ToAuth: Towards Automatic Near Field Authentication for Smartphones

Weixi Gu[1], Zheng Yang[1], Longfei Shangguan[2], Xiaoyu Ji[2], Yiyang Zhao[1]
1 School of Software and TNList, Tsinghua University, Beijing, China
2 Department of Computer Science and Engineering, Hong Kong University of Science and Technology
guweixigavin@gmail.com, {yang, longfei, yunhao}@greenorbs.com, xji@ust.hk, yiyangzhao@tsinghua.edu.cn

Abstract—**Near field authentication is of great importance for a range of applications, and has attracted many research efforts in the past decades. Several approaches have been developed and demonstrated their feasibility. The state-of-art works, however, still have much room to improve their automation and usability. First, user assistance is required in most existing approaches, which will be easily observed and imitated by attackers. Second, the authentications of several works heavily depend on special hardware, e.g., server or high resolution screen, which greatly restricts their application scenarios. In this paper, we present a near field authentication system ToAuth that needs little human assistance and is compatible with most smartphones. ToAuth is based on the key insight that the acceleration traces are similar for a pair of smartphones when they are contacting physically and vibrating. The random vibration patterns are sufficiently uncertain to provide high entropy to generate a pair of cryptographic keys yet are inimitable for a third party who does not get in touch with the vibration source. ToAuth leverages the keys to make authentication for smartphones. We implement ToAuth on Android platform and evaluate its performance under various scenarios. Extensive experiments demonstrate ToAuth could achieve around 90% success rate in stable environment, and prevent attacks depended on vibration noise.**

I. Introduction

The development of wireless and embedded technology has fostered the flourish of smartphone market. Users are gradually used to relying on wireless or NFC (Near Field Communication) to perform various interactions such as file transfer, message synchronization, ticket exchange and bill payment, etc [1–3]. Before communication, a pair of smartphones leverage predesigned key exchange protocol to make a one-time session key. The broadcast nature of wireless link, however, gives rise to its security concerns. Because of the inherent susceptibility of wireless communication, adversaries can easily achieve the session key by man-in-the-middle (MITM) [4] and masquerade one party to communicate with the other. Thus the communication is interrupted and even destroyed. Since not all smartphones have NFC chips, short-range communication is much likely to suffer these kinds of attacks without authentication.

Authentication is necessary for the short-range communication between two smartphones. Previous works either rely on the similarity or simultaneity of human gestures' patterns [5, 6], or depend on special hardware such as camera, high resolution screen [7] or even servers [8] to make authentication between smartphones. Nevertheless, the wide applications of these existing solutions are restricted by the following drawbacks.

Firstly, the ones based on gestures require the involvements of users. For example, users should shake phones [5] or move fingers [6] to generate cryptographic keys and make authentication. The gestures of users, however, may leak information of secret keys and adversaries can easily attack by observing and imitating. Besides, assistance of users may influence their experience, as users have to perform specific actions to achieve authentication. Secondly, for the latter, special hardware is required in authentication system. To name a few, the image-based comparison in [7] requires high resolution screen, and the protocols in [5, 8] ask for servers as trust parties. Relying on special hardware severely limits the application scenarios of these works.

In this paper, we propose ToAuth to generate cryptographic keys and achieve near field authentication for smartphones. The operation of ToAuth needs little user assistance and has no requirement for special hardware. The key insight of ToAuth is to leverage the vibrator in smartphones to generate random vibration. During the random vibration, the initiator turns vibration engine on to vibrate several times. The period of each time is randomly selected from a time span predesigned by ToAuth. When two smartphones are touched together, one of them (e.g., Alice) performs random vibration while the other (e.g., Bob) senses forced vibration and records acceleration values by his accelerometer. The acceleration patterns of Alice and Bob are therefore similar, which can be exploited to generate a pair of cryptographic keys, and further achieve authentication. Compared with the existing approaches, random vibration in ToAuth is automatic and invisible. Thus it is difficult to be imitated. Besides, as vibrators and accelerometers are general on smartphones, ToAuth can be widely employed. The extremely large key space (at least 2^{21}) promises it an efficient and secure authentication system.

To summarize, the key contributions of this paper are as follows:

1) We design a novel and practical mechanism grounded on random vibration to generate cryptographic keys, and further make authentication automatically. This kind of mechanism is simple yet effective.
2) We propose a probabilistic model and a feature reconciliation technology to generate symmetric keys effectively.
3) We implement ToAuth system and evaluate its performance through 36 groups of experiments under various scenarios. The experiment results show that our mechanism achieves around 90% success rate in stable environment. Besides, by experiments, we also

evaluate that ToAuth is able to prevent attacks relying on vibration noise.

The rest of this paper is organized as follows: in Section II, we briefly review the related work. In Section III, we present the system's background, and then introduce the overview of ToAuth's architecture in Section IV. We describe system design and analysis in Section V. The experimental evaluation is shown in Section VI. We finally conclude this paper in Section VII.

II. Related Work

In this section, we broadly review the state-of-the-art research areas related to our work. They can be divided into the following two categories:

Human intervention A number of prior works require users perform gestures to achieve near field authentication. In [5, 9, 10], users are asked to shake two phones together, so that they get similar acceleration patterns. In [6], the authors utilize the simultaneity of finger movements to make authentication. However, a powerful attacker who can emulate the similar gesture is able to carry out MITM [4] if the movements are observable. ToAuth advances these research works with little human intervention. It relies on random vibration generated by smartphones, which only requires users to keep them together. Little human involvement would reduce the possibility of MITM.

Hardware support Some protocols are based on advanced hardware. For example, the proposed scheme SiB in [11] requires a smartphone encode the data into a two-dimensional barcode, and the other reads it by the built-in camera. Given that not all smartphones are equipped with cameras, this approach is not widely used. Snowflake [12] makes authentication by image comparisons. However, this mechanism requires high resolution display that only certain smartphones can support. The approach which leverages infrared to make authentication is proposed in [13]. It is only available for smartphones equipped with infrared transceivers. Bump [8] is a popular exchange protocol for smartphones, which is widely used in many applications. It makes authentication by a server to check the time, location, and force that two smartphones are bumped together. Moreover, the protocols based on RFID have also been widely used in authentication [14–16], Compared with the works above, ToAuth just requires a vibrator and an accelerometer, which are common devices in smartphones.

