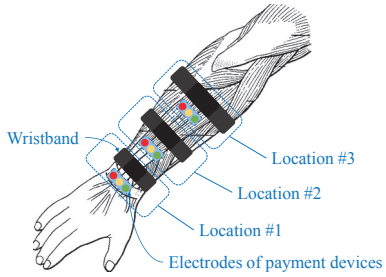


Figure 12. Illustration of electrodes placements. The distances among different placements are 4 centimeters while the spacing between wristband and payment device in each experiment is fixed at 2 centimeters.

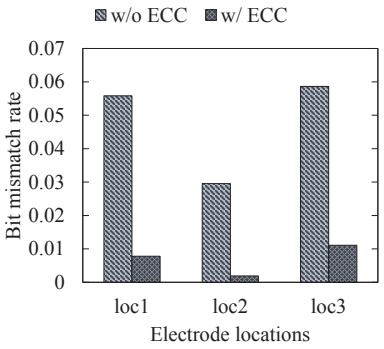


Figure 13. Bit mismatching rate of different electrode placements. Location 2 outperforms the other locations as the EMG signal is much stronger and suffers less interference in this region.

with the help of reconciliation process, the performances at locations 1 and 3 are still acceptable as most mismatching bits in secret keys can be significantly reduced by error correction code.

Also, to quantify the randomness level of secret keys generated under different electrode placements, segment-wise entropy is computed and reported in Figure 14. A higher segment-wise entropy indicates more randomness will be included in secret key and thus it is more difficult to attack. Note that, since we use three predefined shapes to approximate the EMG variation in the shape-based secret key generation, a theoretical upper-bound of the segment-wise entropy is achieved if all these shapes occur in the secret key randomly and uniformly. Thus, the maximum can is computed as: $max(H) = -\sum_1^3 \frac{1}{3} \log_2 \frac{1}{3} \approx 1.58 bit/segment$, which is represented by the dashed line above the bars. According to this figure, the entropies of secret keys generated under different electrode placements are relatively identical and all of them are approaching the theoretical maximum. This indicates that most of the information of EMG randomness is preserved no matter where the electrodes are placed.

6.0.3 *Gesture Complexity*

As our system requires users to perform a gesture to initiate the pairing process, one natural question is *whether the complexity of gestures can affect the system's performance and security level*. This question comes along with an intuitive idea that the high-complexity gestures are hard to imitate, which may introduce more robustness to attacks.

To explore the answer, we design three gestures with increasing complexity, namely, $g1$, $g2$ and $g3$. In $g1$, the user slowly clenches the fist, then releases it gently. The second gesture, $g2$, requires users to clench and release the fist quickly and repetitively. In the last gesture with the highest complexity, the users are asked to randomly moves their fingers quickly as will.

Figure 15 shows the performance of secret key generation under gestures of different complexity. We surprisingly find that the bit mismatching rate gets higher with the increase in gesture complexity. Upon further analysis, this turns out to be rooted in the fact that complex gesture, such as moving fingers randomly, is often accomplished by the collaboration of several muscles. Therefore, multiple individual EMG signals are interfering with each other during a complex gesture. Moreover, some individual EMG signals are quite minor and can easily be overwhelmed by the others. As a result, the interference between individual EMG signals leads to an obvious inconsistency in the EMG measurements between legitimate devices, which eventually results in a degradation in the performance.

Given such frustrating results, a major concern is whether a simple gesture can provide enough randomness for secure pairing. To this end, we again employ the segment-wise entropy to evaluate the randomness level provided by gestures of different complexity and present the results in Figure 16. We notice complex gestures actually does not provide information gain. Also, the average entropy of simplest gesture, *i.e.*, the slow clenching and releasing of the fist, is about 1.51 bit/segment, which approaches the theoretical upper bound of 1.58 bit/segment.

These results imply that, although a high-complexity gesture does not provide any additional enhancement to our system, the simple gesture will suffice as it can preserve enough randomness and provide a good bit mismatching rate.

