Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/MBE

MBE, 16(6): 6562–6586.
DOI: 10.3934/mbe.2019327
Received: 01 April 2019
Accepted: 01 July 2019
Published: 15 July 2019

Research article

Ambient audio authentication

Jia-Ning Luo[1], Meng-Hsuan Tsai[2], Nai-Wei Lo[2], Chih-Yang Kao[1] and Ming-Hour Yang[3,*]

[1] Department of Information and Telecommunications, Ming Chuan University, Taiwan
[2] Department of Information Management, National Taiwan University of Science and Technology, Taiwan
[3] Department of Information and Computer Engineering, Chung Yuan Christian University, Taiwan

* **Correspondence:** Email: mhyang@cycu.edu.tw.

Abstract: In the IoT environment, many terminal devices are deployed in unattended areas. If these devices are moved elsewhere by an attacker, the wrong environmental sensing values are obtained, which causes a major disaster. In this paper, we propose an ambient authentication mechanism based on audio to be used in multi-factor authentication by using the ambient sensors equipped with a smart phone. An ultrasonic signal that is not detectable by the human ear was transmitted by the authenticator, and the attenuated signal received by the user being authenticated was transmitted back to the authentication server. The transmitted and received decoded symbol sequences of the audio signal are used to calculated the bit error rate, which is used to measure the relative distance. Our proposed method can narrow the authentication distance to less than 0.5 meters, which can greatly improve the security of the authentication system.

Keywords: multi-factor authentication; ambient authentication

1. Introduction

In recent years, the security of the Internet of Things has become a very important issue, especially the access control and the remote authentication security [1, 2, 3, 4, 5]. In the early stages of Internet-of-Things (IoT) environment, traditional authentication protocols such as text passwords, certificates, or ciphertext-policy attribute-based encryption (CP-ABE) were used to protect a device's identity [6, 7]; but the shortcoming was that sensitive keys were easily stolen in device's un-protected memory. Moreover, text-based passwords are unable to prevent shoulder-surfing attacks [8, 9].

To reduce the use of passwords, identity authentication for devices has progressed to two-factor-based authentication (2FA), multi-factor authentication (MFA), or two-step verification. Particularly, 2FA is a method to verify a legitimate user by combining two distinct identity authentication factors.

A commonly used 2FA scheme combines a password with personal biometrics, such as fingerprints, facial features, brainwave, or plantar biometrics [10, 11] .

The use of fingerprints is a long-standing and widely used method of identifying individuals [12]. Characteristics of fingerprints are persistent, unique, and reproducible. A fingerprint does not change dramatically throughout a person's life, and it returns to its original shape even if it has worn. In particular, reproducibility of prints of the finger is quicker than that of prints of other body parts. Moreover, nearly every fingerprint is distinct, and it is extremely unlikely that any two people have identical fingerprints. However, the use of fingerprint verification is not necessarily secure. Roy et al. [13] revealed that smart phones do not use full-scale fingerprint sensors because the sensing elements required of such sensors are too large. Therefore, the fingerprint sensors used in smart phone are compact versions of larger sensors. For instance, the fingerprint sensor on an iPhone can detect only a central finger area of approximately 0.8 cm x 0.8 cm instead of the whole finger. Therefore, the system compares only partial fingerprints read by the fingerprint sensor with the fingerprints in the system's database, and identity authentication requires only partial compliance. In short, when a user inputs fingerprint data into an authentication database, the fingerprint is successfully matched when any partial fingerprints read by the sensor match any part of the input fingerprint. Roy et al., proposed a fingerprint database comparison to identify a fingerprint using a fingerprint from the sensor size of a smart phone, and the personal fingerprint is called a MasterPrint. The login success rate was 26.46% if five MasterPrints were used during five attempts to log in through identity verification; this was higher than the 0.1% of the false acceptance rate (FAR) of a full fingerprint sensor.

Another biometric method of identification involves face recognition. For example, the Face ID technology in Apple's products uses a TrueDepth camera to retrieve and analyze more than 30,000 invisible points to create a facial depth map and simultaneously retrieves an infrared image of the face [14]. Subsequently, the depth map and infrared images are converted to mathematical representations for comparison with the registered facial data. However, fingerprint and Face ID authentications do not fully protect users' privacy. Users can choose not to disclose their text passwords, but they are unable to stop the retrieval of biometric information by judicial bodies. According to a news source, in August 2018, an American Federal Bureau of Investigation agent requested a suspect use Face ID scanning to unlock his mobile phone during a forensic process.

Companies such as Apple, Google, and Microsoft are increasingly adopting a two-step verification mechanism to enhance the security of identity authentication. The earliest two-step verification mechanism sends a specific website link or single-use password to a user's mobile phone number or email account. The user must enter the single-use password into the authentication system or click the specific website link to complete the authentication process. Nevertheless, this authentication mechanism is unsafe. An attacker can forge an email or text message and add the URL of a phishing website to decoy users into unintentionally revealing their account numbers and text passwords. Users have difficulty identifying those phishing websites because they often use URL shorteners such as goo.gl and bit.ly to redirect; this problem forced Google to shut down its goo.gl redirection service in March 2018 [15].

Another major problem is that sending a single-use password or URL via short message service (SMS) is unsafe [15]. Currently, the bottom layer of the telephone network is based on the signaling system 7 (SS7) network architecture. The SS7 network architecture is not closed; any Type II telecommunication enterprise can easily access the network. As a result, attackers exploit this drawback and

intercept calls and SMS messages to steal single-use passwords. Moreover, government departments can directly monitor a user's communication and block one-time passwords.

Therefore, the use of special apps instead of SMS for authentication messages has gradually become mainstream. Google's two-step verification mechanism (Google authenticator) uses a time-sensitive single-use password mechanism. Furthermore, Google also opened the Google authenticator source code for public use on GitHub [16]. To protect account security, the Fast Identity Online (FIDO) organization, in which Google participates [17], proposed a series of identity authentication mechanisms: the universal authentication framework, universal second factor, and FIDO 2 [18, 19, 20]. In addition to the use of a user authentication system for a mobile device's web services, additional hardware devices were added for identity authentication.

The 2FA, MFA, and two-step verification systems require users and other hardware devices (e.g., mobile phones) to perform authentication. For example, an authentication server sends an authentication code to a user's mobile device, and the user inputs the code in authentication system interface for authentication. In recent years, researchers have aimed to develop additional authentication factors for back-end authentication servers to use to authenticate a device's identity without external operations or hardware. Such authentication mechanisms are called zero-interaction authentication (ZIA). The earliest proposed ZIA method used short-distance communication technology to determine distance between the two IoT devices [21]. For instance, Bluetooth technology allows users to pair mobile phones with a computer and unlock the computer from its idle screen without additional operations. However, this authentication may be susceptible to a relay attack. Studies have proposed various short-distance wireless communication technologies, such as Bluetooth [22, 23], radio-frequency identification [24], and near-field communication [4, 25], can be used for relay attacks. Findings have suggested that their security vulnerability poses a threat to the ZIA system.

Recent studies have proposed distance-bounding techniques to protect ZIA systems from relay attacks [21, 26]. Distance-bounding techniques assume the prover and verifier are bound to the security parameters, where the verifier transmits the challenge to the prover, and the prover accepts and responds to the challenge sent by the verifier. Subsequently, the authentication server calculates the time difference between when the challenge is sent and when the correct response is received and measures the closest distance between the two ends. However, the disadvantage of distance bounding is that any slight error by the prover can cause a large overall error; thus, this technique is difficult to implement effectively on physical devices.

Because modern IoT devices are embedded with sensors that can collect information from their surroundings, studies have proposed them as alternatives to using distance-bounding techniques. Authentication systems can utilize environmental information to prove that two devices are within the same environment. Many studies have proposed the adoption of authentication mechanisms for smartphones because smartphones are popular and equipped with software and hardware that aid authentication systems through what is called as ambient authentication. In [27], four common ambient factors, namely WiFi, Bluetooth, GPS, and audio were mentioned. WiFi uses wireless access points to record the Ethernet media access control address (BSSID) list of devices and the associated received signal strength indicators (RSSIs); Bluetooth records each Bluetooth device address (BDADDR) and RSSI. The actual communication distance of Bluetooth 4.2 is approximately 10m, whereas the indoor communication distance of unobstructed WiFi can reach 169 m. Longer transmission distances can cause larger errors. Moreover, Bluetooth and WiFi have mature detectors that allow attackers to detect transmitted content

easily by remotely connecting to a user's mobile phone [28] and impersonating the user. GPS uses the signal-to-noise ratio of each GPS satellite to determine its ranging code and signal strength. However, GPS cannot be used indoors. Audio authentication technology fetches feature values in ambient audio signals near a device. For example: Schurmann et al. proposed using the change of frequency energy in the ambient audio signal as the feature value [29] .