III. Background

A. Preliminary

Fig.1 illustrates a case of random vibration in our experiments. As expected, the acceleration traces of Alice and Bob are similar. The vibration regions of them are labelled as L_i and $L_i^{'}$ respectively. During each vibration, acceleration amplitudes rise firstly, and keep large for a while, then decrease finally until vibration disappears. Fig. 3 shows a vibration region in detail. The vibration region is composed of t_1, t_2 and t_3. During the beginning phase t_1, when Alice just turns on her vibration engine, acceleration amplitudes of Alice and Bob rise gradually with the increase of vibration force. During the ending phase t_3, even though the vibration engine stops, the vibrator at Alice's side is still vibrating by inertia. Acceleration amplitudes of both sides fall down with the decrease of vibration force. During the period t_2, vibration force at Alice's side keeps maximum and stable. Meanwhile, the forced vibration at Bob's side also reaches its peak. Acceleration amplitudes at both sides, therefore, are large in general. Based on our experiments, t_1 usually lasts 75ms and t_3 usually lasts 70ms.

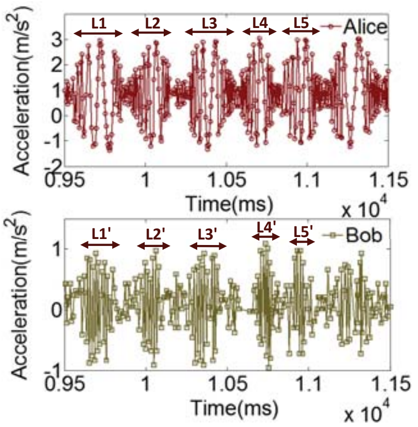


Fig. 1: Acceleration trace of random vibration

The acceleration amplitude trends of random vibrations are similar with traces shown in Fig.1 in our 40 groups of experiments conducted by 20 pairs of volunteers. In the experiments, we utilize vibrator's default frequency around 12000Hz to generate random vibration, and set the sample rate of accelerometer as 90Hz to record the acceleration value. Each pair keeps two smartphones touched and lets them vibrate randomly at least 20 seconds.

In ToAuth, we define t_2 in Fig.3 as *saturated vibration region* to describe the region which begins at 75ms after the vibration engine's starting and ends when the vibration engine stops. The first 75ms period t_1 and the last 70ms period t_3 are called the first *non-saturated vibration region* and the last *non-saturated vibration region* respectively. To distinguish the vibration region, we call the executing time of vibration engine *running time*, which is composed of t_1 and t_2.

B. Challenge

During non-saturated vibration regions, vibration force is not significant enough after transmission, which causes some certain vibration regions could not be detected by Bob but Alice. Thus the offsets labeled in Fig.3 exist. Therefore, we cannot take the whole vibration region as mutual vibration region directly. During saturated vibration regions, vibration force keeps maximum and stable, so that the accelerometer of Bob is able to sense it. To better recognize the regions with similar periods at both sides, we take saturated vibration regions as mutual vibration regions. Since the random vibration process is generated by Alice, Bob does not know the real time of saturated vibration regions. Bob needs to take a swift yet effective method based on acceleration samples to recognize these regions at his side.

The generation of key requires saturated vibration regions and non-saturated vibration regions at both sides to match completely, and any offset could cause authentication failure. Even the accelerometer of Bob could sense significant vibration force and record obvious accelerations during the saturated vibration regions, the lower sample rate of accelerometer, however, still cannot provide a fine-grained acceleration trace of vibration. Thus it is likely to cause few deviations in the

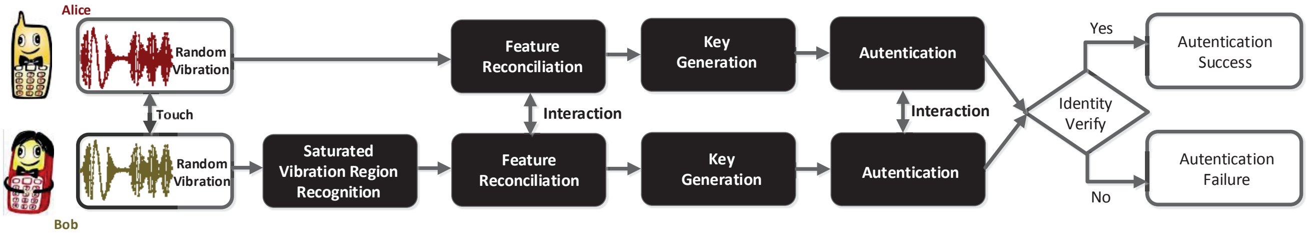


Fig. 2: System Architecture

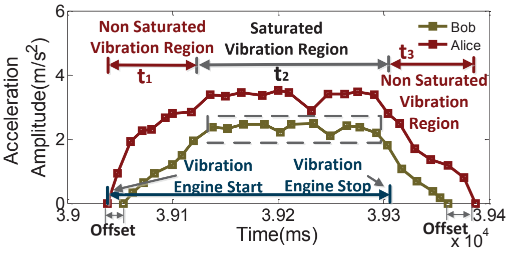


Fig. 3: Vibration region

saturated vibration regions measured by acceleration amplitude samples at Bob's side to their real regions at Alice's side. As shown in Fig.3, the region in the black dashed box is the saturated vibration region measured by Bob, and t_2 is its real time. Even though the regions are similar in most periods, they are still not aligned completely. Accordingly, an approach should be implemented to dismiss such issues with little privacy disclosure.

Encountered the two challenges above, ToAuth leverages a probabilistic model to identify saturated vibration regions, and reconciles the deviations at Bob's side by a feature reconciliation method.

C. Attack Model and Design Goal

An adversary, enabling to tamper, disturb, block and delay messages, exists in public communication channels and will eavesdrop or impersonate a legal part in a conversation between two legal communicators by MITM attack. The adversary aims to obtain the session key from the near filed authentication system stealthily and then acknowledge the conversation content from other parts.

ToAuth aims to prevent the MITM attack automatically relying on the local network. The authentication protocol requires no prior knowledge and should be carried out conveniently.