7 Resistance to Attacks

In this section, we evaluate the security performance of our system. Throughout the experiments, we assume there exists a strong attacker who is able to:

- know every details of our pairing algorithm;
- stand in close proximity, precisely observe and capture all the gestures made by users during the pairing process;
- imitate these gestures exactly;
- eavesdrop on and decode all the packets sent via a public communication link, *e.g.*, WiFi, Bluetooth or NFC;

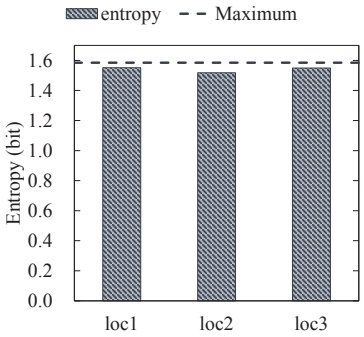


Figure 14. Entropy of the secret key generated at different electrode placements. As they are relatively stable, it suggests different placements of electrodes are acceptable.

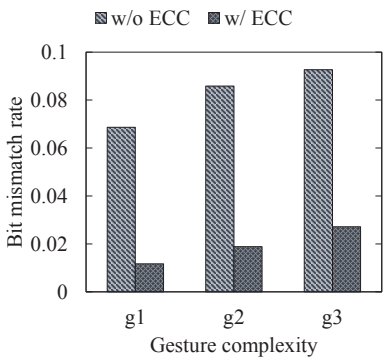


Figure 15. Effect of gestures of different complexity. Complex gesture degrades the bit mismatching rate.

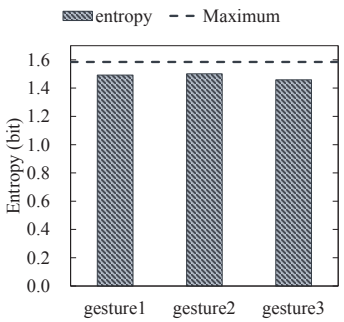


Figure 16. Effect of gestures on entropy. Complex gestures does not provide information gain and simple gesture is sufficient for our system.

In order to examine the our system's robustness to such a strong attacker, we conduct extensive experiments on 10 volunteers, in which nine of them act as normal users while one simulates an attacker who will imitate their gestures. Each user is asked to perform the pairing process 30 times in the presence of attacker and there are $10 \times 30 = 300$ pairing records in total.

We start the evaluation with the analysis on the information leakage to the attacker. The experiments demonstrate that the attacker can only obtain a negligible amount of information about the legitimate devices even when he can imitate a user's gesture exactly.

After that, we take a close look at the bit matching rate of the secret keys generated by different users and attackers, from which we can find that the bit mismatching rate of an attacker is significantly higher even with the adoption of an ECC code.

7.1 Information Leakage

To visualize the correlation between the EMG measurements of devices, we present the pairwise scatter-plots of the normalized EMG measurement of each pair of devices when both user and attacker are performing the same gesture synchronously in Figure 17.

From Figure 17(a), we can clearly observe that the EMG signal from the payment device increases linearly with respect to the measurements from user's wristbands, which implies there exists a strong correlation between them. On the other hand, even through the attacker is imitating the user's gesture synchronously, his/her EMG measurement does not appear to have a strong connection with either the user or the payment device according to Figure 17(b) and 17(c).

To further quantify the amount of information can be learned by imitating a gesture, we compute pairwise mutual information between devices in Table 5. A smaller mutual information implies less information can be learned from one variable to anther. We note that, by measuring the EMG variation in close proximity, the wristband can obtain 1.158 bits of information about the payment device's corresponding secret key. On the contrary, the attacker, albeit imitating the gesture synchronously, can only gain 0.29 bits of information about user's secret key. This indicates that the legitimate devices have 4 times more information about each other than the attacker.

