# GeneWave: Fast Authentication and Key Agreement on Commodity Mobile Devices

Pengjin Xie, Jingchao Feng, Zhichao Cao, Jiliang Wang
School of Software, Tsinghua University, China
{xpj15, fengjc14}@mails.tsinghua.edu.cn, {caozc, jiliangwang}@tsinghua.edu.cn

***Abstract*—Device-to-device (D2D) communication is widely used for mobile devices and Internet of Things (IoT). Authentication and key agreement are critical to build a secure channel between two devices. However, existing approaches often rely on a pre-built fingerprint database and suffer from low key generation rate. We present GeneWave, a fast device authentication and key agreement protocol for commodity mobile devices. GeneWave first achieves bidirectional initial authentication based on the physical response interval between two devices. To keep the accuracy of interval estimation, we eliminate time uncertainty on commodity devices through fast signal detection and redundancy time cancellation.Then we derive the initial acoustic channel response (ACR) for device authentication. We design a novel coding scheme for efficient key agreement while ensuring security. Therefore, two devices can authenticate each other and securely agree on a symmetric key. GeneWave requires neither special hardware nor pre-built fingerprint database, and thus it is easy-to-use on commercial mobile devices. We implement GeneWave on mobile devices (i.e., Nexus 5X and Nexus 6P) and evaluate its performance through extensive experiments. Experimental results show that GeneWave efficiently accomplish secure key agreement on commodity smartphones with a key generation rate 10x faster than the state-of-the-art approach.**

## I. Introduction

Device-to-device (D2D) communication has been widely used as the fast development of mobile and Internet of things (IoTs) technology in recent years. For example, mobile and IoT devices use D2D communication for file sharing, mobile paying, data collection, etc. Despite of its prevalence and convenience, D2D communication has security vulnerability issues in practice. It faces attacks such as eavesdropping, impostor attacks, and man-in-the-middle attacks due to the use of open communication channels [19][5]. For example, it is common that a wearable device (e.g. smart watch) shares health data with a mobile device through open channels. Under an insecure communication channel, private data such as personal identity information, health conditions, and movement trajectory is easily leaked.

To support secure D2D communication in open wireless channels such as Wi-Fi, BlueTooth and ZigBee, device authentication and key agreement should be performed among mobile devices. Before communication, two devices authenticate each other and agree on a symmetric key. Then those two devices can build a secure communication channel by using the symmetric key to encrypt their data.

Secure device authentication and key agreement among mobile devices have attracted many efforts. A large portion of methods use the physical proximity of devices as the feature for device authentication [19][18][11][22][17][12][13]. Those methods are based on an observation that two devices in physical proximity can usually obtain some similar physical information. In the scenario of pairing devices without prior secure associations, two devices have no prior knowledge of each other. The major information they can obtain in authentication is the physical proximity. Therefore, those proximity-based authentication methods are reasonable and necessary for grouping mobile devices on their first acquaintance. For example, they may use similar radio related information such as RSSI and CSI [19][18][11][22][17][12], ambient audio and luminosity [13], and time-varying channels [10][15][21] as a proof of physical proximity for pairing two devices.

However, those methods suffer from practical problems. First, they may rely on dedicated hardware, e.g., CSI based methods [19][18][11][22][9] use Intel 5300 network card which is not used on commercial mobile devices at present. Second, they may not be easy to use in practice. For example, they may require the antennas of the two devices to be very close to each other, e.g., 5 cm [19], 1.25 cm [12]. In practice, it is difficult to achieve especially for embedded antennas on mobile devices. Third, many methods [12][17][3][14][20], which obtain symmetric keys from received signal strength, are vulnerable to predictable channel attacks. Attackers can intentionally block/unblock the signal to the devices to create predictable signal patterns. Last but not least, many methods [12][17][13] are not efficient since they require a long time for key agreement.

Moreover, fingerprinting-based methods [8] [2] [16] [4] [23] [6] leverage hardware properties. [4][23][6] exploit the frequency response of speakers by sound signal with special frequency pattern, and use it to generate fingerprint in authentication. [1] and [7] use hardware (e.g., accelerometer) to generate fingerprint in authentication. Those methods need to learn the fingerprint or share a common fingerprint database in advance.

To address those problems, we propose GeneWave, a general device authentication and key agreement method for secure D2D communication. Instead of using pre-built fingerprint database, two devices in physical proximity authenticate each other by the bounded acoustic round-trip traveling time. GeneWave achieves authentication and key agreement by the following two major steps: *bidirectional initial authentication* and *key agreement*. During authentication, we derive unique features of the acoustic channel, i.e., acoustic channel response

978-1-5090-6501-1/17/$31.00 © 2017 IEEE

(ACR) from two devices for bidirectional initial authentication. In key agreement, we propose a sine wave based pulse coding method to efficiently encode the symmetric key on the acoustic signal. After receiving the acoustic signal, the device can decode the symmetric key as well as verifying the identity of the signal source using the ACR. GeneWave does not require pre-shared fingerprint database.

The practical design of GeneWave faces several challenges. First, how to efficiently distinguish the traveling time difference for two devices, especially in the presence of uncertainty of signal detection and dynamic hardware processing delay. We propose an efficient FFT based fast signal detection approach for signal detection. Meanwhile, we also design a time cancellation method to eliminate hardware latency. The second challenge is how to embed information of secret key in the acoustic signal. The ACR features may not be accurately derived from the signal encoded with message. To address this, we propose a sine wave based pulse coding method supports both data communication and can effectively extract channel ACR for authentication. The third challenge is how to achieve efficient key agreement while ensuring the security of authentication. Intuitively, there exists a dilemma as more information encoded in the signal will destroy more ACR features used by authentication. We optimize the encoding and decoding scheme to significantly improving the coding density while enhancing the decoding rate.

We implement GeneWave on commercial mobile device including Nexus 5X and 6P with Android 6.0.1. The implementation has no special HW/SW requirement and can run on most Android based mobile devices. The evaluation results show that the key generation rate is 10x faster than latest approach [19]. Two devices can finish authentication and key agreement for secure communication flexibly (e.g., holding them in hand, placing them on a table) rather than putting their antennas very closely. The entire process can be finished in less than 2 seconds, which is faster than most existing approaches and is acceptable in real applications. We believe GeneWave provides an efficient device authentication and key agreement approach for many D2D scenarios.

The contributions of this work are as follows.

- We present GeneWave, an efficient and fast authentication and key agreement method for secure D2D communication based on acoustic signal.
- We propose a sine wave based pulse coding and optimize its performance, which supports efficient data communication while preserving the channel features for authentication.
- We implement GeneWave on commercial mobile devices and verify its effectiveness through extensive evaluations.