IV. System Overview

In this section, we introduce the basic components of ToAuth and its work flow. Fig .2 portrays ToAuth's architecture. The authentication system is mainly composed of five parts:

a) Random Vibration: Alice and Bob touch together. Alice boots up the vibration engine to produce random vibration in a period, which initiates Bob's forced vibration. Bob records the acceleration trace by his accelerometers and then calculates the acceleration amplitudes.

b) Saturated Vibration Regions Recognition: Since random vibration process is generated by Alice, Bob has no knowledge about the time of each saturated vibration region during the random vibration process. In order to get mutual vibration regions, he should take methods to acquire the information. Given an acceleration amplitude trace, Bob relies on a probabilistic model, which depends on the Gaussian distribution, to recognize saturated vibration regions at his side.

c) Feature Reconciliation: Feature reconciliation is to reconcile deviations in saturated vibration regions of Bob based on the ground truth of Alice. Relying on feature reconciliation, saturated vibration regions of Alice and Bob could be completely aligned over time.

d) Key Generation: ToAuth converts time slots to bit streams according to mutual vibration regions. More specifically, if time slots fall in mutual vibration regions, ToAuth defines them as 1, otherwise, they are 0. And then, ToAuth generates cryptographic keys by hash value of these bit streams.

e) Authentication: Authentication depends on key confirmation. By encrypting and decrypting predesigned messages, Alice and Bob could verify the validity of their cryptographic keys. If their keys are identical, it means that key agreement is achieved and therefore authentication succeeds. Otherwise, authentication fails.

V. System Design and Analysis

In this section, we discuss the design and implementation of ToAuth.

A. Random Vibration

Alice generates a random vibration sequence $< R_1, I_1, R_2, I_2 ..., R_i, I_i ..., R_n, I_n >$. R_i stands for the ith running time of vibration engine. I_i points to the interval between R_i and R_{i+1}. Based on our observation, R_i should be longer than 75ms to reach the saturated vibration region. Besides, given the fact that a vibration region would last 70ms after the vibration engine stops, I_i should be longer than 70ms to distinguish two consecutive vibration regions. Therefore, ToAuth randomly selects $R_i \in [75ms, \ 500ms]$ and $I_i \in [70ms, \ 500ms]$. The total time of the random vibration sequence is T, which is set to be 10s in ToAuth. So the minimal of vibration number is 10, and the maximal is 68.

Acceleration amplitudes generated by random vibration distribute in x-,y- and z- directions. ToAuth selects one direction with the maximum average acceleration amplitude value to analyze. Firstly, this reduces the negative influence of environment noise nearby so that it makes pattern recognition easier. Secondly, since acceleration amplitudes represent the energy in that direction, larger amplitudes usually mean that more energy concentrates on that orientation. So this process guarantees only small fraction of information would be filtered out. As shown in Fig.4, given an acceleration sequence $V = \langle V_1, V_2, ..., V_i, ..., V_n \rangle$, where V_i stands for the

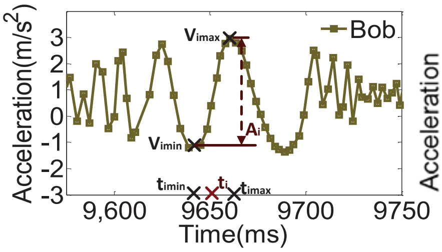


Fig. 4: Acceleration trace

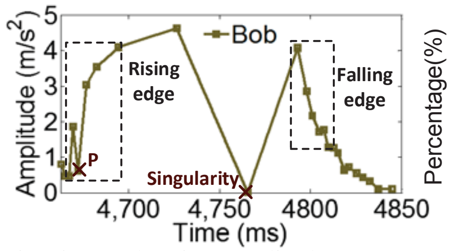


Fig. 5: Acceleration amplitude trace

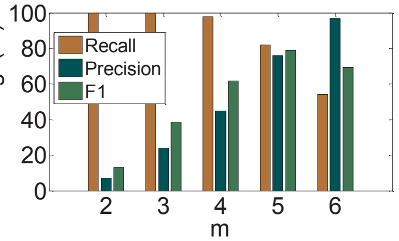


Fig. 6: Performance of various m

ith acceleration value. We calculate acceleration amplitudes $A_i = |V_{imax} - V_{imin}|$ and its time $t_i = \frac{|t_{imax}+t_{imin}|}{2}.A_i$ is the ith acceleration amplitude and t_i is its relevant time. V_{imax} and V_{imin} stand for the maximum and minimal values in ith monotonic sequence of acceleration samples, and their occurrence time slots are at t_{imax} and t_{imin}. The axis with the largest average amplitude in x-, y-, and z- is selected as the analysis source.

B. Saturated vibration regions recognition

Since Bob has no knowledge about the time of saturated vibration regions, he relies on acceleration amplitude samples to recognize them. Saturated vibration regions are where vibration force keeps maximum and stable. Acceleration amplitude samples in saturated vibration region, however, do not always keep high. Fig. 5 shows an instance of acceleration amplitude samples during the vibration region. An acceleration amplitude singularity exists at 4760ms, where vibration force keeps maximum and stable. Its value is close to zero, which is much lower than that of its neighbors and approaches to the amplitudes in stillness. This is because the sample rate of accelerometer is much lower than the vibration frequency, and therefore insufficient samples could be used to provide a fine-grained trend of acceleration amplitudes. Besides, acceleration amplitudes of forced vibration in different smartphones are various according to our experimental observation. Thus we cannot rely a static threshold to find high acceleration amplitude samples and then recognize saturated vibration regions. To solve this problem, we firstly localize a vibration region, and then leverage a probabilistic model to recognize the saturated vibration region in it.

1) Vibration region localization: In Fig.5, we can observe that acceleration amplitudes rise gradually when vibration begins, and fall down gradually to zero. Intuitively, it is a common phenomenon existing in non-saturated vibration regions. We call the trend of acceleration amplitude samples in the first and the last non-saturated vibration region as rising edge and falling edge respectively. For each vibration, it starts from the rising edge and ends at the falling edge, and the saturated region lies between the two edges. Accordingly, ToAuth could determine a vibration region approximately by the occurrence of the rising edge and the falling edge.

However, not all acceleration amplitude samples rise or fall in vibration edges. In Fig.5, P is a singularity at the rising edge. Its amplitude is lower than the prior sample. Under this circumstance, we leverage Rising Rate (RR) and Falling Rate (FR), which are based on Longest Subsequence[17], to detect rising edges and falling edges. Their definitions are listed as follows:

Definition 1: Given an acceleration amplitude sequence $A = \langle A_1, A_2, ..., A_i, ..., A_n \rangle$, $\exists A_{sub} = \langle A_{s1}, A_{s2}, ..., A_{si}, ..., A_{sm} \rangle$, where $s1 < s2 < ... < si < ... < sm$, $si \in [1, n]$. If A_{sub} is the longest increasing subsequence, $RR = \frac{m}{n}$. If A_{sub} is the longest decreasing subsequence, $FR = \frac{m}{n}$.