In a related study of ambient authentication systems, Miettinen et al. proposed a ZIA mechanism for Internet of Things and wearable devices [30] that pairs a user's personal device without the user's participation. The advantage of this mechanism is that it can resist relay attacks, but this method reveals the private information from users in the environment because it collects multiple ambient characteristics from devices for long-term use.

Babins et al. proposed integration of multiple sensing information characteristics to verify whether the authenticator and user being authenticated are within the same environment [31]. Integration of multiple sensors, such as temperature, air pressure, humidity, and altitude sensors, is used to determine the proximity of the authenticator to the user. This method is safer than single-mode sensing techniques, such as those utilizing WiFi, Bluetooth, GPS, and audio, because attackers must simulate multiple environmental features to achieve false authentication. Nonetheless, the downside of multiple sensors is the requirement of additional device-assisted authentication.

Shirvanian et al. proposed a hybrid-bandwidth device authentication mechanism [32] with four novel encryption protocols for WiFi verification that can resist authentication server breaches. One protocol is based on time, and the other three are based on challenges, one of which involves symmetric key encryption and the other two involve public key encryption. These protocols have different authentication mechanisms, which are implemented by a hybrid-bandwidth communication channel that can be established between the device and client browser. The authentication mechanisms are based on (1) one-way and two-way low-bandwidth (19-bit or 6-bit) channels formed by manual personal identification number input or quick response (QR) code; (2) a two-way medium-bandwidth channel established by QR code; and (3) two-way full-bandwidth Bluetooth or point-to-point WiFi channels as well as the combinations of these channels. This results in a total of 13 possibilities offering different security assurance and usability advantages. The advantage of the Shirvanian et al. mechanism is that it can withstand different forms of attacks such as wiretapping and man-in-the-middle attacks on client-to-device communication, including offline and online attacks; the disadvantage is that users must perform this setup process each time they log in to a new computer. This method also requires background monitoring of incoming connections from mobile applications, something currently unfeasible on iOS.

Presently, audio cryptography techniques can bolster the security of authentication mechanisms using the audio features in the environment, and the use of ambient audio requires an apparent noise environment [29]. Consideration of the energy difference between the peak values of the ambient audio during feature calculation enables a password to be derived that possesses high entropy security and is difficult to guess [27, 33, 34, 35]. Schurmann et al. proposed sharing a key among devices that is based on similar ambient audio [29], using the ambient audio to generate the shared key on the device, and using the change in frequency energy in the ambient audio as the feature value. The energy difference of the frequency can lead to a high-security password, but it cannot defend against attackers in the same environment and with similar ambient sound. Karapanos et al. proposed an identity authentication mechanism, Sound-Proof, based on the proximity of ambient sounds [35]; it is a

deployable and developable 2FA mechanism. Sound-Proof analyzes the proximity of the authenticator to the user to be authenticated by comparing the ambient noise recorded by the microphones at both authentication ends; the noise at the two ends must be similar to pass the authentication. In addition, survey results demonstrated that users prefer Sound-Proof to Google 2-Step Verification. Moreover, experimental results verified that ZIA authentication mechanisms based on ambient audio have high usability. In [26], the weakness of Sound-Proof was discovered, and the Sound-Danger attack system was proposed. The attack system comprises active and passive attack modes. Experimental results showed that the proposed attack system had a success rate of up to 100%.

Arp et al. proposed using audio to establish a secret communication channel between devices [36] to transmit a challenge-response authentication message to determine whether the two devices are within the same environment. Schurmann et al. [29] proposed an ambient audio-based secure communication, and Karapanos et al. [35] proposed an ambient audi-based two-factor identity authentication. However, their methods were unable to defend against attackers from the same environment and unable to distinguish attackers with the same sound sources in different environments. The method proposed by Schurmann et al. performs authentication within the same environment and establishes similar environmental modes for various environments. The median of the bit similarity was only 0.05 in their experiments, indicating that attackers could simulate the environment of the user being authenticated for a relay attack.

A passive keyless system based on ambient audio was proposed by Choi et al. [21], which used of the similarity of ambient audio in an existing passive keyless system to prevent two modes of attack, namely out-of-range attacks and record-and-playback attacks. This method can solve the problem of amplified and relay signals encountered in distance-bounding protocol and ultrasonic applications in passive keyless systems. Varshavsky et al.[27] proposed the derivation of keys shared between devices based on WiFi, and the findings revealed that the error was relatively large because the WiFi transmission distance was far. Hien et al. [31] used combinations of four sensing models (WiFi, Bluetooth, GPS, and audio) to detect whether the devices exist in the same environment. Blue et al. proposed an 2MA authentication method by using two microphone to detect the sound source [37]. Wang et al. use audio signals to authenticate users by using the microphone of a smart phone [38].

This study proposed ambient audio authentication based on smart phones that do not require any additional hardware. An ultrasonic signal that is not detectable by the human ear was transmitted by the authenticator, and the attenuated signal received by the user being authenticated was transmitted back to the authentication server. The authentication server determines the symbol error rate (SER) between the original signal and the signal received from the end to be authenticated [36, 39]. Amplitudes of audio signals decrease as the distance increases; thus, the signal-to-noise ratio decreases, causing an increase in the SER. The authentication is successful if the SER is lower than the threshold, and it is unsuccessful if the SER exceeds the threshold. Channel disturbances generated by background noises in disparate environments may result in dissimilar SERs, which may cause difficulty for attackers attempting to simulate a user's environment from another environment. In our experimental result, we can detect an attacker from with 0.5m.

2. Architecture of ambient audio authentication system

In this study, our method using audio to transmit a challenge-response authentication message to determine whether the two devices D1 and D2 are within the same environment (as shown in Figure 1). The process is divided into proximity analysis and the authentication stage (Figure 3). The MFA mechanism proposed in this study mainly consisted of four steps. The following is a detailed description of each step:

1. Password authentication: The user first logs in to the website through password authentication, and the website transmits the authentication request to the authentication server, S. The password will be sent to S for verification. S requests that the user to provide MFA information if the user has passed the password authentication.
2. D1 ambient authentication: S sends a push notification to activate D2's speaker and D1's microphone and allows D2 to transmit an audio signal to D1. D1 receives the audio signal and sends it back to S.
3. D2 ambient authentication: S sends a push notification to activate D1's speaker and D2's microphone. D1 sends a local audio signal to D2 and simultaneously sends the transmitted audio signal back to S.
4. The MFA verification: the authentication server S: Analysis is conducted when S receives the audio signals sent back from D1 or D2. The authentication is successful if the bit error rate (BER) is lower than a predefined value; by contrast, the authentication fails and is discontinued if the BER is higher than the predefined value.

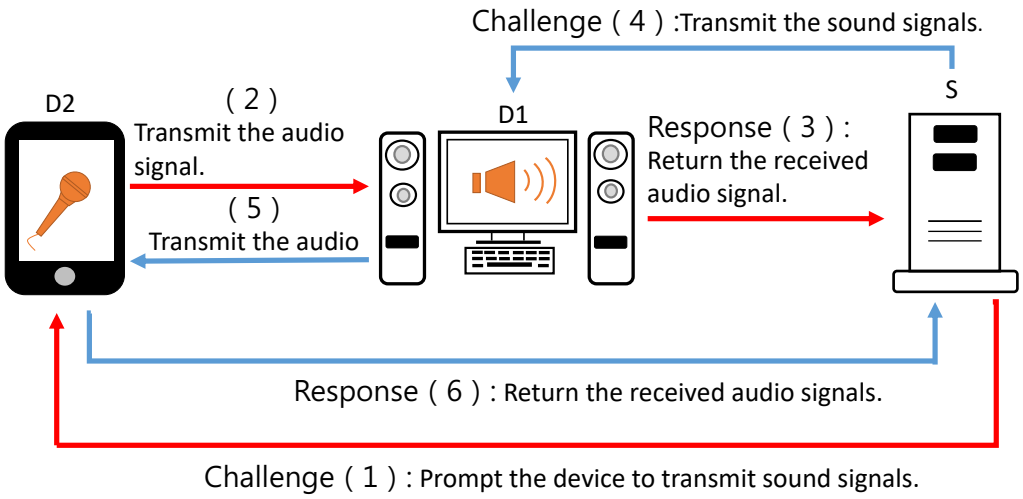

Figure 1. Challenge-Response ambient authentication.