The rest of this paper is organized as follows. Section II presents the system model and observations from our experiments. Section III shows the design overview of GeneWave. Section IV and Section V present the detailed design of device authentication and key agreement respectively. Section VI presents the implementation details and evaluation results. We present the related work in Section VII and conclude this work in Section VIII.

## II. System Model and Observations

In this section, we first explain the system and security model. Then we show the feasibility and challenges of authentication and key agreement between two mobile devices.

### A. System and Security Model

Alice and Bob are two legitimate mobile devices, which need to exchange some private data between each other through wireless channels. Unfortunately, there is no secure communication channel for them to exchange data. In order to protect private data from eavesdropping, they need a common session key to encrypt data for establishing a secure channel. To achieve this, Alice and Bob are placed in physical proximity within a certain safe distance. An attacker Eve is located beyond a far distance to Alice and Bob. Eve can receive the acoustic signal from Alice and Bob, and send any acoustic signal to perform various attacks including spoofing and eavesdropping. Eve has complete knowledge of our device authentication and key agreement algorithm. When Eve moves into the safe distance, Alice and Bob will easily notice it. We consider the attackers have unlimited capacity and negligible signal processing delay, but the capacity of legitimate devices is limited.

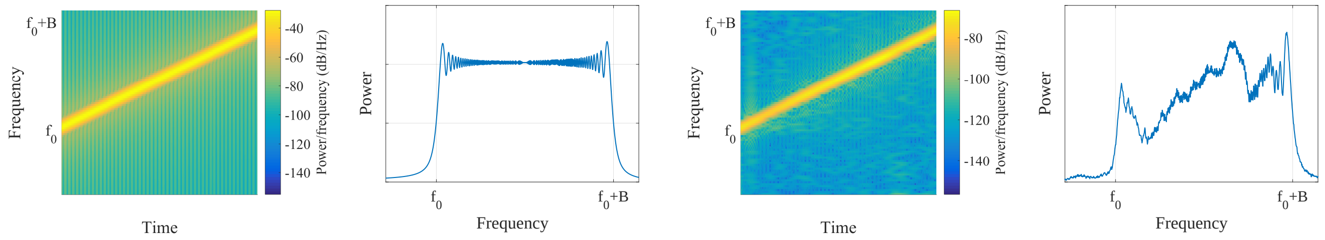
Our goal is to instantly make Alice and Bob agree on a symmetric key for further communication and prevent Eve from knowing the key. This system guarantees security but not availability. For example, Eve can play jamming acoustic signals to prevent Alice and Bob from agreeing on a symmetric key.

### B. ACR based Authentication

In this section, we show the feasibility of using audio channel for authentication. In the experiments, a mobile device $S_1$ transmits a chirp acoustic signal in the range $[f_0, f_0 + B]$ through its speaker. Another mobile device $R_1$ receives the signal using the microphone and records the received signal.

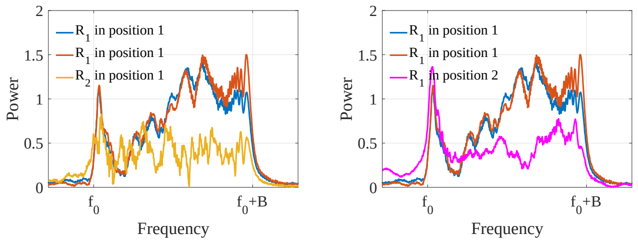
We perform frequency domain analysis on the transmitted and received signal to obtain the time-frequency distribution and energy distribution. Figure 1(a) and Figure 1(b) show the time-frequency distribution and energy distribution on frequency domain of the original chirp signal, respectively. Figure 1(c) and Figure 1(d) show the time-frequency distribution and frequency domain distribution of the received signal, respectively. We can see that the energy distribution of the received signal is different from the original signal due to hardware frequency selectivity and multipath effect. We measure the ***Acoustic Channel Response(ACR)***(i.e., the energy distribution of the chirp signal) as shown in Figure 1(d). We have the following observations for the ACR.

*1) Device Diversity:* We fix the position of sender $S$, and use two receivers $R_1$ and $R_2$ to record the acoustic signal from $S$ at the same position. Figure 2(a) shows the ACR of signal received by $R_1$ and $R_2$. We compare the ACRs in Figure 2(a).



(a) (b) (c) (d)

Fig. 1: (a)-(b)Time-frequency distribution and energy distribution on frequency domain of original chirp signal; (c)-(d)Time-frequency distribution and energy distribution on frequency domain of recorded signal.



(a) (b)

Fig. 2: (a)ACRs on different devices in the same position; (b)ACR on the same device in different positions.

Here $R_1$ records the signal two times in the same position. We can see that the ACRs for the same device in the same position is very similar. However, the ACRs of two devices are significantly different.

*2) Location Diversity:* We use the same receiver $R_1$ to record the acoustic signal form $S$ at two different positions. Figure 2(b) shows the ACR of the signal received by $R_1$. We can see that the ACRs received by the same device in two different positions are significantly different.

*3) Summary:* We can see that: (1) the ACRs are highly similar for the same device on the same position, (2) the ACRs are different for the same device on different positions and (3) the ACRs are significantly different for two different devices.

More specifically, assume the acoustic signal for transmitting is $x$ and the received signal is $y$. The received signal can be denoted as

$$y = x \cdot h_1 \cdot h_2 \cdot h_3 \tag{1}$$

where $h_1$ and $h_2$ are the hardware frequency selectivity parameters of the transmitter's speaker and receiver's microphone, and $h_3$ is the wireless channel response due to multipath and so on. The hardware frequency selectivity is also widely investigated in [4] [23]. Therefore, any change to the speaker, microphone or environment (e.g., multipath) would lead to change of ACR. We can distinguish signals of a static mobile device with ACR.

### C. Feasibility of Initial Authentication

To construct the ACR based authentication. The first problem is how to obtain the initial ACR. As the two legitimate devices are in physical proximity, round-trip time of acoustic signal between them can be shorter than any further device. One of the two legitimate devices (the requester) transmits acoustic signal (request signal) to the other device (the responder). The responder response a signal (response signal) as soon as it detects the request signal from the requester. We define the round trip time between the requester and the responder as *response interval* of the responder to distinguish legitimate devices and attackers.