If $RR > \xi, \xi \in (0, 1)$, A is defined as a rising edge. Similarly, if $FR > \xi$, A is defined as a falling edge. Based on RR and FR, ToAuth could recognize the general trend of edges with few singularities.

ToAuth utilizes a sliding window W to measure RR and FR of vibration edges. The size of window is the length n of the acceleration amplitude sequence A in *Definition 1*. To describe the trend of edges, W is set as 6, which is the minimum number of acceleration amplitude samples in non-saturated vibration regions based on our labelled experimental data. To determine m, the length of the longest increasing or decreasing subsequence A_{sub}, we test precision, recall and F1 [18] of vibration edge detection under various m. When a rising edge is detected in a first non-saturated vibration or a falling edge is detected in a last non-saturated vibration region, it is a positive case, otherwise, it is a negative case. Besides, since the time span of window W is smaller than that of non-saturated region, there are possible more than one window W that meets $RR > \xi$ or $FR > \xi$ in one vibration edge. In such case, we only count a positive case in the vibration edge. Fig.6 portrays the three indices grounded on our experiments. When m is less than 5, recall keeps above 99%, and precision increases with the improvement of m. When m reaches 6, precision arrives at 98%, but recall decreases. Accordingly, in our experiments, ToAuth sets m to be 5, where $F1$ is highest. At this point, recall is 82%, and precision equals 76%. In a sliding window, if $RR > \xi$ or $FR > \xi$, where $\xi = \frac{5}{6}$, it means that a rising edge or a falling edge occurs.

Bob localizes a vibration region from the time of the first sample at the first rising edge to the time of the last sample at the first falling edge which follows the first rising edge. Fig .7 illustrates an instance of vibration region detection process at Bob's side. The serial numbers of acceleration amplitude samples stand for their occurrence order in this vibration. The black square boxes present the longest subsequence in W. When the first sliding window $W_{i,1}$ covers the period from the 1st sample to the 6th sample, the size of longest increasing subsequence is 6, and thus $RR = 1$, which demonstrates the occurrence of a rising edge. When the sliding window arrives at the period from the 13th sample to the 18th sample, the number of longest decreasing subsequence is 4, and thus FR is less than ξ. Accordingly, no edge occurs in $W_{i,13}$. The window $W_{i,24}$, when W slides to the period from the 24th sample to 29th sample, contains a longest decreasing subsequence whose size is 5. FR equals ξ, which indicates a falling edge. In $W_{i,2}$, even though it contains a rising edge, it is not the first rising edge. Therefore, the vibration region lies in the period from the time of the 1st sample to the time of the 29th sample

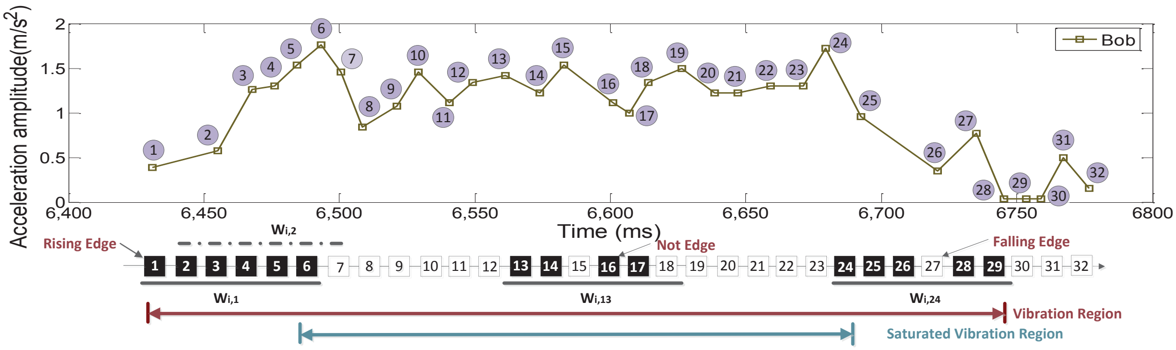


Fig. 7: Vibration region and saturated vibration detection

approximately.

2) Probabilistic model establishment: Since vibration regions are measured approximately, saturated vibration regions cannot be determined by their temporal relations directly. Because of the singularities, we establish a probabilistic model to localize the saturated vibration region. The probabilistic model is grounded on the fact that acceleration amplitude samples conform to Gaussian distribution during the saturated vibration region. We demonstrate this property by physical and mathematical analysis. From physical perspective, centrifugal force of the vibrator is stable during this phase, thus each jigging motion is independent. As acceleration amplitude samples reflect the centrifugal force, they could be assumed to conform to Gaussian distribution. Also, we have demonstrated this fact by *Kurtosis and Skewness test* [19] based on our experimental data from mathematical view.

To determine the parameters of probability density function in Gaussian distribution. ToAuth selects $S = \{S_{A1}, S_{A2}, ...S_{Ai}, ...S_{An}\}$. The element S_{Ai} is the maximum acceleration amplitude sample in the ith vibration during a random vibration process. For instance, in Fig.7, the 6th sample is largest among the acceleration amplitude samples in the vibration, so it is selected in S. Since the highest acceleration amplitude sample only occurs in a saturated vibration region due to its greatest force, S is an assemble composed of acceleration amplitude samples in saturated vibration regions. ToAuth calculates the average value $\overline{S}$ of assemble S, and its standard deviation σ [20]. Based on the property of Gaussian distribution, ToAuth sets $\hbar = \overline{S} - k * \sigma(k = 1, 2, 3...)$ as the lower bound of acceleration amplitudes in saturated vibration regions, and leverages it to distinguish saturated vibration regions from other regions. Fig.8 illustrates the detection performance of acceleration amplitude samples in saturated vibration regions based on our experiments. When k is less than 2, precision achieves a large value around 100%, but recall is less than 60%. It is much possible to overlook the samples whose values are relatively low in saturated vibration regions. In contrast, when k is 3, recall increases but precision falls down. It is because ToAuth mistakes higher acceleration amplitude samples in non-saturated vibration regions as those in saturated vibration regions. Suggested by Fig.8, we configure k to be 2 because of the highest $F1$.

In a vibration region, ToAuth checks acceleration amplitude samples from the beginning to the end. When it detects the first acceleration amplitude sample whose value is higher than $\hbar$, ToAuth marks it as the beginning of a saturated vibration region. Similarly, when ToAuth detects the last acceleration amplitude sample whose value is higher than $\hbar$, ToAuth labels it as the end of the saturated vibration region. In Fig.7, the 6th acceleration amplitude sample is the first one higher than $\hbar$, and the 24th sample is the last one higher than $\hbar$. Accordingly, the saturated vibration region lies in the period from the time of the 6th sample to the time of the 24th sample.