Table 5. Mutual information among user's wristband A, payment device B, and attacker's devices E

	A vs. B	A vs. E	E vs. B
Mutual info.	1.158	0.290	0.274

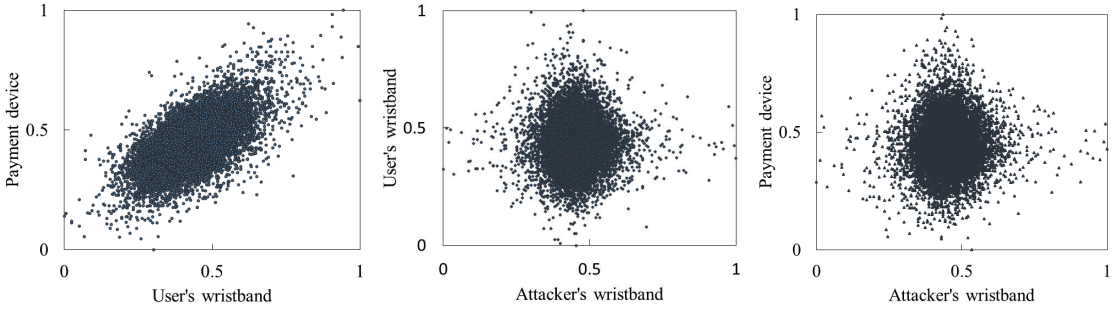
7.2 The Performance of Copy Attacker

In this section, we further assume that the attacker can get the offset information δ transmitted in the reconciliation stage between legitimate devices via eavesdropping, and try to deduce their secret key during the pairing process.

In oder to simulate such an attack, we design an experiment in which the offset information δ between legitimate devices is explicitly shared with the attacker via public communication. The same reconciliation is performed by the attacker to help the secret key estimation used by legitimate devices. The bit mismatching rate is used to quantify the possibility that the attacker can have the same secret key as legitimate devices.

The evaluation result on 10 volunteers (30 pairing experiments for each volunteer) is reported in Figure 18. We can find that the bit mismatching rate between user's wristband and payment device can be efficiently reduced by the reconciliation process (the final average bit mismatching rate is 8.924×10^{-3}). However, the attacker can not benefit from such a process: the bit mismatching rate between the key deduced by the attacker and the real secret key used actually increase after the adoption of the error correction code, which ends up with an average bit mismatching rate of 0.298. This is because if the number of mismatched bits exceeds the error correcting ability of ECC code, some matched bits might be erroneously flipped and thus more mismatching bits are introduced.

As a result, it is impossible for an attacker to hack the pairing process even if he can eavesdrop the offset information. Consider PIN codes commonly used, *e.g.*, 4-digit PIN in traditional Bluetooth and 6-digit PIN for



(a) User vs. payment device (b) Attacker vs. user (c) Close-up of shape-based encoding

Figure 17. Flow of EMG-KEY.

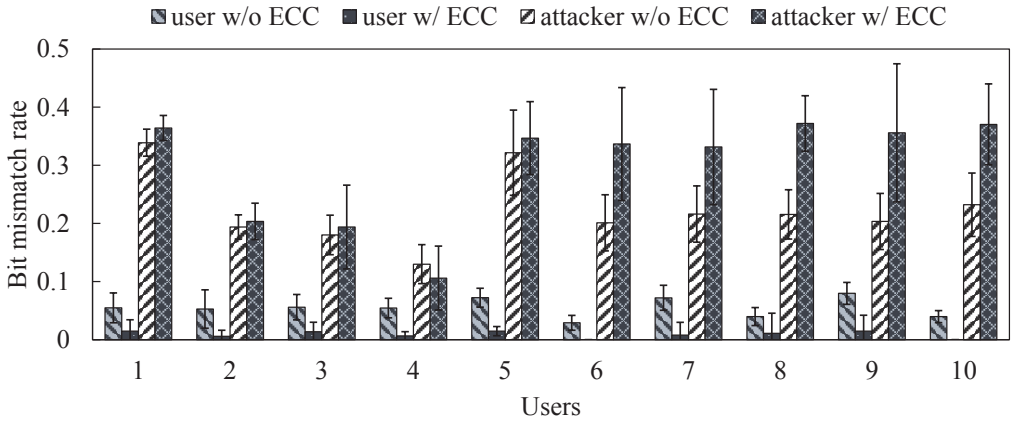


Figure 18. Bit mismatching rate of users and copy attackers.