The distance between two legitimate devices (denoted as legitimate distance) is $d_l$ and the distance from an attacker to those devices (attack distance) is $d_a$. In practice we have $d_a > d_l$. Otherwise, the attacker is easy to be observed by users. Assume a legitimate device transmits a signal and a responder responses with a signal. The response interval can be calculated as $\frac{2d_l}{c} + t_{d(A)} + t_{d(B)}$ where $t_{d(A)}$ is the processing delay on the requester A and $t_{d(B)}$ is the processing delay on the responder B, and $c$ is the sound speed. To verify wether the responder is a legitimate device in physical proximity, we should have

$$\begin{aligned} \frac{2d_l}{c} + t_{d(A)} + t_{d(B)} < \delta \\ \frac{2d_a}{c} + t_{d(A)} > \delta \end{aligned} \tag{2}$$

where $\delta$ is a time threshold. The first equation means the response interval of legitimate device should be less than a threshold while the second equation means the response interval of an attacker should be larger than the threshold. Thus we have $\frac{2d_l}{c} + t_{d(B)} < \frac{2d_a}{c}$ which means

$$d_a - d_l > \frac{t_{d(B)}c}{2} \tag{3}$$

As long as $d_l$ and $d_a$ satisfies Eq. (3), a threshold exists to distinguish the legitimate device from attackers. The attack distance should be at least $d_l + \frac{t_{d(B)}c}{2}$. As a result, when $t_{d(B)}$ is small, the minimum attack distance is small. A user needs to check the area within the minimum attack distance to ensure there is no attacker. Thus it is important that the minimum attack distance is small for mobile devices to exclude attackers.

### D. Challenges

To achieve ACR based authentication, several challenges should be addressed. The first challenge is how to minimize the processing delay $t_{d(B)}$. As shown in Figure 3, the processing

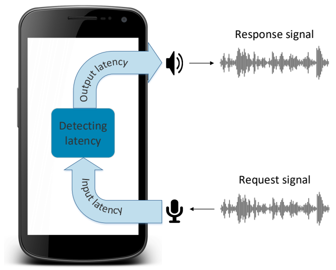


Fig. 3: Audio processing latency is combined by audio input latency, audio detecting latency and audio output latency.

delay $t_{d(B)}$ on most mobile devices consists of the following parts: (1) input latency on B: the uncertain time delay from the signal arrives at B to the signal is received. (2) detecting latency: the time used to detect the requested signal with noise. (3) output latency on B: the uncertain time delay from the audio is played to the time the acoustic signal is emitted. It is difficult to reduce (1), and (3) since they are usually uncertain and impacted by many factors in the operating system. Meanwhile, reducing (2) is also difficult as we need to effectively detect the request signal within a very short time. In this work, we design an efficient detecting method which can detect the request signal with short delay and low computation overhead. The details for minimizing $t_{d(B)}$ are explained in Section IV.

After obtaining the initial authentication information, the second challenge is to achieve efficient communicate and ACR authentication simultaneously. Intuitively, we can encode key on chirp signal. However, how to encode key on chirp signal while preserving the ACR features is a challenge. Meanwhile, a tradeoff is between the coding rate and the ACR similarity observed. More messages coded in a chirp signal would lead to a higher data rate. However, it would also destroy more ACR features and affect the reliability in authentication. Therefore, it is a challenge to find a proper coding method to optimize the performance. In this work, we propose a novel coding/decoding method to address this challenge. The details of the solution are explained in Section V.

## III. GeneWave Overview

### A. Design Goals

From the security perspective, the system has the following goals:

- *Authenticity*: A device can guarantee it is making key agreement with legitimate devices.
- *Confidentiality*: The key should not be exposed to other attackers.
- *Consistency*: The key should be identical on all devices that are making key agreement.

From the performance perspective, the system has the following goals:

- *Fast*: The system should accomplish key agreement in a high speed, e.g., complete authentication and key agreement in a tolerable time.
- *General*: The system should be able to run on commercial mobile devices (e.g., smartphones) without pre-built fingerprint database.
- *Easy-to-use*: The system should be very easy to use, e.g., no need to put the antennas or the mobile devices very close to each other as in traditional approaches [19][12].

### B. Design Overview

We propose an acoustic device authentication and key agreement system, GeneWave, to meet the design goals. Overall, GeneWave system consists of the following two major steps: *bidirectional initial authentication* and *key agreement*.

*1) Bidirectional Initial Authentication:* In GeneWave, for Alice and Bob who want to make key agreement, we assume they have no prior information of each other. Thus they need to do initial authentication to verify the validity of each other. In this step, we use the response interval less than the threshold $\delta$ to distinguish legitimate devices and attackers. Meanwhile, we use ACR features from the response signal to identify a speaker-to-microphone channel. For example, Alice transmits acoustic signal to Bob and authenticates Bob by the response interval. Then, Alice also derives ACR features for the acoustic channel from the response signal that Bob reply to Alice. Similarly, Bob also authenticates Alice and derives the ACR features from Alice's response signal for the acoustic channel from Alice to Bob.

*2) Key Agreement:* The agreement of symmetric key is accomplished by public key system. As shown in Figure 4, Alice encodes her public key $k_p$ into acoustic signal and transmits the signal to Bob. The encoded acoustic signal from Alice should preserve the channel ACR features. Bob decodes Alice's public key after verified whether it is from Alice using ACR features. The message coding should be efficient and be able to tolerate errors in the channel. Then, Bob generates a session key $k_s$ and encrypts it using Alice's public key $k_p$. Assume the encrypted session key is $E_{k_p}(k_s)$, Bob encodes $E_{k_p}(k_s)$ into acoustic signal and transmits the signal to Alice. Alice verifies the signal source is from Bob. Then she decodes $E_{k_p}(k_s)$ and uses her private key to obtain $k_s$. Then the session key $k_s$ can be used by Alice and Bob for further communication. In this progress, the attackers have no opportunity for spoofing due to the identity verification by ACR, and the public key system prevents attackers from deriving the session key $k_s$.

## IV. Initial Authentication with ACR

In this section, we present the design details of device authentication using ACR.

### A. Response Interval Measurement

We use response interval for device verification. Due to the diversity of hardware capability and software processing, it is essential to accurately measure the response interval on different mobile devices. The process of initial authentication between two users (e.g., Alice and Bob) is shown in Figure 5. Alice begins to send a request signal at time $t_1$ and the signal is

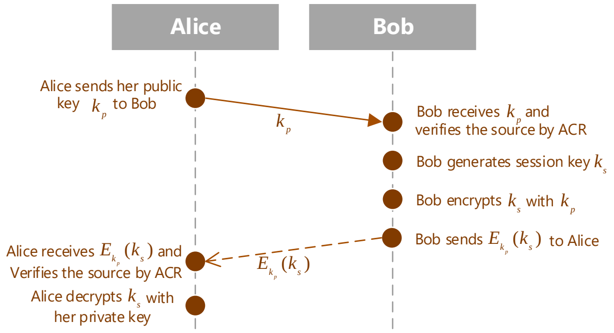

Alice
Bob
Alice sends her public key $k_p$ to Bob
$k_p$
Bob receives $k_p$ and verifies the source by ACR
Bob generates session key $k_s$
Bob encrypts $k_s$ with $k_p$
Bob sends $E_{k_p}(k_s)$ to Alice
$E_{k_p}(k_s)$
Alice receives $E_{k_p}(k_s)$ and Verifies the source by ACR
Alice decrypts $k_s$ with her private key

Fig. 4: Key agreement process.