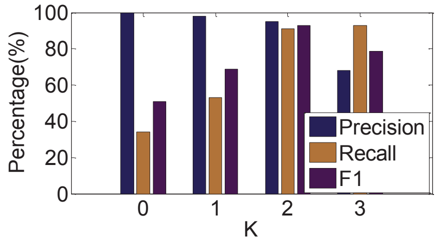


Fig. 8: Detection performance of different k

C. Feature reconciliation

Fig.9 illustrates a saturated vibration region recognized by Bob compared with its ground truth. Even though the saturated vibration region possesses a similar time span compared with the ground truth, deviations β_{i1} and β_{i2} exist at two terminals of the saturated vibration region. It is due to two reasons. One is the sample rate of accelerometer is slower than the vibration frequency, so it cannot record all acceleration amplitudes in saturated vibration regions. The other is that even the probabilistic model is able to recognize most acceleration amplitude samples in saturated vibration regions, it is still possible to mistake a few samples with low probability. Intuitively, the deviation prevents us from generating symmetric keys. We should try to dismiss the deviation at terminals. In this section we rely on feature reconciliation[21] to solve this problem.

We define that δ is the parameter to measure the maximum deviation at the terminals of Bob's saturated vibration regions in a random vibration process. Based on the *running time*, Alice records the starting and the ending times of saturated vibration regions: T=$\{\langle tb_1, te_1\rangle, \langle tb_2, te_2\rangle, ..., \langle tb_j, te_j\rangle ...\langle tb_l, te_l\rangle\}$, where l is the total number of saturated vibration regions in the random vibration process. tb_j and te_j stand for the starting and the ending times of jth saturated vibration region. For each $\langle tb_j, te_j\rangle$, ToAuth calculates $\langle \widetilde{tb_j}, \widetilde{te_j}\rangle$, where $\widetilde{tb_j} = tb_j$ mod $(2\delta + 1)$, $\widetilde{te_j} = te_j$ mod $(2\delta + 1)$. Alice gets $\widetilde{T}$ =$\{\langle \widetilde{tb_1}, \widetilde{te_1}\rangle, \langle \widetilde{tb_2}, \widetilde{te_2}\rangle, ..., \langle \widetilde{tb_j}, \widetilde{te_j}\rangle ...\langle \widetilde{tb_l}, \widetilde{te_l}\rangle\}$. Besides, for each $\langle tb_j, te_j\rangle$, Alice calculates $H(\langle tb_j, te_j\rangle)$ and get $\mathbf{H}$ =$\{H(\langle tb_1, te_1\rangle), H(\langle tb_2, te_2\rangle), ..., H(\langle tb_j, te_j\rangle)...H(\langle tb_l, te_l\rangle)\}$. H is a hash function defined in [22], which is predesigned by ToAuth. She sends $\widetilde{T}$ and $\mathbf{H}$ to Bob.

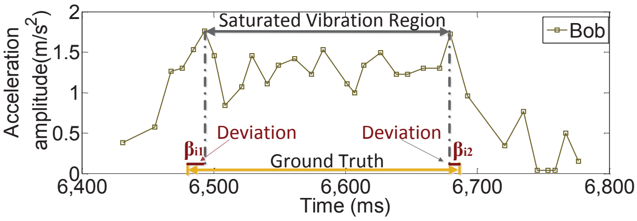


Fig. 9: Saturated vibration region *VS.* Ground Truth

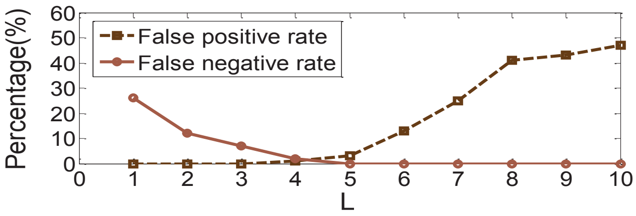


Fig. 10: Performance of various L

Assuming Bob gets T'=$\{\langle tb'_1, te'_1\rangle, \langle tb'_2, te'_2\rangle, ..., \langle tb'_i, te'_i\rangle ... \langle tb'_n, te'_n\rangle\}$, tb'_i and te'_i point to the ith starting and the ending times of saturated vibration regions recognized by the probabilistic model. For each $\langle tb'_i, te'_i\rangle$, he calculates $\langle tb^*_i, te^*_i\rangle$, where tb^*_i= $tb'_i - tb'_i$mod $(2\delta + 1)$ + $\widetilde{tb_j}$, and te^*_i= $te'_i - te'_i$mod $(2\delta + 1)$ + $\widetilde{te_j}$. $\widetilde{tb_j}$ and $\widetilde{te_j}$ are extracted from $\tilde{T}$ in order. If $|tb'_i - tb_j| \leq \delta$ or $|te'_i - te_j| \leq \delta$, we could have tb^*_i =tb_j or te^*_i =te_j. Accordingly, once Bob computes that $H(\langle tb^*_i, te^*_i\rangle)$ equals $H(\langle tb_j, te_j\rangle)$, which is the jth element in $\mathbf{H}$, he puts $H(\langle tb^*_i, te^*_i\rangle)$ to assemble Q. We empirically set δ to be 20ms, which is a little longer than the largest deviation 17ms in our experiments. We provide a more formal description of this process in Algorithm 1.

If the size of Q is larger than L, which is the predesigned threshold of the mutual vibration regions' size detected by ToAuth, Bob sends Q to Alice. Otherwise, the authentication fails. Bob uses $\langle tb^*_i, te^*_i\rangle$, $i \in [1, n]$, whose $H(\langle tb^*_i, te^*_i\rangle)$ in Q to be the mutual vibration regions. Similarly, Alice leverages $\langle tb_j, te_j\rangle$, $j \in [1, l]$, whose $H(\langle tb_j, te_j\rangle)$ in Q to be the mutual vibration regions. Therefore, Alice and Bob achieve mutual vibration region assemble T_{mutual}. Since the performance of ToAuth is related to the threshold L, we conduct 40 groups of experiments to determine L. In the first 20 groups of experiments, we let a pair of smartphones *vibrating together*, which is the same way as Alice and Bob in ToAuth, and then calculate assemble Q_1 by Algorithm 1. The results of these experiments are set to be the positive data. For the left 20 groups of experiments, we ask two smartphones to initiate random vibration simultaneously but separately, and then calculate their assemble Q_2 by Algorithm 1. Their results are collected as negative data. Fig. 10 illustrates the performance of ToAuth with various L by Q_1 and Q_2. The higher L, the lower false negative rate but the higher false positive rate. Based on Fig. 10, we set L to be 5, where the false negative rate drops to zero, while the false positive rate is still low at 7 %.