many bankcards [37], the corresponding pairing probabilities between legitimate devices are $(1-0.008924)^{\log_2 10*4} \approx 88.84\%$ and $(1-0.008924)^{\log_2 10*6} \approx 83.64\%$, respectively. Meanwhile, the attacker only has a extreme low chance to to deduce the same secret key: $(1-0.298)^{\log_2 10*4} \approx 0.91\%$ in terms of 4-digit PIN, and $(1-0.298)^{\log_2 10*6} \approx 0.09\%$ in terms of 6-digit PIN.

8 Discussion

In this section, we discuss the practical issus of our system, and possible directions of future exploration.

EMG Wearables. As the major security of our system relies on the employment of EMG measurements, one may question ***whether the EMG sensor is available for wearable devices***. According to our study, there are already several wearable products embedded with EMG sensors, *e.g.*, Myo armband [7], Athos gear [3], and Leo smart band [6], which enable many promising applications. For instance, the Myo armband can recognize a user's gestures and provide a new way for human-computer interaction, while the Athos gear can monitor the contraction state of the muscle and be used to help physical training. We envision that, in the near future, there will be more wearable devices equipped with EMG sensors due to the fast development of Augmented Reality (AR) and the health-care market [4, 42].

Another practical concern is that ***wearing EMG electrodes is not convenient and comfortable***. However, our system does not require users to wear the EMG electrode all the time. It only needs to be contacted with skin during the pairing process, which normally takes a few seconds. Besides, many other commercial devices with EMG sensors also impose the same constraint, we believe this issue can be alleviated by better industrial design in near future.

Threat of electromagnetic emanation. Recent studies have exposed a new threat derived from electromagnetic emanation (EM). By using the electromagnetic nature of devices, it is possible for adversaries to eavesdrop the information [27] or even perform an EM signal injection attack, in which the attacker manipulates the input to the device by emitting chosen electromagnetic waveforms [48]. However, such attack techniques can not defeat our system. First, due to the fact that the EMG voltage is unobtrusive (often withing ± 10 mv), it is extremely hard to eavesdrop on its EM radiation in practice. Also, the EM signal injection attacks can be prevented via better hardware design.

Multi-Channel EMG. To make our system more reliable and practical, there are some possible directions worth exploring in the future. The first one is *the adoption of the multi-channel EMG*. To measure the muscle activity accurately, many existing wearable devices are equipped with more than one EMG sensor. We believe that the performance of our system can be further enhanced if the EMG signals from different channels can provide more information and randomness. Also, our current system only employ three basic shapes to quantify the EMG variation, therefore a more fine-grained quantization level can be adopted to improve the

system's performance.

Impact of Body States. As our system depends on the EMG signal from human body, one may concern whether the body state changes can affect the sensing of EMG.

The first one is the ***muscle fatigue***. Although it is evidenced that the EMG signal drifts to lower frequency when the muscle fatigue occurs [15], it does not significantly affect our system as a successful pairing only requires both legitimate devices have a consistent measurements on the same EMG source. However, to comprehensively examine other side effects, an in-depth medical study should be conducted. We leave it for future exploration.

Besides, as our system requires users to wear electrodes on their skin, ***sweating*** inevitably degrades the signal quality and introduces more mismatching bits. To alleviate such problem, an ECC code of stronger error correction ability can be adopted to ensure the key matching rate with a sacrifice of bit rate.

Another factor is the ***body mass***. Many studies indicate that body fat can restrain the sensing of surface EMG signal [39]. To understand the effect of this issue on our system performance, we intentionally include an overweight volunteer in the evaluation (volunteer 1 in Table 2, BMI=34.7). Throughout all the experiments, we have not found any difference between this overweight volunteer and the others. In the future, we plan to conduct a larger-scale experiment and recruit more volunteers of different BMI to further validate this issue.