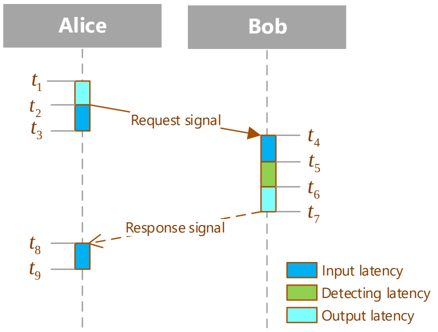


Fig. 5: The process of Alice authenticating Bob.

emitted from Alice's speaker at time $t_2$. Note that $t_1 \neq t_2$ due to the internal processing latency (output latency) on mobile phone. The signal is received by Alice herself at time $t_3$ and reaches Bob at time $t_4$ due to audio input latency. The request signal is received by Bob at time $t_5$. After Bob finishes the request signal detection, he sends a response signal at time $t_6$ to reply. Similarly, due to output latency, the response signal leaves Bob's speaker at time $t_7$ and reaches Alice at time $t_8$. Then Alice is ready to process the response signal at time $t_9$. In order to make the response signal not overlap with the request signal at Alice, the transmission of the request signal must be finished before $t_8$, i.e., the length of request signal must be shorter than $t_8 - t_2$.

The response interval of Bob, i.e., $t_8 - t_2$, indicates the distance between Alice and Bob. However, it is difficult to obtain $t_2$ and $t_8$ at hardware layer. Normally, Alice can measure $t_1$, $t_3$ and $t_9$ at the software layer. This contains Alice's input latency $(t_9 - t_8)$ and output latency $(t_2 - t_1)$. We find that the intervals $(t_3 - t_2)$ and $(t_9 - t_8)$ are both Alice's audio input latency. Although audio latency have diversity on different devices, the audio latency on the same device are usually stable. The intervals $(t_3 - t_2)$ and $(t_9 - t_8)$ should be the same. Thus, we define the response interval of Bob as $t_{d(B)} = t_9 - t_3$ which removes the audio input and output latency.

### B. Request Signal Detection

From Figure 3, we can see the device processing delay on Bob consists of audio input latency, audio output latency and signal detecting latency. The audio input and output latency is determined by the hardware and the detecting latency is related to the processing software. In order to make the response interval as small as possible, we need to reduce those latencies. To reduce the detecting latency of request signal, we design a fast request signal detection method. First, we use the audio strength as a filter criteria. We ignore the signal with average strength less than a pre-configured threshold. This only incurs a linear overhead and can be finished very quickly. Then we apply FFT (Fast Fourier Transform) to check whether the received signal contains the frequency components of the request signal. If the frequency components of the request signal are detected, Bob considers the received signal as a request signal from Alice. In our experiment, the fast signal detection method detects the request signal correctly in 70 audio samples (70/48000 = 1.46 ms).

### C. Bidirectional and Repeated Authentication

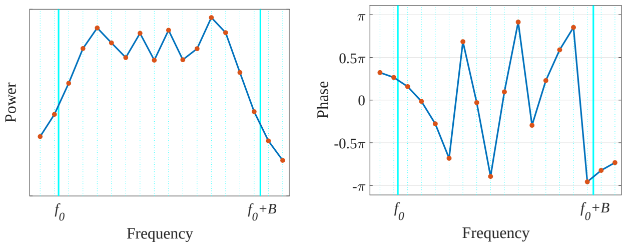
The key agreement process needs Alice and Bob to exchange key while preserving the ACR feature of the bi-direction channels (Alice's speaker to Bob's microphone and Bob's speaker to Alice's microphone). Thus we first need to do device initial authentication bidirectionally. To prevent attackers from guessing the emission time of request signal and playing the response signal in advance, Alice conducts initial authentication for multiple times (five in our implementation) with random time interval. Then she analyzes all response signal. If those ACRs are highly correlated and the response intervals are similar, she considers that those response signal are from the same legitimate device.

### D. Request/Response Signal Design

There are two requirements for request signal design and response signal design. First, the request signal should be able to support quick detection. Second, the response signal should be able to reflect the hardware and multipath frequency selectivity. Thus we propose a double layer signal design, which uses different signal for request signal and response signal. To support fast signal detection, the request signal is a combination of sine waves of several pre-configured frequencies. Meanwhile, to facilitate device authentication using ACR features, the response signal from Bob should be able to reveal the hardware and multipath frequency selectivity. Compared with a combination of sine waves of several pre-configured frequencies, linear chirp signal has an evenly distribution on frequency domain in a certain frequency band as shown in Figure 1 (b). Since the chirp signal has a even frquency distribution on the bandwidth, the channel selectivity can be fully revealed in received signal. Figure 1 (d) shows the channel frequency selectivity according to the intensity of the frequencies in bandwidth. As a result, we choose linear chirp signal as the response signal.

## V. Key Agreement

After bidirectional initial authentication, Alice and Bob can build the ACR features of the channels. Next, they need to perform key agreement based on symmetric key for secure



(a) Power.

(b) Phase.

Fig. 6: Power and phase of the frequencies in $F_S$.

communication. In this section, we show the design details for key agreement.

### A. Encoding While Preserving ACR features

The most important requirement here is to enable key agreement while ensuring security. More specifically, a mobile device is required to encode its key into acoustic signal for key agreement while being able to successfully derive the ACR features for authentication. Intuitively, we can use the frequency changing directions of chirp signal to represent '0' and '1'. For example, the chirp signal that sweeps from $f_0$ to $f_0 + B$ represents '0' and the chirp signal that sweeps from $f_0 + B$ to $f_0$ represents '1'. However, the coding rate for such an encoding scheme is very low.