D. Key Generation

ToAuth sets a bit to 1 for each time slot (ms) in mutual vibration region within T_{mutual}, and sets a bit to 0 for each time slot in the other vibration regions. Accordingly, a bit stream γ could be obtained by each side.

Assume Alice gets a bit stream γ_A and achieves a cryptographic key $sk_A = H(\mu, \gamma_A)$. μ is randomly selected by Alice as a seed. She sends μ to Bob through a public communication channel. Bob uses μ and calculates his cryptographic key $sk_B = H(\mu, \gamma_B)$. γ_B is the bit stream at Bob's side. Therefore, Alice's cryptographic key sk_A and Bob's cryptographic key sk_B are generated.

Algorithm 1 Assemble Q generation

```
Input: $\tilde{T} = \{\langle\widetilde{tb_1}, \widetilde{te_1}\rangle, \langle\widetilde{tb_2}, \widetilde{te_2}\rangle, ..., \langle\widetilde{tb_j}, \widetilde{te_j}\rangle ... \langle\widetilde{tb_l}, \widetilde{te_l}\rangle\}$
    $T' = \{\langle tb'_1, te'_1\rangle, \langle tb'_2, te'_2\rangle, ..., \langle tb'_i, te'_i\rangle ... \langle tb'_n, te'_n\rangle\}$
    $\mathbf{H} = \{H(\langle tb_1, te_1\rangle), ..., H(\langle tb_j, te_j\rangle) ... H(\langle tb_l, te_l\rangle)\}$
Output: Assemble $Q$
function GENERATION $Q(\tilde{T}, T', \mathbf{H})$
    $Q = \emptyset$, $location = 1$, $\delta = 20ms$
    for $i = 1$ to $n$ do
        for $j = location$ to $l$ do
            $tb^*_i$= $tb'_i - tb'_i$mod $(2\delta + 1)$ + $\widetilde{tb_j}$
            $te^*_i$= $te'_i - te'_i$mod $(2\delta + 1)$ + $\widetilde{te_j}$
            if $H(\langle tb_j, te_j\rangle) == H(\langle tb^*_i, te^*_i\rangle)$ then
                $Q = H(\langle tb^*_i, te^*_i\rangle) \bigcup Q$
                $location = j + 1$
                Break
            end if
        end for
    end for
    return $Q$
end function
```

Suppose the total time slots of a random vibration process is T, ToAuth detects that the number of mutual vibration regions is λ with the probability of $1 - \epsilon_1$, $\epsilon_1 \in (0, 1)$. δ is the value defined in feature reconciliation. The length of cryptographic key sk is l bits. These bits have a distance less than ϵ_2 from the uniform distribution over $\{0, 1\}^l$, $\epsilon_2 \in (0, 1)$, which we set to be 2^{-16} in ToAuth. According to [21], the minimal entropy $H_{T,\lambda,\delta}$ of the bit stream γ is $\log \binom{T}{\lambda} - \lambda\lceil\log(2\delta + 1)\rceil$, and the length l of a cryptographic key is $H_{T,\lambda,\delta} + 2 - 2\log(\frac{1}{\epsilon_2})$. For example, in our experiments, $T = 10000ms$, $\delta = 20ms$, if λ in a random vibration process is 40, the entropy $H_{T,\lambda,\delta} = 132.24$, and we can get a cryptographic key with 102 bits which have a distance less than 2^{-16} from uniform distribution over $\{0, 1\}^{102}$. The shortest length of key in ToAuth is 21 bits, where $\lambda = 10$. Thus the minimal key space in ToAuth is 2^{21}.

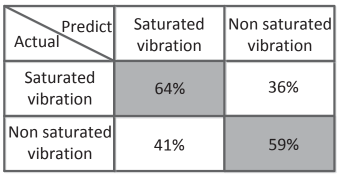
E. Authentication

ToAuth leverages the cryptographic keys of Alice and Bob to make authentication. The steps are presented as follows:

1) Alice selects one message $\mathcal{M}_\mathcal{A}$ stochastically and encrypts it by her cryptographic key: $\mathcal{C}_\mathcal{A}$ = $En_{sk_A}(\mathcal{M}_\mathcal{A})$. $\mathcal{C}_\mathcal{A}$ is a ciphertext. She sends $\mathcal{C}_\mathcal{A}$ to Bob. Similarly, Bob selects one message $\widetilde{\mathcal{M}_\mathcal{B}}$ stochastically and encrypts the message using his cryptographic key:$\widetilde{\mathcal{C}_\mathcal{B}} = En_{sk_B}(\widetilde{\mathcal{M}_\mathcal{B}})$. he sends $\widetilde{\mathcal{C}_\mathcal{B}}$ to Alice.

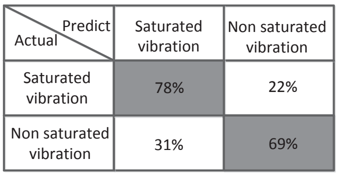
2) Upon receiving the cipher text $\mathcal{C}_\mathcal{A}$ from Alice, Bob decrypts it with sk_B and gets one message $\mathcal{M}'_\mathcal{A}$, Bob re-encrypts message $\mathcal{C}'_\mathcal{A}$=$En_{sk_B}(\mathcal{M}'_\mathcal{A} + \mathcal{C})$, where $\mathcal{C}$ is a constant number predesigned by ToAuth. Bob sends $\mathcal{C}'_\mathcal{A}$ to Alice. In the same way, Alice obtains $\widetilde{M'}_B$ by her cryptographic key and re-encrypts message $\widetilde{\mathcal{C}'_\mathcal{B}}$=$En_{sk_A}(\widetilde{\mathcal{M}'_\mathcal{B}} + \mathcal{C})$, she sends it to Bob.