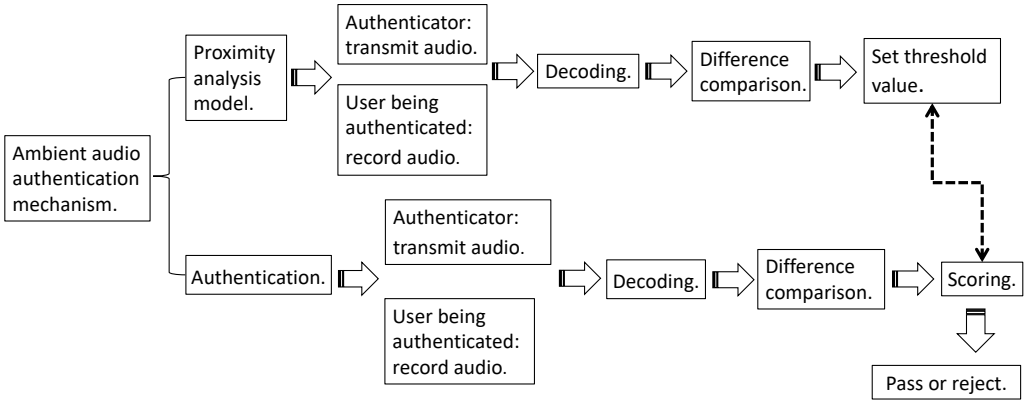

Figure 2. Ambient audio authentication.

2.1. Ambient audio authentication mechanism

This study proposed a method in which the authentication server uses the authentication end and the end to be authenticated to transmit and receive an encoded audio signal and determine whether the ends are within the same environment. The authenticator and user being authenticated in the proximity analysis model send and receive audio signals 100 times, and the system records the transmitted and received signals and decodes them into a symbol sequence. A differential formula is used to calculate the differences between the two symbol sequences. Subsequently, the BER is calculated, and the threshold value is defined. Finally, the threshold value is stored in the authentication server; when the authenticator and user being authenticated send and receive audio signals, the calculated BER is compared with the threshold value. The notations are shown in Table 1.

2.1.1. Proximity analysis

The proximity analysis model of this study is divided into five steps: (a) encoding the symbol sequence of transmitted signals, (b) decoding the symbol sequence of received signals, (c) comparing the two symbol sequences, (d) calculating the BER, and (e) defining the threshold value.

A. Encoding symbol sequence of transmitted signals

The authentication server records the audio sequence number transmitted and received by the authenticator and user being authenticated a total of 100 times. The encoded signal is defined as symbols encoded at different frequencies, where the audio signal $s(t)$ has the ith frequency at time t, and the entire audio signal $s(t)$ is randomly formed and contains 20 frequency codes (Table 2).

Table 1. Notations.

Symbols	*Definitions*
$Symbol$	20 frequency sequences used to transmit audio, $Symbol = \{symbol_1, symbol_2, \ldots, symbol_n\}$, n= 20.
S	Sequence of symbols encoded at the ith point of time with a total of n symbols. $S = \{s_1, s_2, \ldots, s_n\}$
$\#(S)$	Length of the symbol sequence s_i
$x(t)$	Signal model is decoded by the authentication server, and the signal $x(t) = A(t)cos(\phi(t)) = A(t)cos \int_0^t w(t)dt$ changes over time.
$\#(x(t))$	Total duration of the signal
$x_a(t)$	The analytic signal of the original signal $x(t)$
$x_a(f)$	The frequency-domain signal of $x_a(t)$.
A	Amplitude
ϕ	Phase
ω	Frequency
$\omega(t)$	Instantaneous frequency of the signal $x(t)$, $\omega(t) = \phi'(t) = \frac{d}{dt}\phi(t)$.
$frequ$	Instantaneous frequency value in the ith second; n = 4800 is the instantaneous frequency value. $frequ = \{\omega_1, \omega_2, \ldots, \omega_n\}$.
X	Sequence of symbols decoded at the ith second, with a total of n symbols. $X = \{x_1, x_2, \ldots, x_n\}$.
$\#(T)$	Total length of the symbol sequence.
Err	Calculation of the difference between the transmitted and received symbols, with a total of k nonzero vector values. $Err = X - S = \{err_1, err_2, \ldots, err_n\}$
eb	Calculation of the total number of error symbols in the form, $eb = \sum_{i=1}^{k} err_i$.
ber	Calculation of the BER for scoring, $ber = \frac{eb}{\#(s)}$.
Ber	Count the kth calculation result in the storage array of the BER, $Ber = \{ber_1, ber_2, \ldots, ber_k\}$.
$threshold$	Threshold value

Table 2. Frequency and symbol definitions.

Frequency (Hz)	*Symbol*	*Frequency* (Hz)	*Symbol*
16,000	A	16,075	B
16,150	C	16,225	D
16,300	E	16,375	F
16,450	G	16,525	H
16,600	I	16,675	J
16,750	K	16,825	L
16,900	M	16,975	N
17,050	O	17,125	P
17,200	Q	17,275	R
18,000	Start	19,000	END

B. Decoding symbol sequence of received signals

The audio signal received by the authenticator and user being authenticated is sent back to the authentication server for decoding. The signal model decoded by the authentication server is used to define the instantaneous frequency calculated at the independent time t.

First, the original signal $x(t)$ is analyzed by the analytic signal model. The analytic signal simplifies the calculation of the instantaneous frequency. In addition, Hilbert transform is used for analyzing the signal.

The time signal is subjected to Fourier transform to obtain the distributions of different frequency components. After Fourier transform, the complex form is presented with real and imaginary parts, which exhibit positive and negative frequencies. A negative frequency is obtained when the signal passes the Fourier transform. Assuming that the number of sampling points of the time-domain signal is 1,024, the theoretical Fourier transform resolution is 1,024 points, which can resolve a relatively large number of sampling points. According to the Nyquist Theorem, the actual sampling frequency band is not 11,024 Hz but 1512 Hz. Another component of the sampling points is the frequency of imaging range sent back by the Fourier transform, that is, the negative frequency produced by the signal after Fourier transform. Removal of the signal $x(t)$ using the frequency of imaging range from the Fourier transform $x(f)$ can form signal $x_a(f)$, and inverse Fourier transform is applied on $x_a(f)$ to convert the frequency-domain signal into the time-domain signal, which forms the analytic signal $x_a(t)$. The negative frequency component in the Fourier spectrum of the time signal is removed, and the signal $x(t)$ is Fourier transformed to obtain the frequency-domain signal $x(f)$ of the signal $x(t)$, which varies in frequency intensities. The $x_a(f)$ discriminant is established by removing the negative frequency values (set to 0).

The $x_a(f)$ discriminant can be applied to a state equation $sgn(x)$ that varies with x regardless of the value (as long as x is negative (-) and the value of all outputs is -1). By contrast, the value has a positive sign (+) when the value of all outputs is +1; and the value of all outputs is 0 when the value is 0. Adjust the state equation $sgn(x)$ by adding 1 to shift all discriminants up:

Multiply the original signal $x(f)$ by $2 * u(f)$ to remove the negative frequency components from the Fourier spectrum:

$$x_a(f) = \begin{cases} 2 * x(f), f > 0 \\ x(f), f = 0 \\ 0, f < 0 \end{cases} \tag{2.1}$$

Here, $x_a(f)$ can be called an analytical function when the negative frequency components of the Fourier spectrum are removed. However, the final goal is to return to the time sequence for calculation, because this study aimed to calculate the frequency that changes with time for any instant; thus, $x_a(f)$ must be converted into $x_a(t)$ through inverse Fourier transform by using the Hilbert transform algorithm.

When the result of the inverse Fourier of $x_a(t)$ is obtained, it can be used directly to calculate the instantaneous amplitude, phase, and frequency.

Instantaneous frequency is the derivative of instantaneous phase. Suppose the analytic signal $x_a(t)$ is the signal at any real time; its return is actually a complex signal. The real part of a complex number is similar to the original $x(t)$, and an imaginary number is added to the back of each real part. Therefore, difference between the $x_a(t)$ and $x(t)$ is an additional imaginary component that can be used for angle calculation. The angle is an arctangent function of the real and imaginary coefficients, resulting in the phase of $x_a(t)$. Here, arg $(x + i\hat{x})$ indicates that each sampling point integrated both real and imaginary parts, where x represents a real number and $(i\hat{x})$ represents an imaginary number. The coefficients of x and $i\hat{x}$ can be used with the arctangent function to calculate the phase for deriving the instantaneous phase angle.

