This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

InaudibleKey2.0: Deep Learning-Empowered Mobile Device Pairing Protocol Based on Inaudible Acoustic Signals

Huanqi Yang, *Graduate Student Member, IEEE*, Zhenjiang Li, *Member, IEEE*, Chengwen Luo, Bo Wei, and Weitao Xu, *Senior Member, IEEE*

***Abstract*— The increasing proliferation of Internet-of-Things (IoT) devices in daily life has rendered secure Device-to-Device (D2D) communication increasingly crucial. Achieving secure D2D communication necessitates key agreement between various IoT devices without prior knowledge. Despite existing literature proposing numerous approaches, they exhibit limitations such as low key generation rates and short pairing distances. In this paper, we present InaudibleKey2.0, an inaudible acoustic signal based key generation protocol for mobile devices. Based on acoustic channel reciprocity, InaudibleKey2.0 exploits the acoustic channel frequency response of two legitimate devices as a shared secret for key generation. To significantly enhance performance, InaudibleKey2.0 incorporates novel technologies, including a deep learning-enabled channel prediction model for improved channel reciprocity, a quantization model for increased key generation rates, and a transformer-based reconciliation method for augmented key agreement rates. We conduct comprehensive experiments to evaluate InaudibleKey2.0 in diverse real-world environments. In comparison to state-of-the-art solutions, InaudibleKey2.0 achieves 1.3–9.1 times improvement in key generation rates, 3.2–44 times extension in pairing distances, and 1.2–16 times reduction in information reconciliation counts. Security analysis substantiates that InaudibleKey2.0 is resilient to numerous malicious attacks. Furthermore, we implement InaudibleKey2.0 on modern smartphones and resource-limited IoT devices. The results indicate that it is energy-efficient and can operate on both powerful and resource-limited IoT devices without causing excessive resource consumption.**

***Index Terms*— Key generation, mobile devices, acoustic signal, device pairing, deep learning.**

Manuscript received 2 August 2023; revised 10 April 2024; accepted 26 May 2024; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor C. Peng. This work was supported in part by NSFC under Project 62101471; in part by Shenzhen Research Institute, City University of Hong Kong; in part by the Natural Science Foundation of Guangdong Province under Project 2024A1515010192 and Project 2414050001974; in part by the Research Grants Council of Hong Kong Special Administrative Region, China, under Project CityU 21201420 and Project CityU 11201422; and in part by NSF of Shandong Province under Project ZR2021LZH010. *(Corresponding author: Weitao Xu.)*

Huanqi Yang, Zhenjiang Li, and Weitao Xu are with the City University of Hong Kong Shenzhen Research Institute, Shenzhen 518057, China, and also with the Department of Computer Science, City University of Hong Kong, Hong Kong (e-mail: huanqi.yang@my.cityu.edu.hk; zhenjiang.li@cityu.edu.hk; weitaoxu@cityu.edu.hk).

Chengwen Luo is with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: chengwen@szu.edu.cn).

Bo Wei is with the School of Computing, Newcastle University, NE1 7RU Newcastle upon Tyne, U.K. (e-mail: bo.wei@newcastle.ac.uk).

Digital Object Identifier 10.1109/TNET.2024.3407783

I. INTRODUCTION

A. Background

AS MOBILE computing and embedded technology continue to advance, the prevalence of Internet-of-Things (IoT) devices, such as smartphones, smartwatches, and voice assistants, has grown significantly in our daily lives. Consequently, the need to pair devices for tasks like data sharing, synchronization, and collaboration has become increasingly common. For instance, two individuals meeting for the first time may wish to temporarily connect their smartphones to exchange digital business cards. Given the inherently open nature of wireless communication, cryptographic key agreement is a fundamental prerequisite for securing Device-to-Device (D2D) communication and ensuring confidentiality [1], [2].

Secure key distribution between two communicating parties can be addressed using public key infrastructure (PKI). However, public key-based solutions are often inapplicable to mobile devices. This is because PKI is only effective when the other party's identity is known out-of-band or when trusted parties have identities signed by pre-established certificate authorities. An alternative solution is pre-distributed keys, typically in the form of master keys or key materials. However, key pre-distribution schemes lack scalability, rendering them unsuitable for dynamic environments where devices frequently join and leave. Near field communication (NFC) has gained popularity in modern mobile devices, but its communication range is limited to a few tens of centimeters (typically less than 20 cm). The Diffie-Hellman protocol (D-H protocol) is a widely used key establishment protocol for generating cryptographic keys over a public channel. However, the D-H protocol is vulnerable to man-in-the-middle (MITM) attacks, and authenticated D-H protocols necessitate the presence of a certificate authority (CA). Currently, the most prevalent method for pairing mobile devices involves users scanning for nearby devices, selecting the desired device, and manually confirming the connection. This method is neither convenient for users nor appropriate for devices lacking displays.