3) After receiving the cipher text $\mathcal{C}'_\mathcal{A}$, Alice decrypts it by sk_A and gets $\mathcal{M}''_\mathcal{A}$. If $\mathcal{M}''_\mathcal{A}$=$\mathcal{M}_\mathcal{A} + \mathcal{C}$, she acknowledges



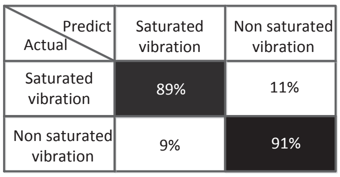
Actual \ Predict	Saturated vibration	Non saturated vibration
Saturated vibration	64%	36%
Non saturated vibration	41%	59%

Fig. 11: Confusion matrix of bus



Actual \ Predict	Saturated vibration	Non saturated vibration
Saturated vibration	78%	22%
Non saturated vibration	31%	69%

Fig. 12: Confusion matrix of subway



Actual \ Predict	Saturated vibration	Non saturated vibration
Saturated vibration	89%	11%
Non saturated vibration	9%	91%

Fig. 13: Confusion matrix of office

that Bob is a legal user. Otherwise, Bob is an illegal user and communication stops. Bob checks Alice by $\widetilde{\mathcal{M}''_{\mathcal{B}}}=\widetilde{\mathcal{M}_{\mathcal{B}}}+\mathcal{C}$ in the same way.

F. Security Analysis

ToAuth is able to prevent MITM attack for the following reasons. Firstly, since the generation of session key is based on the vibration interval, whose duration is generated randomly, it is almost impossible for attackers to acknowledge this message. Secondly, once an attacker tries to change the interactive information, the inconsistent session keys of two parties will alarm this abnormal case and then the communication stops, and thus ToAuth terminates the MITM attack timely.

VI. Experimental Evaluation

ToAuth is implemented as a demo process in Java that runs on smartphones with android platform. We choose Samsung Galaxy S4 I9508 (2GB RAM, quad-core 1.964 GHz) with Android 4.2.2 platform, and TCL Y910T (2GB RAM, quad-core 1.5GHz) with Android 4.0.3 platform to do our system evaluation. In this section, we detail the methodology and results of our experiments.

A. System performance

We evaluate system performance in three distinctive environments (including on a running public bus, on a running subway and in an office), and conduct 36 groups of experiments. The surroundings of public bus and subway reflect the performance of ToAuth in dynamic environments, and the condition of office represents its performance in stable environments. The goal of these experiments is to demonstrate the pervasiveness of ToAuth under various surroundings.

Fig.14 presents the success rate of cryptographic key generation in three environments. In each environment, we conduct 12 experiments. In the first six experiments, Samsung S4 takes the role of Alice to initiate random vibration, and TCL, who plays as Bob, senses forced vibration. And then the two smartphones reverse their roles to do the left experiments. In Fig.14, ToAuth achieves the best performance when it is conducted in office, where the success rate is 90% on average. The average success rate of key generation in subway is about 74%, which is lower than that in the office. And the worst case appears on the bus, the success rate only arrives at 62%. It is due to the fact that interferences in the running bus or the running subway are much fiercer than those in the office, which deteriorate the success rate of cryptographic key generation. We analyze this phenomenon more specially by confusion matrix. In Fig.11, classification accuracy of saturated vibration region and that of non-saturated vibration region on bus are 64% and 59 % respectively, which make serious negative influence on the success rate of key generation. When buses are running, their sudden stops or bumps would cause the change of acceleration, and accelerometer is more likely to mistake these changes as vibration. Thus the probability of key generation's failure increases. Fig.12 provides the classification result of ToAuth on subway. The classification accuracy of saturated vibration region reaches 78%, and the accuracy of non-saturated vibration region arrives at 69%. Both of them increase around 10% compared with those in the bus. It stems from the fact that subways run usually more stably than buses, and thus the influence of inference falls down. The confusion matrix of office is presented by Fig.13. It achieves remarkable classification accuracy results: 89% in saturated vibration region and 91% in non-saturated vibration region. Intuitively, the static environment promises high stability when ToAuth is running. Less additional interference from environment would impact the process of key generation. About 10% misclassification might be due to the detection misjudge in the modules of ToAuth. Overall, ToAuth could get reasonable classification results of saturated vibration regions and non-saturated vibration regions. Especially, the distinguished effect when it conducts in stable environments promises an outstanding success rate of key generation.

TABLE I: Time cost of various module

ToAuth Module	S4	TCL
Random vibration	262ms	287ms
Saturated vibration region recognition	173ms	175.5ms
Feature reconciliation	261ms	263ms
Key generation	104ms	153ms

We make further performance comparison between ToAuth and a representative shaking approach proposed by [10] under three scenarios in Fig.15. We can learn that the average performance of ToAuth is superior to that of shaking method under all scenarios, especially in the subway. Even though ToAuth is rid of the manual assistance, its authentication ability is still advanced in protocols belonged to the same category based on acceleration.

B. System Overhead

1) CPU share: We investigate CPU share of ToAuth in daily life on Samsung S4 and TCL. We installed a process monitor software in each smartphone to record 10 groups of experiments. We find the average CPU share of S4 is only 0.3%. The average CPU share of TCL is 0.5%, which is little higher than that of S4. However, both of them are less than 1%. It is a little overhead for the CPU in smartphones.

2) Running time: We test running time of four main modules: random vibration, saturated vibration region recognition, feature reconciliation and key generation. The running time of user authentication is done in constant time, which relies on the quality of wireless public communication channel. For the random vibration module, we only calculate the time cost that acceleration value converts to acceleration amplitude. The running time of vibration is not included. Table I illustrates the average time cost of these modules by 40 groups of experiments. For each module, we conduct 10 groups of experiments. We utilize Samsung S4 to initiate vibration and

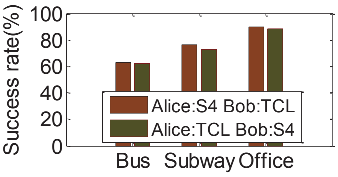


Fig. 14: Success rate

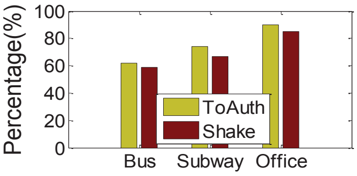


Fig. 15: ToAuth vs. Shaking method

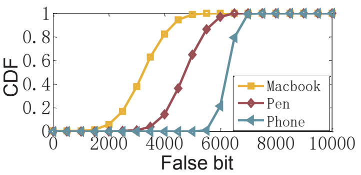


Fig. 16: CDF of false bits

TCL to sense it 5 times. And in the other 5 groups, the roles reverse. The average time cost of each module is recorded in Table I. We observe the average time costs of S4 is little less than that of TCL. It is because of the outstanding CPU. However, TCL still costs 878.5ms to complete cryptographic key generation. The short time promises our protocol could complete user authentication process in reality.