The angle position of phase ϕ is the angle of a fixed direction, and the vector length generated by simple rotation does not change with respect to the origin. The rotation speed is the angle of a time-unit rotating vector from the start point to the end point. The time interval is $\Delta t = t_2 - t_1$, and the average angular speed is defined as $\bar{\omega}$:

$$\bar{\omega} = \frac{\Delta\omega}{\Delta t} = \frac{(\phi_2 - \phi_1)}{t_2 - t_1} \tag{2.2}$$

Similar to the linear rate, it is the first-order derivative of the linear displacement, where the angular speed is the first-order derivative $\omega = \phi'$ of the angular phase. The calculated first-order reciprocal size is the rotation speed, and its sign represents the rotational direction; positive denotes counterclockwise, and negative denotes clockwise. The first-order derivative of the instantaneous phase is defined as the instantaneous frequency $\omega(t)\phi'(t)$, which is calculated in radians per second.

The symbol rate used in this study is 0.1s; that is, one symbol is transmitted every 0.1s When the sampling rate of the received signals is 48 kHz (48,000 sampling points per s), 4,800 sampling points represent one symbol every 0.1s. In the Hilbert-transformed signal model, an instantaneous frequency value is calculated at each signal sampling point. Therefore, 4,800 instantaneous frequency values can be calculated and stored in the vector ω_i that represents a symbol at 0.1s. A large quantity of data (i.e., 4,800 data) is not required to decode a signal. Therefore, this study's method retrieves an average value out of every 480 data points, and the vector average variance extracted (AVE) contains 10 instantaneous phase data points, for which AVE is used for decoding.

$$AVE_j = \frac{\sum_{i=1}^{N} frequ_{Nj+i}}{N} \tag{2.3}$$

The method for determining the correct symbol is as follows: each frequency f (as shown in Table 2) corresponds to a different symbol M, and the range of the instantaneous frequency value of the decoded symbol should fall within ± 15 Hz of the corresponding frequency. By contrast, the output is zero if the instantaneous frequency does not exist in this range. The decoded results are stored in $T = \{x_1, x_2, \ldots, x_n\}$ to calculate the difference.

Algorithm 1: Algorithm for decoding received signal

input : AVE
output: T

```
for i ← 1 to #(AVE) do
    count_i ← 0
for i ← 1 to #(AVE) do
    count_i ← count_i + 1
    if count_i == 10 then
        for k ← 1 to #(AVE) do
            for j ← 1 to #(Symbol) do
                y ← Symbol(j)
                if y − 15 ≤ AVE(k) ≤ y + 15 then
                    T_i ← 1
                else
                    T_i ← 0
```

C. Comparison of the two symbol sequences

The symbol sequence T is obtained when the signal $x(t)$ is decoded by the authentication server. Subsequently, the difference between the transmitted symbol sequence S and received symbol sequence T (a total of a nonzero vector values) and the sum of the error are calculated to further determine the BER.

D. Calculation of BER

This study determines whether the authenticator and user being authenticated exist in the same environment by calculating and comparing the transmitted and received decoded symbol sequences of the audio signal. Therefore, calculation of BER is required.

E. Defining threshold values

Ambient authentication contains many fluctuations due to the varying conditions of each physical environment, and a signal may be affected even if it is received from less than 0.5m away. Therefore, defining the threshold value is vital for the identification result. Before the authentication, training data of the relationship between the distance and BER in the authenticator server are established, and a critical point that suits the environment is set.

The system uses the following two equations to obtain the threshold value and numerical average of 100 samples. Therefore, during authentication, a BER that is lower than the threshold value passes the authentication, otherwise it fails.

$$Score \leftarrow \sum_{j=1}^{100} \{ber_1, ber_2, \ldots, ber_k\} \tag{2.4}$$

$$threshold \leftarrow \frac{Score}{\#(Ber)} \tag{2.5}$$

2.1.2. Authentication stage

The authentication stage is divided into five steps: (1) transmission of the encoded signal symbol sequence, (2) receipt of the decoded signal symbol sequence, (3) comparison of the two symbol sequences, (4) calculation of the BER, and (5) scoring. Because steps 1 to 4 are similar to the steps in the proximity analysis model, only step 5 is described as follows.

In step 5, the decoded audio signal received by the authenticator and user being authenticated is sent back to the authentication server. The output is subjected to the authentication symbol sequence X_i, and BER analysis is performed for the symbol sequence X_i, and original symbol sequence S_i stored in the authentication server. Authentication is successful if the value is below the threshold value; otherwise, it fails.

3. Experiment

This section is divided into five parts to introduce the experiment design and tools used in the experiment. Section 3.1 provides the experimental scenario and introduces the three experimental structures of this study, including the use of several devices and amplitudes, testing of various environments and distances, transmission of multiple symbols, and execution of one-way and two-way transmission and reception. Section 3.2 describes the encoding method for transmitting audio. Section 3.3 explains that the signal must be accurately synchronized before decoding; otherwise, it may cause decoding errors because symbols exist at each time point after audio encoding. Finally, section 3.4 proposes a method for analyzing ambient noise. The transmission speed of the transmitted audio in this study was 10 bps, and a frequency was transmitted every 0.1 s. The frequency range was 16–19 kHz, and the audio was transmitted with different amplitudes. The user being authenticated transmits the encoded audio signal from the authenticator back to the authentication server, and the authentication server analyzes the BER of the returned audio file from the user's mobile device to determine whether the authenticator and user being authenticated are within the same environment. The authentication is successful if the audio is within the same range; otherwise, it fails. The range was set to 0.5m in this study.

3.1. Experimental scenario

This study conducted three experiments:

Experiment 1 For audio amplitude and distance measurement, two devices were used as the authenticator and the user being authenticated. A total of 20 frequencies were transmitted each time, and eight distances were measured: 0.1, 0.2, 0.3, 0.4, 0.5, 1, 2, and 3 m. A total of five amplitudes

were used: 1, 0.5, 0.1, 0.01, and 0.05. Each distance was tested for each of five amplitudes, with 10 transmissions and receptions per test.

Experiment 2 One-way reception measurement also employed two devices as the authenticator and the user being authenticated; experiments were conducted in two environments. Twenty frequencies were transmitted each time, and five distances were measured: 0.1, 0.5, 1, 2, and 3 m. A total of three amplitudes were used: 1, 0.1, 0.05. Each distance was tested with each of the three amplitudes, with 10 transmissions and receptions per test.

Experiment 3 Two-way reception measurement also used two devices were used as the authenticator and the user being authenticated; automatic transmission and reception applications were designed, and experiments were conducted in three environments. Fifty frequencies were randomly transmitted, and three distances were measured: 0.5, 1, and 2 m. Only one amplitude was used: 1. The same amplitude was used for 100 measurements in different environments at each distance.

Table 3 shows the scenario.

Table 3. Experimental ambient settings.

Features	*Experiment #1*	*Experiment #2*	*Experiment #3*
One-way or Two-way	One-way	One-way	Two-way
# of Ambience Measured	1	2	3
# of Distance Measured	8	5	3
# of Amplitude Measured	5	3	1
Fixed / Random Transmission	Fixed	Fixed	Random

3.2. Audio encoding

This study used a frequency of 16 kHz to 19 kHz to encode secret messages through sound waves. Two frequency modulation methods were considered for use. The first method used two frequencies $f1$ and $f2$ for frequency modulation, and the sines of the waves of $f1$ and $f2$ are 1 and 0, respectively. Furthermore, American Standard Code for Information Interchange (ASCII) encoding was transmitted at 100 bps, and five symbols were transmitted (Figure 3). The experiment demonstrated that the transmission result error rate was high, and the available space for symbol transmission was relatively small. Therefore, this study applied the second method.