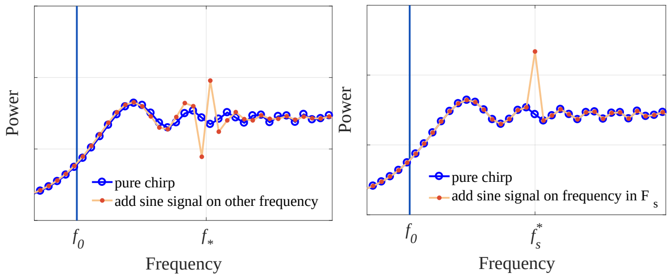
To address these problems, we design a sine wave based pulse coding method. More specifically, we add sine waves of several specific frequencies (these added sine waves are called *message signal*) on the chirp signal to encode key information. We define the combined signal as *overlapped signal*. However, there still exists a challenge to tradeoff the encoding rate (i.e., message transmission speed) and security. If we add more information in the message signal, the message transmission speed increases. On the other hand, it also increases the difficult to derive the ACR features (it is possible that the ACR features cannot be derived), and thus the device authentication may fail.

Next, we introduce how to add message signal on chirp signal while ensuring the efficiency of ACR features extraction.

By applying FFT on the discrete chirp signal, we can get the amplitude and initial phase information on different frequencies between $f_0$ to $f_0 + B$. We define the set of these frequencies as $F_s = \{F_s^1, F_s^2, F_s^3, ..., F_s^{k-1}, F_s^k\}$ as shown in Figure 6.

There are two parameters impacting the performance of sine wave based pulse coding. First, the frequency of the added sine wave signal impacts the performance. We need to examine the impact of the frequency of added sine signal, and search for the best coding position. Second, the phase of the added sine wave impacts the performance. We analyze the characteristic of linear chirp signal and find the best phases for coding.

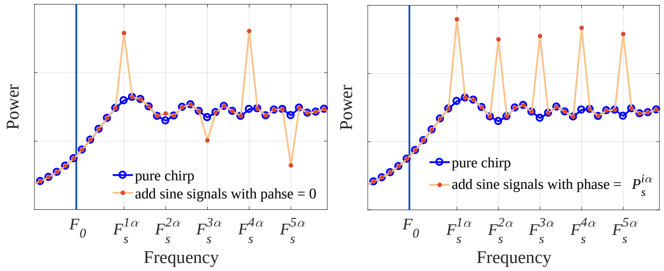
By adding a sine signal of a specific frequency $f^*$ on the original chirp signal, we obtain an overlapped signal. This overlapped signal has a slope line and a horizontal line on the corresponding time-frequency spectrum.



(a) $f^* \notin F_s$.

(b) $f_s^* \in F_s$.

Fig. 7: Different added frequencies can have different performance.



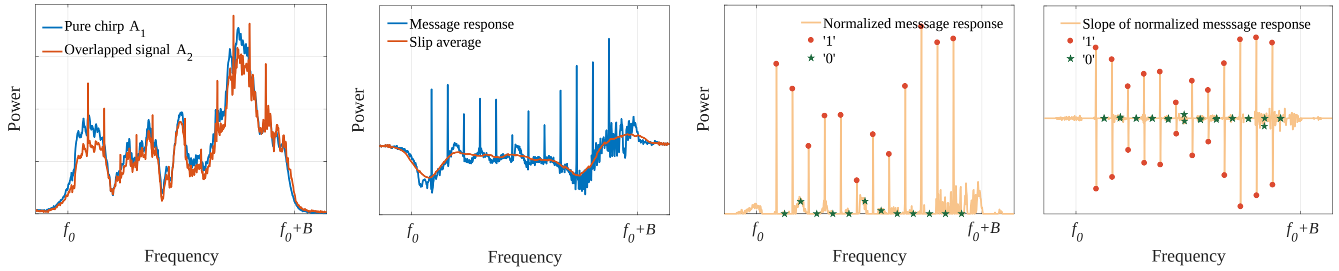
(a) sine signal phase = 0.

(b) sine signal phase = $P_s^{i\alpha}$.

Fig. 8: Different added phases can have different performance.

With different $f^*$, we find that the resulted time-frequency spectrums of the overlapped signal are similar. However, the resulted energy distribution is different. We can see the impact of two different $f^*$ in Figure 7. As shown in Figure 7b, if the added frequency $f^*$ is in $F_s$, the extra energy concentrates on the selected frequency and the energy on other frequencies are almost not changed. Otherwise, as shown in Figure 7a, the energy of multiple frequencies in $F_s$ are impacted and changed. The energy changing of multiple frequencies will further impact the calculation of ACR and decoding the encoded message. Thus, it is better to add sine signal with the frequencies in $F_s$ as it introduces less noise. As a result, we choose the frequency of sine wave from $F_s$ to preserve ACR features as much as possible.

We also examine the impact of added sine wave phases. It should be noted that different frequencies on the chirp signal have different initial phases (as shown in figure 6 (b)). Therefore, we should also consider the impact of the phase of the added sine signal for different frequencies. To show the impact of phase, we add multiple sine signal of different frequencies in $F_s$ on a chirp signal. Figure 6(b) shows the initial phase $P_s^i$ for different frequency $F_s^i$ in $F_s$. First, we set the phases for all added sine waves to 0. We can see that the energy of corresponding frequencies are significantly different as shown in Figure 8(a). The energy at some frequency is absorbed. Moreover, the energy at some frequency is even reduced. This is because the phase of the added signal is not the same with the original signal on the chirp. In such a circumstance, it is difficult to design decoding method for these varying energy distribution. In our encoding, we adjust the initial phase of sine signal to $P_s^i$ for frequency $F_s^i$ in order



(a) (b) (c) (d)

Fig. 9: (a)Frequency domain energy distribution of original chirp and encoded chirp; (b)Message response and the moving window average; (c)Normalized message response with ‘0’ and ‘1’ marked on it; (d)Slope of normalized message response with ‘0’ and ‘1’ marked on it.

to obtain unified energy amplitude on each frequency as shown in Figure 8(b).

For a specific frequency in $F_s$, we add a sine signal with corresponding initial phase to represent ‘1’ and add nothing to represent ‘0’. In order to test the performance of sine wave based pulse coding, we use these 40 coding frequencies (called *encoding frequency*)to encode a binary sequence with ‘1’ and ‘0’ alternately. We let Alice transmit this overlapped signal to Bob. Figure 9(a) illustrates the ACR features from the pure chirp and the overlapped signal. The trends of these two curves are very similar. The correlation coefficient is 0.9761 which indicates that the ACR features is preserved for device authentication. Meanwhile, the encoded information can be effectively decoded.