C. Impact of vibration noise

Most smartphones generate noise during vibration. Whether this noise could be exploited by adversaries nearby to achieve the periods of saturated regions and non-saturated regions? To demonstrate ToAuth is able to prevent the adversaries relying on vibration nosie in common cases, we select TCL, which generates loud noise when the engine is running as Alice, and choose Samsung S4 as Bob to do random vibration for 10s in a quiet office room. Meanwhile, we select three types of common recorder devices to record the vibration noise: Free Sound Recorder 9.5.1 [23] installed in a MacBook, a recorder pen (Philips VTR8000 8GB), and a smartphone (HTC Sensation G14). The three devices lie within 5cm from the vibration phones. We conduct 30 groups of experiments and each device records the surrounding noise during the random vibration process 10 times. For each experiment, we label the vibration periods from the recorder based on their acoustic characters manually, and then calculate the saturated vibration regions according to their definitions. During the saturated vibration regions of Alice and recorders, we convert each time slot to 1. In those non-saturated vibration regions, we mark each time slot as 0. Thus we leverage the bit stream from the recorder to compare with the bit stream generated by Alice. Fig16 illustrates CDF of false bits detected by these recorders compared with the bits of ToAuth. 90% false bits recorded by smartphone G14 lie from around 5500 to 7000. The least false bits are more than 5000. The range of false bits recorded by pen spreads from about 3000 to 6000, which are less than those recorded by the phone, but much more than those recorded by the software in Macbook. In Macbook, 80% false bits lie from 1800 to 4000, and 10% false bits are under 2100. It is because the software could provide a high precision recording effect. Nevertheless, the least number of false bits generated by Macbook is longer than 1500, which is still hard for adversaries to crack. Therefore, ToAuth is able to get rid of the attacks depended on vibration noise.

VII. Conclusion

This paper proposes an automatic and practical near field authentication mechanism for smartphones. To guarantee the performance of ToAuth, we have proposed and validated the approaches implemented in it. Since ToAuth requires little human assistance, it is hard to be observed and emulated by adversaries. High security and low efforts make ToAuth well suitable for near field authentication.

VIII. Acknowledgment

This work is supported in part by the NSFC Major Program 61190110 and NSFC under grant 61272429.

Reference

[1] R. Want, “Near field communication,” *Pervasive Computing, IEEE*, vol. 10, pp. 4–7, 2011.
[2] S. McHugh and K. Yarmey, “Near field communication: Introduction and implications,” *Journal of Web Librarianship*, vol. 6, pp. 186–207, 2012.
[3] C. Xu, S. Li, G. Liu, Y. Zhang, E. Miluzzo, Y.-F. Chen, J. Li, and B. Firner, “Crowd++: Unsupervised speaker count with smartphones,” in *ACM UbiComp*, 2013.
[4] U. Meyer and S. Wetzel, “A man-in-the-middle attack on umts,” in *Proceedings of the 3rd ACM workshop on Wireless security*, 2004.
[5] A. Studer, T. Passaro, and L. Bauer, “Don’t bump, shake on it: the exploitation of a popular accelerometer-based smart phone exchange and its secure replacement,” in *Proceedings of the 27th Annual Computer Security Applications Conference*, 2011.
[6] L. Li, X. Zhao, and G. Xue, “Near field authentication for smart devices,” in *INFOCOM, 2013 Proceedings IEEE*. IEEE, 2013.
[7] N. Saxena and J. H. Watt, “Authentication technologies for the blind or visually impaired,” in *Proceedings of the USENIX Workshop on Hot Topics in Security (HotSec)*, 2009.
[8] “Bump technologies.bump.” http://bu.mp/.
[9] D. Bichler, G. Stromberg, M. Huemer, and M. Löw, “Key generation based on acceleration data of shaking processes,” in *UbiComp 2007: Ubiquitous Computing*, ser. LNCS, J. Krumm *et al.*, Eds., vol. 4717. Springer, 2007, p. 304317.
[10] R. Mayrhofer and H. Gellersen, “Shake well before use: Authentication based on accelerometer data,” in *Pervasive computing*, ser. LNCS, A. LaMarca *et al.*, Eds., vol. 4480. Springer, 2007, p. 144161.
[11] J. M. McCune, A. Perrig, and M. K. Reiter, “Seeing-is-believing: Using camera phones for human-verifiable authentication,” in *Security and privacy, 2005 IEEE symposium on*, 2005.
[12] I. Goldberg, “Visual key fingerprint code,” 1996.
[13] D. Balfanz, D. K. Smetters, P. Stewart, and H. C. Wong, “Talking to strangers: Authentication in ad-hoc wireless networks,” in *NDSS*, 2002.
[14] B. Niu, X. Zhu, H. Chi, and H. Li, “Privacy and authentication protocol for mobile rfid systems,” *Wireless Personal Communications*, pp. 1–19, 2014.
[15] J. Han, C. Qian, D. Ma, X. Wang, J. Zhao, P. Zhang, W. Xi, and Z. Jiang, “Twins: Device-free object tracking using passive tags,” *arXiv preprint arXiv:1308.6805*, 2013.
[16] J. Han, D. Ma, C. Qian, W. Xi, W. Zhi, Z. Jiang, and S. Longfei, “Cbid: A customer behavior identification system using passive tags,” in *ICNP*, 2014.
[17] A. Odlyzko and E. Rains, “On longest increasing subsequences in random permutations,” *Contemporary Mathematics*, vol. 251, pp. 439–452, 2000.
[18] Y. Yang, “An evaluation of statistical approaches to text categorization,” *Information retrieval*, vol. 1, pp. 69–90, 1999.
[19] P. Mantalos, “Three different measures of sample skewness and kurtosis and their effects on the jarque-bera test for normality,” *International Journal of Computational Economics and Econometrics*, vol. 2, pp. 47–62, 2011.
[20] R. V. Hogg and A. Craig, “Introduction to mathematical statistics,” 1994.
[21] B. Azimi-Sadjadi, A. Kiayias, A. Mercado, and B. Yener, “Robust key generation from signal envelopes in wireless networks,” in *Proceedings of the 14th ACM conference on Computer and communications security*, 2007.
[22] I. B. Damgård, “A design principle for hash functions,” in *Electronic Proceedings of the Crypto and Eurocrypt Conferences*, 1998.
[23] “Free sound recorder 9.7.1,” http://free-sound-recorder.updatestar.com/.