9 Related Work

9.1 Secure Pairing

Many techniques have been proposed to enable secure pairing between mobile devices based on pre-shared secrets. A variety of information sources have been exploited to generate shared secret keys without prior information exchange. Such sources can be wireless channel measurements [13, 28, 33, 36, 43], human motion [10, 38, 49], vibration [9], or ambient environments [37, 46]. Azimi et al. [13] are among the first to leverage the channel reciprocity to generate secret keys from wireless signal strength. Jana et al. [28] propose an environmental-adaptive key generation scheme to boost the bit generation rate. Liu et al. [33] take one step further by using the fine-grained channel state information (CSI) as the reciprocal information to extract more information for key generation in OFDM systems. Similarly, Puzzle [43] leverages the frequency shapes of channel measurements to obtain more robust secret bits. Checksum Gestures [10] uses a single-continuous gesture to generate an authentication code to replace the traditional PIN input for wearables. Mayrhofer [38] establishes a secure link between two devices by shaking them together, and leverages their trajectories as shared information. Gait is also exploited by Xu *et al.* to pair on-body devices [49]. Instead of using hand-incurred motion, Ving [9] leverages the vibration of a desk as the shared secret for all devices on the desk. Ambient environment based approaches authenticate the proximity of two devices based on ambient wireless signals [37] or ambient audios [46].

Another research direction is the secure near-field communication. Dhwani leverages the self-jamming and self-interference cancellation to provide an secure acoustic communication in near field [41], EnGarde is a compact hardware design to jam malicious interactions for NFC [26], and nShield attenuates the signal strength against passive eavesdropping to provide better security [51].

Different from these approaches, EMG-KEY exploits a new dimension, *i.e.*, EMG, to provide secure pairing. Due to its subtle and random characteristics, EMG variation is random in nature and can only be sensed in close proximity with physical contact, which makes our system robust to proximate eavesdroppers and even camera-based shoulder-surfers.

9.2 EMG Analysis

Traditionally, EMG is used by clinic doctors and biomedical scientist to study muscle fatigue [15, 24], neuromuscular diseases [23, 47] and human kinesiology [40]. In recent years, the EMG is also widely adopted to enable different promising applications, e.g., controlling prosthetic [12], emotion recognition [16,21], and speech recognition [29,34]. Apart from this, extensive effort has been devoted to the exploration of using EMG as an interface of Human-machine interaction [11, 31]. Alternatively, our system leverages the EMG signal to secure the pairing of devices.

9.3 Other Biometric Applications

Apart from EMG, fingerprint, face, gait and voice are wildly used on modern devices to strengthen security [32]. Other biometrics, such as bio-impedance [18], electrocardiograph [20], body electric potential change [25], bio-vibration response [50], and body capacitance profile [45] are also exploited to provide similar secure functions. In complement to these works, our system is the first one to utilize EMG signal to enable secure pairing with a comparative performance and high security level.

10 Conclusion

In this work, we propose a secure pairing system for wearable devices by exploring the randomness embedded in an EMG signal. We design a shape-based secret key generation scheme and leverage the error correction code to alleviate the inconsistency between devices. Extensive experiments on ten volunteers indicate that our system is robust to many confounding factors and can achieve a competitive bit generation rate of 5.51 bit/s while maintaining a highly successful pairing rate of 88.84%. Also, evaluation results in the presence of copy attackers demonstrate our system can defend against strong attacks.

11 Acknowledgments

We thank our shepherd and anonymous reviewers for their help and invaluable comments. The research was supported in part by grants from 973 project 2013CB329006, RGC under the contracts CERG 622613, 16212714, 16203215 and M-HKUST609/13.

12 References

[1] Apple pay. http://www.apple.com/apple-pay.
[2] Arduino uno. https://www.arduino.cc/en/Main/ArduinoBoardUno.
[3] Athos gear. https://www.liveathos.com.
[4] Devices with emg sensor. http://vandrico.com/wearables/device-categories/components/emg-sensor.