3.3. Signal synchronization

The audio signals at the authenticator and user being authenticated must be synchronized for comparison to investigate whether both authentication ends are in the same environment. The study method uses a search for partial cross-correlation to process signal synchronization problems. Because of filtering and other operational delays, offset occurs between the received and transmitted bits. The amount of offset must be determined before comparing two bit sequences for error checking, and one method of doing this is by correlating two sequences and searching for the correlation peak value. The notations are shown in Table 4. Assume that the transmitted signal bits are stored in vector T_x, and the

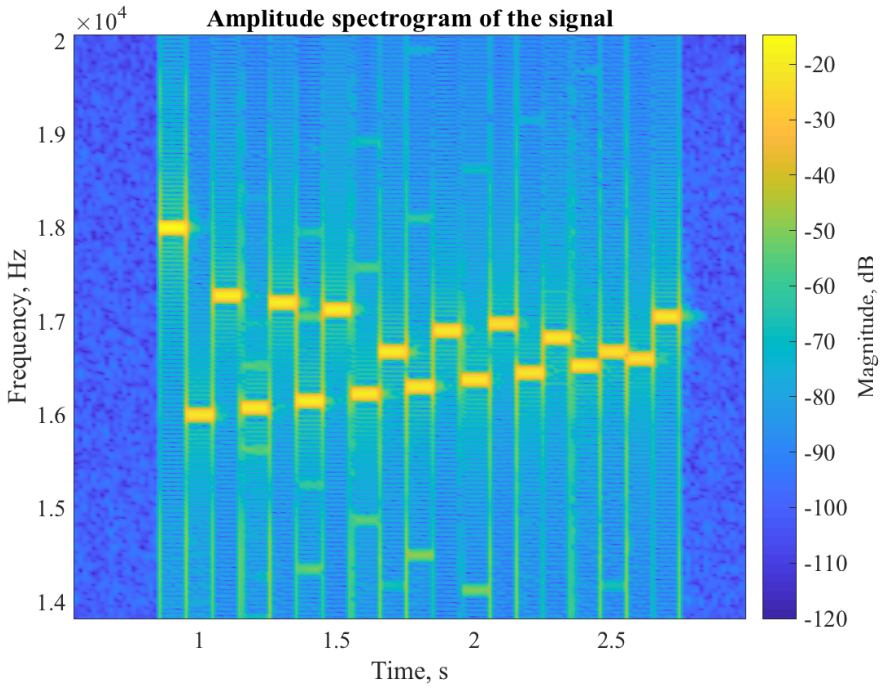

Figure 3. Encoding using various frequencies.

received bits are stored in vector R_x. The received vector should contain more bits than the transmitted vector because the receiver produces a meaningless output during delays in transmission, filling, and flushing processes.

Table 4. Notations.

Symbols	*Definitions*
T_x	Time series of the transmitted signal, $T_x = \{t_{x_1}, t_{x_2}, t_{x_3}, \ldots, t_{x_n}\}$
R_x	Time series of the received signal, $R_x = \{r_{x_1}, r_{x_2}, r_{x_3}, \ldots, r_{x_n}\}$
C_{RT}	Calculate the time-series correlation of T_x and R_x by using the partial autocorrelation function $C_{RT} = \{c_1, c_2, c_3, \ldots, c_n\}$
sb	Marked as the maximum value m of C_{RT}, which is the starting position in the time series for the delayed received signal $sb = C_{RT}[m]$
err	Difference vector between the transmitted and received signals $err = \{e_1, e_2, e_3, \ldots, e_n\}$
eb	Sum of the difference vectors, which is the measurement for signal errors; $eb = \sum_{k=1}^{n} err[k]$
ber	Calculation of signal error rate, which is the probability of a signal error $ber = \frac{eb}{\#(T_x)}$

The possible offset range is between 0 and $\#(R_x) - \#(T_x) - 1$ if the length of the transmitted and

received bit vector are $\#(T_x)$ and $\#(R_x)$, respectively. The offset can be identified by performing a partial autocorrelation function for the two vectors. Algorithm 2 define the partial autocorrelation function.

Algorithm 2: Algorithm of the partial autocorrelation function.

input : R_x, T_x
output: C_{RT}

while $R_x > T_X$ **do**
  for $i \leftarrow 1$ *to* $\#(R_x) - \#(T_x) - 1$ **do**
    $C_{RT}(i) \leftarrow T_{x_i} * R_{x_i}]$

The results showed that vector C_{RT} is the partial autocorrelation of the transmitted and received bits, with 0 to $\#(R_x) - \#(T_x) - 1$ as the possible delay range. Let m be the position of the maximum C_{RT} value, where $sb = C_{RT}[m]$. After identifying the offset between the transmitted and received bit vectors, the signal errors can be calculated. For signals with 0 and 1, a simple difference can cause error. When signal error occurs, the difference is ±1; otherwise, it is 0. Calculating the error vector err from the transmitted bit vector T_x and received bit vector R_x that contains an sb offset uses the following equation:

$$err = T_x - R_x \tag{3.1}$$

The error vector err is a nonzero element with n signal errors. The total number of nonzero elements can be calculated by the total number of bit errors eb:

$$eb = \sum_{k=1}^{n} err[k] \tag{3.2}$$

When performing the calculation of the BER, the transmitted and received signals are synchronized to determined how many errors are in the received bits. The BER is calculated by dividing the number of bit errors by the total number of bits in the transmitted signal. Calculation of the BER is as follows, where eb is the total number of bit errors, and $\#(T_x)$ is the total number of bits transmitted:

$$ber = \frac{eb}{\#(T_x)} \tag{3.3}$$

3.4. Analysis of ambient noise

In reality, noise is randomly normally distributed; this is different from a simulated method. This study used a signal that had been mixed with noise at the receiving end, $\alpha(t) + N_e$, with N_e as the background noise subtracted by the original signal $s(t)$ at the transmitting end to obtain the ambient noise value. Before the signal at the transmitting end is subtracted from the signal at the receiving end, the original signal at the transmitting end and the signal containing the noise at the receiving end must first be normalized, with d representing the channel attenuation.

4. Experimental results

In the experiment we designed, we chose three different environments (Room A, Room B, and Room C). The three rooms contains the two devices (D1 and D2) we are going to authenticate are

placed in the same environment. In each of the three rooms, the background noise is different.

This section analyzes the experimental results of this study, which were mainly divided into three parts. Section 4.1 describes the measurement of audio amplitude and distance. This study tested the effects of five amplitudes at five distances during audio reception. Section 4.2 discusses the investigation of the one-way reception measurement of the authenticator and user being authenticated. This study tested the effects of two environments and three amplitudes at five distances during audio reception. Section 4.3 explores the two-way reception measurement of the authenticator and user being authenticated. This study tested the effects of three environments and one amplitude at three distances during audio reception. Note that this experiment differed from the ones described in Sections 4.1 and 4.2 in that both D1 and D2 transmitted and received, and each transmitted audio signal was randomly generated.

4.1. Audio amplitude and distance measurement

In the past, ambient audio authentication could not resist attackers from the same environment. Therefore, this study proposed to use an audio signal transmission channel that is undetectable by the human ear for MFA and to improve the recognition. Although it is impossible to completely guard against attackers from the same environment, narrowing the range of attacks can enhance security. First, the transmission range of the audio signal must be identified. Because different amplitudes affect the audio signal transmission range, the first experiment tested the audio signals with different amplitudes at different distances. The experimental results showed that when the distance was 0.5 m, the received signal was stable when the amplitude was 1. When the amplitude was not 1, the BER increased as the amplitude of the audio signal decreased. At 1-m range, the BER was increasing significantly because of audio attenuation, indicating that the BER increased as the distance increased.

4.2. One-way reception measurement of the authenticator and user being authenticated

Although the audio transmission range was identified in previous experiment, the experiment was tested only in a single environment. Therefore, this study tested two different environments to confirm the feasibility of the method proposed. The experimental results of the previous experiment showed that resolution of the audio signal did not differ much when the amplitudes were 1 and 0.5. In this section, the audio signal with an amplitude of 1 was selected. When the amplitude of the audio signal was 0.01, performance of the audio transmission was notably lower than with other amplitudes. Therefore, an audio signal with an amplitude of 0.01 was not selected for this experiment. Figure 4 presents the results of audio signals with three amplitudes and five distances measured in Room A. (1) When the amplitude was 1 with distances of 0.1, 0.5, 1, 2, and 3 m, the BERs were 0, 0.1, 0.2, 0.3, and 0.4, respectively. (2) When the amplitude was 0.1 with distances of 0.1, 0.5, 1, 2, and 3 m, the BERs were 0.1, 0.15, 0.25, 0.4, and 0.6, respectively. (3) When the amplitude was 0.05 with distances of 0.1, 0.5, 1, 2, and 3 m, the BERs were 0.1, 0.2, 0.4, 0.7, and 0.9, respectively. Figure 5 presents the results of audio signals with three amplitudes at five distances measured in Room B. (1) When the amplitude was 1 with distances of 0.1, 0.5, 1, 2, and 3 m, the BERs were 0, 0, 0.05, 0.25, and 0.4, respectively. (2) When the amplitude was 0.1 with distances of 0.1, 0.5, 1, 2, and 3 m, the BERs were 0, 0.15, 0.4, 0.6, and 0.9, respectively. (3) When the amplitude was 0.05 with distances of 0.1, 0.5, 1, 2, and 3 m, the BERs were 0, 0.2, 0.5, 0.7, and 0.8, respectively.