## B. Decoding

The next task is decoding binary sequence from the encoded chirp. For the frequency domain energy distribution of a received overlapped signal, the fluctuations over frequencies is caused by both the message encoding on coding frequencies and the channel frequency selectivity. It is difficult to separate the information of encoding information (called *message response*) from the impact of channel selectivity (called *channel ACR*). Meanwhile, the channel ACR also introduces noise for message decoding. To address this problem, we find that acoustic channel frequency selectivity is stable with the same devices and the same multipath environment as shown in section II-B. The impact of channel selectivity on the pure chirp and the overlapped signal is the same. We can use the ACR from the pure chirp as the channel ACR of the overlapped signal. As we have obtained the channel ACR features of pure chirp in bidirectional initial authentication, message response of the overlapped signal can be obtained by subtracting the channel ACR.

The frequency domain energy distribution of a received pure chirp $A_1$ and a received overlapped signal $A_2$ are shown in figure 9(a). We can derive message response in figure 9(b) by $A_2 - A_1$. In the message response, the presence of spikes on coding frequencies indicates ‘1’ and ‘0’ otherwise. Meanwhile, we can see that the impact of channel frequency selectivity is very small in the message response as shown in Figure 9(b). In order to remove the base energy difference over frequencies, we normalize the message response by subtracting the slip average of message response as shown in figure 9(b). The normalized message response is shown in Figure 9(c). We can easily distinguish the bits ‘1’ from ‘0’ by their energy levels. To reduce the decoding bit error rate, we further calculate the first derivative of the curve in Figure 9(c). The result is shown in Figure 9(d). Then we decode ‘1’ and ‘0’ based on the resulted energy distribution.

## C. Dual Microphone based Decoding

We also observe that the coding bits are difficult to decode on the frequencies that are in the valley of the ACR curve. Acoustic channel response is very weak on these frequencies, and the sine signal carrying bit ‘1’ on these frequencies is easy to be absorbed. We can see the impact of ACR on sine signal in Figure 9(a) and 9(d), the signal strength is low on frequencies with low ACR. This phenomenon makes the bits encoded in the valley of the ACR curve difficult to decode.

Here, we propose a dual microphone based decoding method to significantly enhance the decoding rate. Most smartphones have a pair of microphones which are usually on the top and bottom respectively. For the same sound signal, these two microphones have different relative positions. Thus, their ACRs are different. More specifically, the peaks and valleys in their ACR curves may be on different frequencies. This is also validated in real experiments. Figure 10 shows the different ACRs of the two microphones on the same smartphone. Therefore, we can combine the signal from two different microphones for decoding. If the decoding results of two microphones for a specific bit are different, we will check the signal condition of two microphones on this bit. We design a measuring method to the exam a “valley-peak” value $V$ for the coding bits. The value of $V$ is with in the range of [0,1]. $V = 0$ means this bit is absolutely in a valley, $V = 1$ means this bit is absolutely in a peak. A value between 0 and 1 means this bit is between a valley and a peak. The microphone which has a higher value of $V$ on the debatable bit will be selected to decide the decoding result. This can significantly reduce the decoding error rate. The detailed result is shown in section VI-B4.

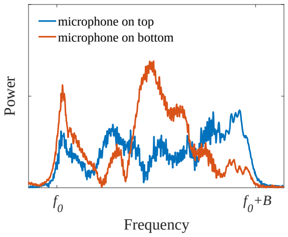


Fig. 10: ACRs of two microphones on smartphone.

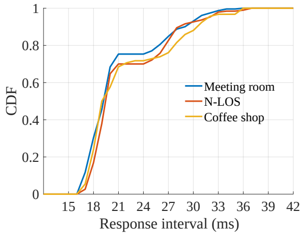


Fig. 11: Response interval CDF in three circumstances.

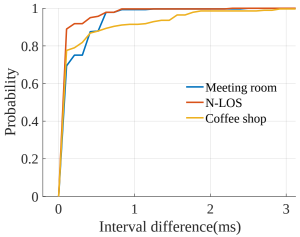


Fig. 12: Response interval deviation in groups.

## VI. Implementation and Evaluation

In this section, we conduct extensive experiments to evaluate our D2D authentication and key agreement protocol.

### A. Implementation and Experiment Settings

We implement our protocol in Android 6.0.1 and evaluate it on two mobile devices (i.e., Nexus 5X and Nexus 6P). The real time audio processing is achieved by Android Fast Mixer audio path. We set the start frequency $f_0$ and the bandwidth $B$ of the chirp signal as $14KHz$ and $7KHz$.In order to evaluate the influence of surrounding environment, we conduct the experiments in three different environments (i.e., meeting room, N-LOS (None Line of Sight) room, and coffee shop). To mitigate the influence of environment changes, we clear all human activity in meeting room. Thus, the multi-path of audio channel keeps stable in meeting room. Moreover, we set up the N-LOS condition in the meeting room by blocking the LOS between a pare of devices with a booklet which is 0.5cm thick. In the coffee shop, the experiments are conducted during business hours. The walking of some customers makes the multi-path dynamic in the coffee shop.

Some variables are further set and measured in different experiment situations. They are defined as follows:

(1) **Device distance** is the distance between the edges of two devices when we place them side by side. In our experiments, we test the performance of our protocol with different device distance.

(2) **Response interval** is the time interval between the initiator receives the request signal from itself and the initiator receives the response signal from responder. We measure the response interval in our experiments of different environments.

### B. Evaluation

The evaluation results that show the influence of response interval, the performance of authenticate device with ACR, and the speed of key agreement are respectively illustrated as following.

*1) Quick Response Evaluate:* To test the performance of quick response in different circumstances, we conduct 3 groups of experiments on Nexus 5x in meeting room, N-LOS meeting room, and coffee shop. Device distance in three circumstances is set as 0.5cm which is the minimal distance in N-LOS environment. We set the sampling rate $f_s$ as $48000Hz$. We respectively take total 283 request-response examination in each environment. Figure 11 shows the CDF (cumulative distribution function) of response interval of all request-response examinations in three circumstances. We can see that the distributions of response interval under different circumstances are similar. This means that the quick response method works well in all three environments. We can see that more than 68% response intervals are less than 21ms in all three circumstances. In the air, the sound speed is constant as $c = 340m/s$. If we set the authentication interval threshold $\delta$ as 21ms, considering the attackers may have 0ms round-trip audio latency, we can get a safe distance $D_a \approx 3.57m$ by Eq. 4. Safe distance means our method can resist the attack from any possible high performance devices out of this distance.

$$D_a = \frac{\frac{Auth_t}{1000} \times c}{2} \approx 3.57m \tag{4}$$

*2) Authentication Efficiency:* In every authentication process, the request-response examination is continuously conducted for five times. If the deviation of the five measured response intervals is small, the responder will be considered as a legitimate device. As shown in Figure 11, we can see the response interval of the same device varies from 15ms to 40ms.