[5] Fitbit. https://www.fitbit.com.

[6] Leo smartband. http://leohelps.com.

[7] Myo armband. https://www.myo.com.

[8] Olimex emg shield. https://www.arduino.cc/en/Main/ArduinoBoardUno.

[9] J. Adkins, G. Flaspohler, and P. Dutta. Ving: Bootstrapping the desktop area network with a vibratory ping. In *Proceedings of the 2nd International Workshop on Hot Topics in Wireless*, pages 21–25. ACM, 2015.

[10] I. Ahmed and *et al.* Checksum gestures: continuous gestures as an out-of-band channel for secure pairing. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 391–401. ACM, 2015.

[11] C. Amma and *et al.* Advancing muscle-computer interfaces with high-density electromyography. CHI '15, pages 929–938. ACM, 2015.

[12] A. H. Arieta and *et al.* Study on the effects of electrical stimulation on the pattern recognition for an emg prosthetic application. In *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the*, pages 6919–6922. IEEE, 2006.

[13] B. Azimi-Sadjadi and *et al.* Robust key generation from signal envelopes in wireless networks. In *Proceedings of the 14th ACM conference on Computer and communications security*, pages 401–410. ACM, 2007.

[14] R. Beck. Muscle fiber conduction velocity. *Wiley Encyclopedia of Biomedical Engineering*, 2006.

[15] B. Bigland-Ritchie, E. Donovan, and C. Roussos. Conduction velocity and emg power spectrum changes in fatigue of sustained maximal efforts. *Journal of applied physiology*, 51(5):1300–1305, 1981.

[16] B. Cheng and G.-Y. Liu. Emotion recognition from surface emg signal using wavelet transform and neural network. In *Proceedings of the 2nd international conference on bioinformatics and biomedical engineering*, pages 1363–1366, 2008.

[17] G. C. Clark Jr and J. B. Cain. *Error-correction coding for digital communications*. Springer Science & Business Media, 2013.

[18] C. Cornelius and *et al.* A wearable system that knows who wears it. MobiSys '14, pages 55–67. ACM, 2014.

[19] T. M. Cover and J. A. Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[20] H. P. da Silva and *et al.* Finger ecg signal for user authentication: Usability and performance. In *Biometrics: Theory, Applications and Systems (BTAS), 2013 IEEE Sixth International Conference on*, pages 1–8, 2013.

[21] M. De Wied and *et al.* Facial emg and heart rate responses to emotion-inducing film clips in boys with disruptive behavior disorders. *Psychophysiology*, 46(5):996–1004, 2009.

[22] S. R. Devasahayam. *Signals and systems in biomedical engineering: signal processing and physiological systems modeling*. Springer Science & Business Media, 2012.

[23] R. Ferri and *et al.* A quantitative statistical analysis of the submentalis muscle emg amplitude during sleep in normal controls and patients with rem sleep behavior disorder. *Journal of sleep research*, 17(1):89–100, 2008.

[24] P. A. Gribble and J. Hertel. Effect of lower-extremity muscle fatigue on postural control. *Archives of physical medicine and rehabilitation*, 85(4):589–592, 2004.

[25] Grosse-Puppendahl and *et al.* Platypus: Indoor localization and identification through sensing of electric potential changes in human bodies. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '16, pages 17–30. ACM, 2016.

[26] J. J. Gummeson and *et al.* Engarde: Protecting the mobile phone from malicious nfc interactions. MobiSys '13, pages 445–458. ACM, 2013.

[27] Y. Hayashi and *et al.* A threat for tablet pcs in public space: Remote visualization of screen images using em emanation. CCS '14, pages 954–965. ACM, 2014.

[28] S. Jana and *et al.* On the effectiveness of secret key extraction from wireless signal strength in real environments. In *Proceedings of the 15th annual international conference on Mobile computing and networking*, pages 321–332. ACM, 2009.

[29] S.-C. S. Jou and *et al.* Towards continuous speech recognition using surface electromyography. In *INTERSPEECH*, 2006.