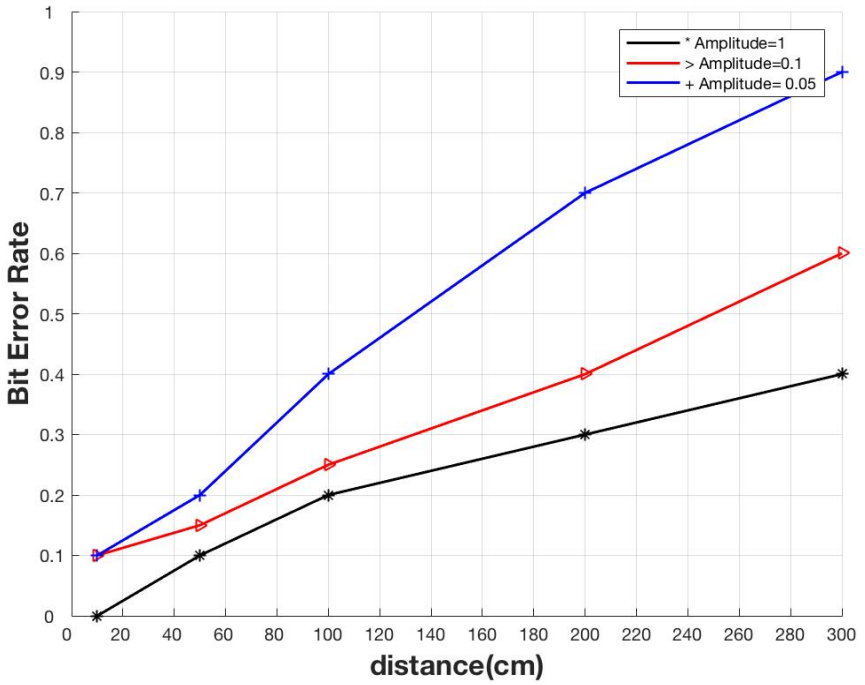

Figure 4. Results of signal transmission and reception in Room A.

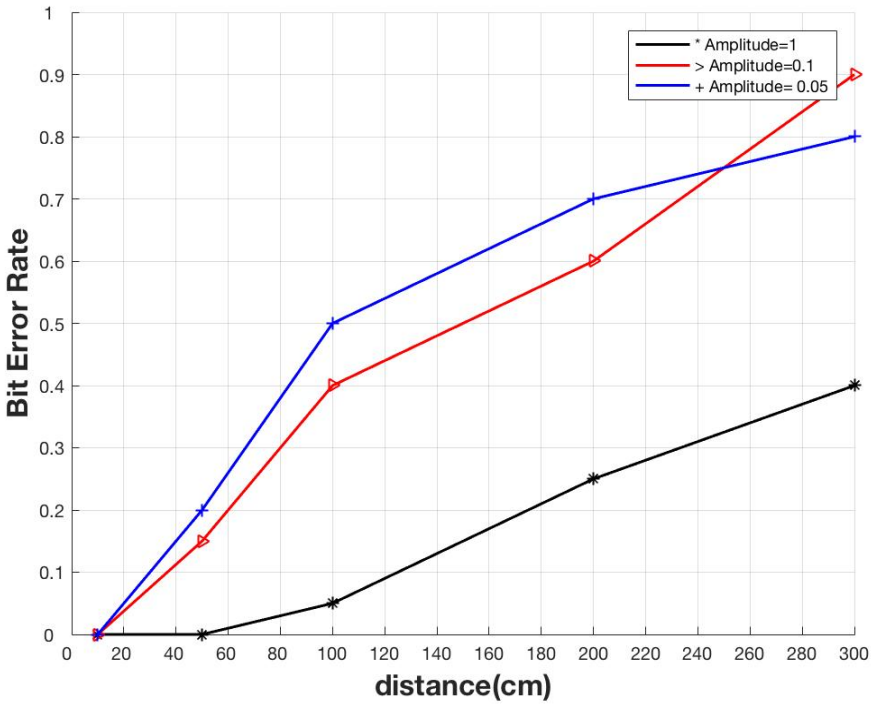

Figure 5. Results of signal transmission and reception in Room B.

4.3. Two-way reception measurement of the authenticator and user being authenticated

Because the method proposed in this study is two-way verification, both the authenticator and user being authenticated must perform a challenge-response authentication. Therefore, this experiment was conducted with two devices, D1 and D2, and a random signal was transmitted each time. Different noises exist in different physical states, and errors may occur even if the distance is only 0.5 m. Therefore, the ambient authentication sets a special threshold value through the proximity analysis model to prevent false negatives. Performance of the microphones and speakers of D1 and D2 may vary, and this may result in different error rates. This experiment investigated the false negatives in the measurement results of different devices in the same environment and the BERs for the same distances between D1 and D2 in different environments. Different environments produced disparate error rates; therefore, the receiving environment of a user being authenticated cannot be simulated even if the attacker obtains the original transmitted signals. This study divided the experiment into two parts for analysis: (1) D1 and D2 at different distances in the same environment and (2) D1 and D2 at the same distance in different environments.

4.3.1. Experimental results of D1 and D2 at different distances in the same environment

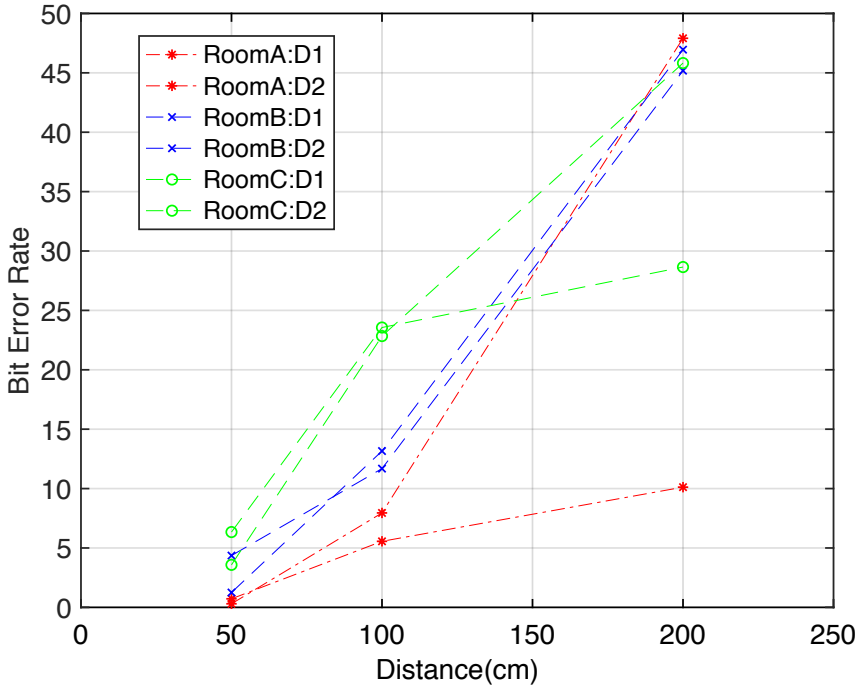

Figure 6. Test result of D1 and D2 at different distances and difference environments.

Figures 6 shows the transmission and reception results of D1 and D2 in different Room A, Room B, and Room C, respectively. Table 5 shows the experimental results of 50 randomly transmitted symbol sequences. At the distances of 0.5 and 1 m, the SERs of D1 were 0.3, 1.25, and 3.58 and 7.95, 13.16, and 22.84, respectively. At 0.5 and 1 m, the SERs of D2 were 0.72, 4.36, and 6.34 and 7.56, 11.68, and 23.56, respectively. This experiment demonstrated that the SER was 0.2 at the transmission range of

0.5m. The differences of SERs at 0.5 and 1m were higher compared with the fixed transmitted symbol sequences of the two previous experiments.

Table 5. Experimental results of D1 and D2 at different distances in the same environment.

Devices	*D*1			*D*2		
Distance (m)	0.5m	1m	2m	0.5m	1m	2m
Symbol error rate of Room A	0.3%	7.95%	0.72%	0.72%	5.57%	10.12%
Symbol error rate of Room B	1.25%	13.16%	4.36%	4.36 %	11.68%	45.18%
Symbol error rate of Room C	3.58%	22.84%	45.82%	6.34%	23.56%	28.62%

Table 6 shows the false negative rate (FNR) and false acceptance rate (FAR) of D1 and D2. In Room A, the SER was 0.02 at 0.5 m, which is the threshold value selected in this study. When the SER was greater than 0.02, the system refused the login if the authenticator was $\geq 0.5m$ from the user being authenticated. The experimental results of D1 and D2 provided false negative rates of 0.02 and 0.07, respectively, and the FARs were 0 and 0.12.