If the response interval is not stable in a group of five request-response examination, the authentication will be not safe that we can not distinguish a legitimate device from an attacker which randomly plays the response signal. We conduct total 50 groups of authentication in three different environments.

Figure 12 shows the CDF of the deviation of each response interval with the group average. We can see that more than 87% percent of deviations are less than 0.5ms. Although the average response interval may vary among different groups, but the response intervals are stable in each group.

We set the threshold to 0.5 ms for guaranteeing the five response signals are from the same device. The five request-response examinations can work well for detecting legitimate response. We set the authentication interval threshold to 21ms in last section. Considering the length of response signal which we set to 8ms, and the response interval difference threshold that we set to 0.5ms, the randomness of intervals between two adjacent authentications varies between 40ms and 80ms will be enough for guaranteeing the response signal will not be covered by the next requesting and preventing attackers from

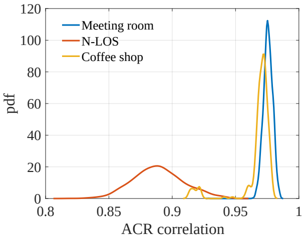


Fig. 13: ACR correlation in three circumstances.

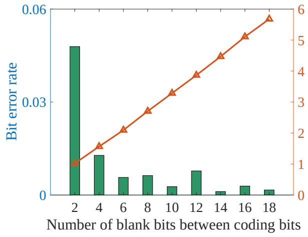


Fig. 14: Bit error rates of different coding densities.

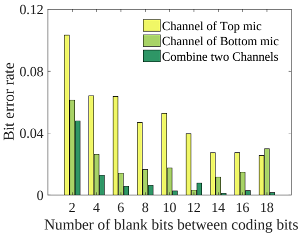


Fig. 15: Bit error rate declines by combining two channels.

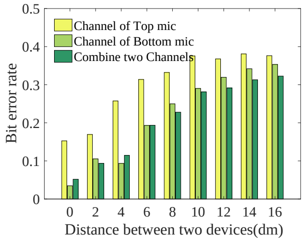


Fig. 16: Bit error rate of different device distance

guessing the time of request signal. Then the bidirectional initial authentication with 10 times request-response examination can be finished in $T \leq 10 * 80ms = 0.8s$.

*3) Performance of Authentication by ACR:* Since we use the ACR to verify device identity while key agreement, the correlation of detected ACR must be highly related with the response signal ACR. We conduct three groups of experiments in meeting room, N-LOS meeting room, and coffee shop to see the correlation between overlapped signal ACR and response signal ACR. Figure 13 shows the PDF (Probability density function) of correlation coefficients between the ACR of overlapped signal and the ACR of the corresponding response signal. We can see that the correlations in the meeting room and coffee shop concentrate in the interval between 0.95 and 1. In comparison, the correlations in N-LOS circumstance mainly distribute in the interval less than 0.95. Although people are frequently walking in coffee shop, the influence on ACR is limited. The N-LOS experiments are taken place in a meeting room with the LOS blocked with a booklet, in this circumstance recorded acoustic signal mainly comes from reflection paths. The multi-path effects on ACR is more obvious in N-LOS circumstance, any tiny change of the multi-path environment (e.g. human walking out of the door, tree out the window swinging with the wind) can effect the ACR. As we can detect attackers in N-LOS, the users only need to exclude attackers in LOS within the safe distance.

*4) The Influence of Coding Density on Transmission Rate:* For the purpose of decoding with our sine wave based pulse coding method successfully, we need to preserve some blank bits between two adjacent coding bits for revealing the pulse shape on coding bits. More blank bits in coding bits interspace means fewer coding bits on the chirp signal. Thus, the more the blank bits, the lower the coding density. Figure 14 shows the bit error rate of decoding by varying the blank bits when we keep the device distance at 3 cm. We can see that the decoding error rate is decreasing with the number of blank bits. This means lower coding density can achieve lower bit error rate. And we can also see the bit error rates are below 0.1 for all the coding density.

To successfully agree on the 2048 bits public key and session key, it is necessary to add enough error correction bits for corresponding bit error rate. If we use RS (reed-solomon) Code as the error correction code, the public key transmit time $T$ for each coding density can be calculated by Eq. 5:

$$T = N_k(1 + 2R_e)\frac{blank + 1}{B} \tag{5}$$

Where $N_k = 2048$ is the length of the 2048 bits public key, $R_e$ is bit error rate, $blank$ and $B$ stand for the number of blank bits between two adjacent coding bits and band width. The line in Figure 14 shows the corresponding $T$ of each coding density. Although the bit error rate is much higher with a high coding density, the time for successfully transmit is shorter with higher coding density. If we choose the coding density with two blank bits between each two adjacent coding bits, the time for transmitting a 2048 bits public key is only 1s. This key generation rate is ten times of TDS[19]. Considering the bidirectional initial authentication with 10 request-response examination take 0.8s, 2048 bits public key transmission time is 1s, and the 256 bits encrypted session key transmission time is 0.1s, our complete authentication and key agreement process can be finished in 2s.

Figure 15 shows the benefit of decoding by combining the two audio channels from top and bottom microphones on the same smartphone. It is obvious that combination of two channels can decrease bit error rate significantly. We can also find that bit error rate of top microphone channel is much higher than bottom microphone. This phenomenon is due to the distance of sound source device with two microphones are different. We place two smartphones side by side in our experiments and the loudspeaker of the source device is on the bottom of it, so the responder's bottom microphone is closer with the sound source than its top microphone. The influence of device distance on decoding error rate is illustrated in Figure 16, We find that the decoding error rate increases with the increasing of device distance.

## VII. Related Work

In the scenario of pairing devices without prior security associations, devices have no prior knowledge of each other, and the only feature that they can obtain for device authentication is the physical proximity. The proximity-based approaches always use location-sensitive features such as received signal strength (RSS)[12][17][3][14][20], and channel state information (CSI) from orthogonal frequency division multiplexing (OFDM) [19][18][11][22][9]. The RSS-based methods suffers a serious disadvantage on the efficient of key agreement, it takes more than one minute for ProxiMate [12] to agree on a

256-bit key due to its key generation rate is less than 5 bits per seconds. These methods are also vulnerable to predictable channel attack. CSI can provide much richer information and lead to a higher key generation rate. However, nowadays CSI can only obtained by Intel 5300 wireless NICs. CSI is very sensitive to location, TDS [19] needs the authentication distance is less than 5cm between antennas of devices for considerable bit error rate. This distance is too close for mobile devices like smartphones which carry built-in network cards.