[30] E. J. Keogh and M. J. Pazzani. Scaling up dynamic time warping for datamining applications. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 285–289. ACM, 2000.

[31] D. Kim and *et al.* Digits: freehand 3d interactions anywhere using a wrist-worn gloveless sensor. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, pages 167–176. ACM, 2012.

[32] D. J. Kim, K. W. Chung, and K. S. Hong. Person authentication using face, teeth and voice modalities for mobile device security. *IEEE Transactions on Consumer Electronics*, 56(4):2678–2685, 2010.

[33] H. Liu and *et al.* Fast and practical secret key extraction by exploiting channel response. In *INFOCOM, 2013 Proceedings IEEE*, pages 3048–3056. IEEE, 2013.

[34] L. Maier-Hein and *et al.* Session independent non-audible speech recognition using surface electromyography. In *Automatic Speech Recognition and Understanding, 2005 IEEE Workshop on*, pages 331–336. IEEE, 2005.

[35] E. N. Marieb and K. Hoehn. *Human anatomy & physiology*. Pearson Education, 2007.

[36] S. Mathur and *et al.* Radio-telepathy: extracting a secret key from an unauthenticated wireless channel. In *Proceedings of the 14th ACM international conference on Mobile computing and networking*, pages 128–139. ACM, 2008.

[37] S. Mathur and *et al.* Proximate: proximity-based secure pairing using ambient wireless signals. In *Proceedings of the 9th international conference on Mobile systems, applications, and services*, pages 211–224. ACM, 2011.

[38] R. Mayrhofer and H. Gellersen. Shake well before use: Intuitive and secure pairing of mobile devices. *Mobile Computing, IEEE Transactions on*, 8(6):792–806, 2009.

[39] R. Merletti and P. A. Parker. *Electromyography: physiology, engineering, and non-invasive applications*, volume 11. John Wiley & Sons, 2004.

[40] F. Mokaya and *et al.* Myovibe: Vibration based wearable muscle activation detection in high mobility exercises. UbiComp '15, pages 27–38. ACM, 2015.

[41] R. Nandakumar and *et al.* Dhwani: secure peer-to-peer acoustic nfc. In *ACM SIGCOMM Computer Communication Review*, volume 43, pages 63–74. ACM, 2013.

[42] t. Peter Harrop. Wearable technology 2015-2025: Technologies, markets, forecasts. pages 285–289. IDTechEx, 2015 Feb.

[43] Y. Qiao, K. Srinivasan, and A. Arora. Shape matters, not the size: A new approach to extract secrets from channel. In *Proceedings of the 1st ACM workshop on Hot topics in wireless*, pages 37–42. ACM, 2014.

[44] A. Rukhin and *et al.* A statistical test suite for the validation of random number generators and pseudo random number generators for cryptographic applications. *NIST special publication*, pages 800–22.

[45] M. Sato and *et al.* Touché: Enhancing touch interaction on humans, screens, liquids, and everyday objects. CHI '12, pages 483–492. ACM, 2012.

[46] D. Schurmann and S. Sigg. Secure communication based on ambient audio. *Mobile Computing, IEEE Transactions on*, 12(2):358–370, 2013.

[47] A. Subasi. Classification of emg signals using pso optimized svm for diagnosis of neuromuscular disorders. *Computers in biology and medicine*, 43(5):576–586, 2013.

[48] M. Vuagnoux and S. Pasini. Compromising electromagnetic emanations of wired and wireless keyboards. SSYM'09, pages 1–16. USENIX Association, 2009.

[49] W. Xu and *et al.* Walkie-talkie: Motion-assisted automatic key generation for secure on-body device communication. In *IPSN*, pages 1–12. IEEE, 2016.

[50] L. Yang, W. Wang, and Q. Zhang. Vibid: User identification through bio-vibrometry. In *IPSN*, pages 1–12. IEEE, 2016.

[51] R. Zhou and G. Xing. nshield: A noninvasive nfc security system for mobiledevices. MobiSys '14, pages 95–108. ACM, 2014.