Table 6. False negative rates and false acceptance rates of D1 and D2.

Room	*D*1 *FNR*	*D*1 *FAR*	*D*2 *FNR*	*D*2 *FAR*	*Threshold value*
Room A	0.02	0	0.07	0.12	2
Room B	0	0.22	0.17	0.2	5
Room C	0.01	0	0.12	0.18	8

In Room B, the SER was 0.05 at 0.5 m, which was selected as the threshold value in this study. When the SER was greater than 0.05, the system refused the login if the authenticator was greater than 0.5 m from the user being authenticated. D1 and D2 exhibited false negative rates of 0 and 0.17, respectively, and FARs of 0.22 and 0.2. In Room C, the SER was 0.08 within 0.5 m, which was selected as the threshold value in this study. When the SER was greater than 0.08, the system refused the login if the authenticator was greater than 0.5 m from the user being authenticated. D1 and D2 resulted in false negative rates of 0.01 and 0.12, respectively, and FARs of 0 and 0.18. The findings showed that the BER was different for D1 and D2 in the same environment. Therefore, this study calculated the false negative rates of D1 and D2 in dissimilar environments. Room C is a relatively noisy environment, where the results showed that the false negative rate of D1 was satisfactory, but the false negative rate of D2 was slightly higher (within 0.20). Overall, the method proposed in this study was highly feasible and showed a low false negative rate.

4.3.2. Experimental results of D1 and D2 at the same distance in different environments

Figures 7 and 8 present the BERs of D1 and D2, respectively, at same distance in three different environments. Values on the graph are the average values of BERs after 100 tests for D1 and D2 in each environment at each distance. The experimental results demonstrated that the BERs of D1 and D2 varied in different environments for the same distance. This was caused by the different noises of

the unique environments. This finding confirms that a user's environment cannot be fully simulated, even if the attacker taps into different environments.

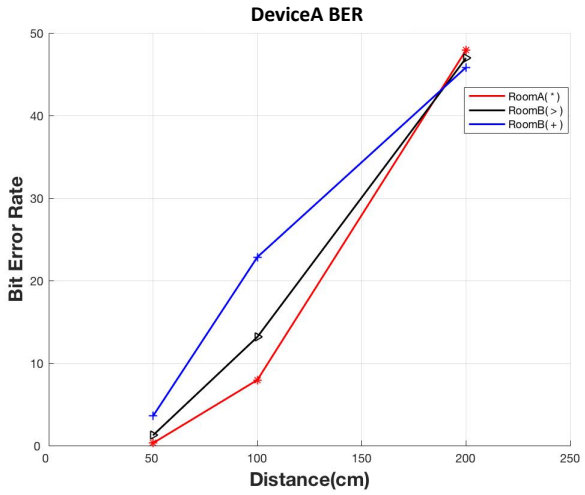

Figure 7. Receiving conditions of D1 in three different environments.

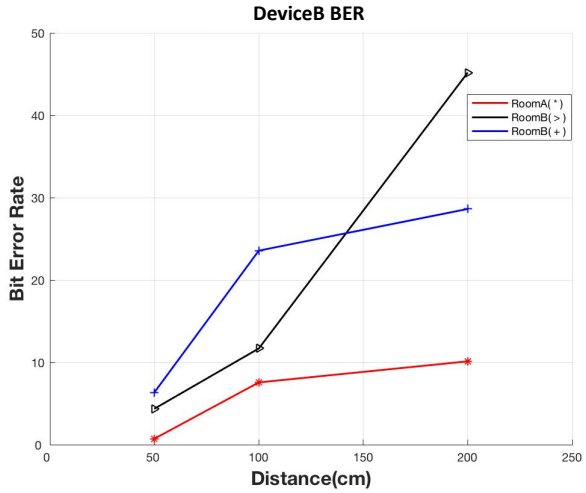

Figure 8. Receiving conditions of D2 in three different environments.

4.4. Comparison of authentication range

In this section, we compare the authentication range of varios methods, as shown in Table 7. The experimental authentication range proposed by Choi et al. [21] was 2 m. In Varshavsky et al. [27]'s method, it was unable to defend against attackers with more than 3 m; Our methods used the distance of 0.5 m, a shorter transmission distance and authentication range than used in related studies, as the threshold value, which improved the security of the authentication. Table 7 compares the FAR of the methods proposed in relevant studies.

Table 7. Comparison of authentication range of various methods.

Methods	*Authentication Range*
This study	0.5 m
Schurmann et al. [29]	Unable to distinguish
Karapanos et al. [35]	Unable to distinguish
Choi et al. [21]	2 m
Varshavsky et al. [27]	3 m
Hien et al. [31]	Unable to distinguish

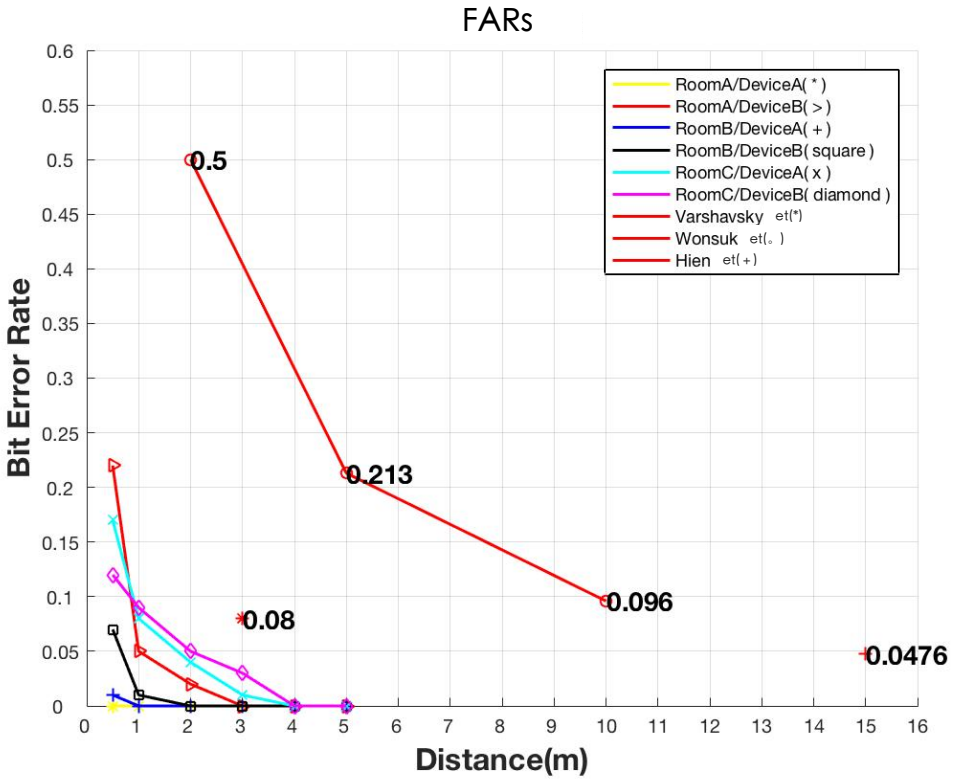

Figure 9. Comparison of false acceptance rates.

Figure 9 shows the false acceptance rate of our method and the others. The method proposed by Schurmann et al. [29] performs authentication within the same environment and establishes similar environmental modes for various environments. The median of the bit similarity was only 0.05 in their experiments, indicating that attackers could simulate the environment of the user being authenticated for a relay attack. The method proposed by Karapanos et al. [35] achieved an FAR of 1 when the users were within the same city sharing the same Internet and TV service providers. In Choi et al. [21]'s method, the FARs are 0.5%, 0.213%, and 0.096% when the attacker was located at distances of 2m, 5m, and 10m, respectively. In Hien et al. [31] 's method, the FARs of various sensor models (WiFi, Bluetooth, GPS, and audio) reached 4.76% when the time cost was 10s; the FARs increased to 7.14% when the time cost was shortened to 5 s. The method proposed by Varshavsky et al. [27] exhibited an FAR of 0.08 when the attacker was located within 3 m of the authenticator. The method proposed by this study committed no false acceptance when the range was more than 3 m; this is effective for narrowing the authentication range to improve security relative to related studies .