The hardware fingerprinting-based approaches [8] [2] [16] [4] [23] [6] generate fingerprints based on the complex physical characteristics of the hardware in mobile devices. These methods need to learn the fingerprint or share a common fingerprint database in advance. S2M [4] authenticates devices using the frequency response (FR) of speaker and microphone from two wireless IoT devices and it needs a learning process to obtain the FR in advance. S2M [4] and [23] both consider the RF of the acoustic channel mainly related with hardware and ignore the effect of environment multipath reflection. In our experiments, we find the FR of the acoustic channel both highly related with hardware and multipath reflection, which is illustrated in Figure 2 (b). The fingerprint of learning progress and verification progress must differ a lot with each other when verification is not taken place at the same position of learning progress in S2M.

## VIII. Conclusion

We present GeneWave, a fast authentication and key agreement protocol for commodity mobile devices to agree on a symetric key using acoustic signal. GeneWave first achieves bidirectional initial authentication based on the response interval between two devices. We eliminate time uncertainty on devices through fast signal detection and redundancy time cancellation. We also obtain the initial acoustic channel response (ACR) for authentication through bidirectional initial authentication. To address the challenge of improving coding rate while preserving ACR features, we design a novel encoding scheme to optimize encoding rate in key agreement while ensuring security. Therefore, two devices can authenticate each other and securely agree on a symmetric key. GeneWave does not require special hardware or pre-built fingerprint database, and thus it is easy-to-use on commercial mobile devices. We conduct extensive experiments to show the flexibility and robustness of GeneWave. The experimental results show that GeneWave can achieve a secure and easy-to-use authentication and key agreement for mobile devices. We believe GeneWave provides a convenient way for authentication and key agreement on commodity devices.

## Acknowledgment

This work is in part supported by NSFC under grant 61572277, grant 61532012, grant 61529202.

## References

[1] H. Bojinov, Y. Michalevsky, G. Nakibly, and D. Boneh. Mobile device identification via sensor fingerprinting. *arXiv preprint arXiv:1408.1416*, 2014.

[2] V. Brik, S. Banerjee, M. Gruteser, and S. Oh. Wireless device identification with radiometric signatures. In *Proceedings of MobiCom*, 2008.

[3] G. Chandrasekaran, J.-A. Francisco, V. Ganapathy, M. Gruteser, and W. Trappe. Detecting identity spoofs in ieee 802.11 e wireless networks. In *Proceedings of GLOBECOM*, 2009.

[4] D. Chen, N. Zhang, Z. Qin, X. Mao, Z. Qin, X. Shen, and X.-Y. Li. S2m: A lightweight acoustic fingerprints-based wireless device authentication protocol. *IEEE Internet of Things Journal*, 4(1):88–100, 2017.

[5] N. Cheng, X. O. Wang, W. Cheng, P. Mohapatra, and A. Seneviratne. Characterizing privacy leakage of public wifi networks for users on travel. In *Proceedings of INFOCOM*, 2013.

[6] A. Das, N. Borisov, and M. Caesar. Do you hear what i hear?: Fingerprinting smart devices through embedded acoustic components. In *Proceedings of CCS*, 2014.

[7] S. Dey, N. Roy, W. Xu, R. R. Choudhury, and S. Nelakuditi. Accelprint: Imperfections of accelerometers make smartphones trackable. In *Proceedings of NDSS*, 2014.

[8] S. Jana and S. K. Kasera. On fast and accurate detection of unauthorized wireless access points using clock skews. *IEEE Transactions on Mobile Computing*, 9(3):449–462, 2010.

[9] Z. Jiang, J. Zhao, X.-Y. Li, J. Han, and W. Xi. Rejecting the attack: Source authentication for wi-fi management frames using csi information. In *Proceedings of INFOCOM*, 2013.

[10] L. Lai, Y. Liang, and H. V. Poor. A unified framework for key agreement over wireless fading channels. *IEEE Transactions on Information Forensics and Security*, 7(2):480–490, 2012.

[11] H. Liu, Y. Wang, J. Yang, and Y. Chen. Fast and practical secret key extraction by exploiting channel response. In *Proceedings of INFOCOM*, 2013.

[12] S. Mathur, R. Miller, A. Varshavsky, W. Trappe, and N. Mandayam. Proximate: proximity-based secure pairing using ambient wireless signals. In *Proceedings of MobiSys*, 2011.

[13] M. Miettinen, N. Asokan, T. D. Nguyen, A.-R. Sadeghi, and M. Sobhani. Context-based zero-interaction pairing and key evolution for advanced personal devices. In *Proceedings of CCS*, 2014.

[14] N. Patwari and S. K. Kasera. Robust location distinction using temporal link signatures. In *Proceedings of MobiCom*, 2007.

[15] K. Ren, H. Su, and Q. Wang. Secret key generation exploiting channel characteristics in wireless communications. *IEEE Wireless Communications*, 18(4), 2011.

[16] G. E. Suh and S. Devadas. Physical unclonable functions for device authentication and secret key generation. In *Proceedings of DAC*, 2007.

[17] A. Varshavsky, A. Scannell, A. LaMarca, and E. De Lara. Amigo: Proximity-based authentication of mobile devices. In *Proceedings of UbiComp*, 2007.

[18] W. Xi, X.-Y. Li, C. Qian, J. Han, S. Tang, J. Zhao, and K. Zhao. Keep: Fast secret key extraction protocol for d2d communication. In *Proceedings of IWQoS*, 2014.

[19] W. Xi, C. Qian, J. Han, K. Zhao, S. Zhong, X.-Y. Li, and J. Zhao. Instant and robust authentication and key agreement among mobile devices. In *Proceedings of CCS*, 2016.

[20] L. Xiao, L. Greenstein, N. Mandayam, and W. Trappe. A physical-layer technique to enhance authentication for mobile terminals. In *Proceedings of ICC*, 2008.

[21] S. Xiao, W. Gong, and D. Towsley. Secure wireless communication with dynamic secrets. In *Proceedings of INFOCOM*, 2010.

[22] J. Zhang, A. Marshall, R. Woods, and T. Q. Duong. Efficient key generation by exploiting randomness from channel responses of individual ofdm subcarriers. *IEEE Transactions on Communications*, 64(6):2578–2588, 2016.

[23] Z. Zhou, W. Diao, X. Liu, and K. Zhang. Acoustic fingerprinting revisited: Generate stable device id stealthily with inaudible sound. In *Proceedings of CCS*, 2014.