5. Conclusion

This study proposed a multi-factor authentication mechanism based on ambient audio. The authentication server uses an audio channel that is not easily detectable by the human ear to establish a secure channel for authentication. In the past, ambient audio authentication was unable to defend against attackers from within the same environment. In addition, previous ambient audio authentication methods were unable to identify attackers with the same sound source even from different environments. The method we propose can be used to authenticate whether a device is in the same environment as another device. Even if there is a malicious attacker in the environment near the device we want to authenticate, and the attacker can obtain the ambient features from the server, the method we propose can also distinguish between users and attackers. Results shown the method proposed in this study can detect an attacker from within 0.5 m, which is a low attack range and improves security over previous methods.

Acknowledgement

This research was supported by the National Science Council of Taiwan under grant no. MOST 108-2218-E-011- 021-, MOST 107-2221-E-033-010-, and MOST 107-2221- E-130-001-.

Conflict of interest

The authors declare no conflict of interest.

References

1. S. Babar, A. Stango, N. Prasad, et al., Proposed embedded security framework for internet of things (IoT), in 2011 2^{nd} International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology (Wireless VITAE) , *IEEE*, (2011), 1–5.

2. C. M. Chen, B. Xiang, Y. Liu, et al., A secure authentication protocol for internet of vehicles, *IEEE Access*, **7** (2019), 12047–12057.
3. J. C. W. Lin, J. M. T. Wu, P. Fournier-Viger, et al., A sanitization approach to secure shared data in an IoT environment, *Multimed. Tools Appl.*, **75** (2016), 14075–14087.
4. K. H. Wang, C. M. Chen, W. Fang, et al., On the security of a new ultra-lightweight authentication protocol in IoT environment for RFID tags, *J. Supercomput.*, **74** (2018), 65–70.
5. H. Xiong and Z. Qin, Revocable and scalable certificateless remote authentication protocol with anonymity for wireless body area networks, *IEEE T. Inf. Foren. Sec.*, **10** (2015), 1442–1455.
6. T. Y. Wu, C. M. Chen, K. H. Wang, et al., A provably secure certificateless public key encryption with keyword search, *J. Chin. Inst. Eng.*, **42** (2019), 20–28.
7. H. Xiong, Y. Zhao, L. Peng, et al., Partially policy-hidden attribute-based broadcast encryption with secure delegation in edge computing, *Future Gener. Comp. Sy.*, **97** (2019), 453–461.
8. J. N. Luo and M. H. Yang, A mobile authentication system resists to shoulder-surfing attacks, *Multimed. Tools Appl.*, **75** (2016), 14075–14087.
9. J. N. Luo, M. H. Yang and C. L. Tsai, An anti-shoulder-surfing authentication scheme of mobile device, *J. Internet Technol.*, **19** (2018), 1263–1272.
10. K. H. Yeh, C. Su, W. Chiu, et al., I walk, therefore I am: continuous user authentication with plantar biometrics, *IEEE Commun. Mag.*, **56** (2018), 150–157.
11. L. Zhou, C. Su, W. Chiua, et al., You think, therefore you are: transparent authentication system with brainwave-oriented bio-features for IoT networks, *IEEE T. Emerg. Top. Com.*, (2017).
12. M. Gao, X. Hu, B. Cao, et al., Fingerprint sensors in mobile devices, in 2014 9^{th} IEEE Conference on Industrial Electronics and Applications, *IEEE*, (2014), 1437–1440.
13. A. Roy, N. Memon and A. Ross, Masterprint: Exploring the vulnerability of partial fingerprint-based authentication systems, *IEEE T. Inf. Foren. Sec.*, **12** (2017), 2013–2025.
14. A. Bud, Facing the future: The impact of apple Face ID, *Biometric Technol. Today*, **2018** (2018), 5–7.
15. GSMA.com, SS7 vulnerabilities and attack exposure report 2018, 2018. Available from: `https://www.gsma.com/membership/ss7-vulnerabilities-and-attack-exposure-report-2018/`.
16. Google Inc., Google authenticator open source, 2018. Available from: `https://github.com/google/google-authenticator`.
17. FIDO Alliance, FIDO (Fast IDentity Online) Alliance, 2018. Available from: `https://fidoalliance.org`.
18. FIDO Alliance, FIDO speficication 1.0, 2014. Available from: `https://fidoalliance.org/fido-1-0-specifications-published-and-final/`.
19. FIDO Alliance, FIDO UAF architectural overview, 2017. Available from: `https://fidoalliance.org/specs/fido-uaf-v1.1-ps-20170202/fido-uaf-protocol-v1.1-ps-20170202.html`.

20. FIDO Alliance, FIDO Universal 2nd factor (U2F) overview, 2017. Available from: https://fidoalliance.org/specs/fido-u2f-v1.2-ps-20170411/fido-u2f-overview-v1.2-ps-20170411.html.
21. W. Choi, M. Seo and D. H. Lee, Sound-proximity: 2-factor authentication against relay attack on passive keyless entry and start system, *J. Adv. Transport.*, (2018), 1–13.
22. J. Krumm and K. Hinckley, The nearme wireless proximity server, in *International Conference on Ubiquitous Computing*, Springer, (2004), 283–300.
23. A. Levi, E. Çetintaş, M. Aydos, et al., Relay attacks on bluetooth authentication and solutions, in *International Symposium on Computer and Information Sciences*, Springer, (2004), 278–288.
24. A. Francillon, B. Danev and S. Capkun, Relay attacks on passive keyless entry and start systems in modern cars, in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, Internet Society, (2011).
25. L. Francis, G. Hancke, K. Mayes, et al., Practical NFC peer-to-peer relay attack using mobile phones, in *International Workshop on Radio Frequency Identification: Security and Privacy Issues*, Springer, (2010), 35–49.
26. B. Shrestha, M. Shirvanian, P. Shrestha, et al., The sounds of the phones: Dangers of zero-effort second factor login based on ambient audio, in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ACM, (2016), 908–919.
27. A. Varshavsky, A. Scannell, A. LaMarca, et al., Amigo: Proximity-based authentication of mobile devices, in *International Conference on Ubiquitous Computing*, Springer, (2007), 253–270.
28. Wireless Cables Inc., Aircable, 2019. Available from: https://www.aircable.net/extend.php.
29. D. Schürmann and S. Sigg, Secure communication based on ambient audio, *IEEE T. Mobile Comput.*, **12** (2013), 358–370.
30. M. Miettinen, N. Asokan, T. D. Nguyen, et al., Context-based zero-interaction pairing and key evolution for advanced personal devices, in Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, ACM, (2014), 880–891.
31. B. Shrestha, N. Saxena, H. T. T. Truong, et al., Drone to the rescue: Relay-resilient authentication using ambient multi-sensing, in International Conference on Financial Cryptography and Data Security, Springer, (2014), 349–364.
32. M. Shirvanian, S. Jarecki, N. Saxena, et al., Two-factor authentication resilient to server compromise using mix-bandwidth devices, in Proceedings of the Network and Distributed System Security Symposium (NDSS), Internet Society, (2014).
33. T. K. Hon, L. Wang, J. D. Reiss, et al., Audio fingerprinting for multi-device self-localization, *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, **23** (2015), 1623–1636.
34. N. Nguyen, S. Sigg, A. Huynh, et al., Using ambient audio in secure mobile phone communication, in 2012 IEEE International Conference on Pervasive Computing and Communications Workshops, *IEEE*, (2012), 431–434.

35. N. Karapanos, C. Marforio, C. Soriente, et al., Sound-proof: usable two-factor authentication based on ambient sound, in 24th *USENIX Security Symposium (USENIX Security 15)*, (2015), 483–498.

36. D. Arp, E. Quiring, C. Wressnegger, et al., Privacy threats through ultrasonic side channels on mobile devices, *2017 IEEE European Symposium on Security and Privacy (EuroS&P)*, (2017), 35–47.

37. L. Blue, H. Abdullah, L. Vargas, et al., 2MA - Verifying Voice Commands via Two Microphone Authentication., in *AsiaCCS 2018*, (2018), 89–100.

38. M. Wang, W. T. Zhu, S. Yan, et al., SoundAuth: Secure Zero-Effort Two-Factor Authentication Based on Audio Signals, in 2018 IEEE Conference on Communications and Network Security (CNS), *IEEE*, (2018), 1–9.

39. L. Deshotels, Inaudible sound as a covert channel in mobile devices, in 8th *USENIX Workshop on Offensive Technologies (WOOT 14)*, (2014).

AIMS Press