B. Motivation

The absence of efficient and user-friendly pairing methods has motivated researchers to investigate suitable alternatives for authenticating mobile devices. A widely adopted design

1558-2566 © 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Technical University of Denmark DTU Library. Downloaded on July 01,2024 at 07:58:30 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

principle in literature assumes that devices do not possess any shared knowledge prior. In cases where multiple devices have a comparable perception of a specific random signal, this signal can be employed for key extraction. Diverse designs investigate various types of signals [3], [4], [5], [6], [7], [8], [9], [10], [11], all aiming to achieve a key generation system design that is *fast* (with a sufficient bit generation rate), *practical* (not requiring additional hardware), and *ubiquitous* (usable in diverse environments).

A pioneering set of efforts has successfully leveraged wireless channel information [1], [3], [4], [5], [6], [12], including metrics like Received Signal Strength Indicator (RSSI) and Channel State Information (CSI). The foundation of these approaches lies in channel reciprocity, which means that the channel attributes (RSSI or CSI) determined between two devices by quickly swapping a pair of probe packets will be nearly the same. Nevertheless, RSSI-based techniques are prone to low key generation rates and predictable channel attacks [3], [5]. On the other hand, CSI-based systems have the potential to enhance key generation rates significantly and display higher resistance to predictable channel attacks [4]. The main drawback of CSI-based systems is the need for specialized toolkits to obtain channel information from wireless cards since currently only a limited number of chipsets, such as Intel's 5300 NIC, support CSI extraction [13], [14]. Common short-range wireless communication technologies, such as RFID and ZigBee, are unsuitable due to their absence in most commercial mobile devices.

To address this constraint, some methods have been developed for mobile devices by leveraging built-in sensor data, such as acoustic information, motion sensor readings, and biosensor measurements. Nonetheless, our analysis demonstrates that these designs often sacrifice sensor availability for other limitations — either the system functions within a very limited pairing range, like 1.25 cm in Proximate [8] and 5 cm in TDS [4], or in a restricted set of predetermined contexts, for example, specific environments or during particular user activities [10], [11]. The limited pairing distance substantially narrows the utility of such systems. For example, they might not be appropriate in scenarios where individuals must keep social distance (typically greater than 1.5 m) during a pandemic or in the aftermath. Many other methods may also suffer from extended authentication delays [15], demand costly software support (e.g., public key) [16], or call for supplementary hardware (e.g., bio-sensors) [17], [18]. In this paper, we aim to investigate the possibility of designing a fast and ubiquitous key agreement system capable of pairing two mobile devices beyond social distance without relying on additional sensors.

C. Our Approach and Design Challenges

We find that acoustic signals hold considerable promise, drawing motivation from the accomplishments of earlier radio signal-based approaches. Acoustic waves, as a type of wave, exhibit numerous similarities with radio waves. Specifically, our goal is to leverage *acoustic channel reciprocity* for key generation. The advantages of utilizing acoustic signals are twofold. Firstly, the ubiquity of microphones and speakers in mobile devices obviates the need for any specialized hardware or software modifications. Secondly, the capacity of acoustic signals to transmit over several meters enables a more extended device pairing range. Our preliminary study confirms that the acoustic channel indeed displays channel reciprocity, as well as temporal variation and spatial decorrelation, which could form the basis for key generation. However, due to the restricted acoustic channel bandwidth and the positional discrepancy between speakers and microphones, several obstacles need to be overcome in order to devise an efficient and robust key agreement protocol based on acoustic signals.

1) The first challenge lies in achieving a high bit generation rate through narrowband acoustic channels. To accomplish this, we need to extract fine-grained acoustic channel information. Regrettably, unlike wireless cards, microphones are not designed to provide channel information such as RSSI and CSI. In order to extract fine-grained channel information, we design an effective transmitting scheme that employs inaudible acoustic signals to modulate Orthogonal Frequency-Division Multiplexing (OFDM) symbols. While the application of OFDM modulation in acoustic signals has been proposed in FingerIO [19], using OFDM in a key generation system can provide more acoustic channel information for key generation. Consequently, our evaluation demonstrates a significant improvement in the key generation rate.
2) The second challenge arises from the fact that the microphone and speaker are not located at the same position in mobile devices. As a result, the transmitted signal and received signal will experience slightly different channels. Furthermore, due to hardware diversity and manufacturing imperfections [20], different microphones and speakers may selectively attenuate certain frequencies, leading to additional errors. To address this challenge, we first propose a novel deep learning-empowered acoustic channel prediction[1] and quantization[2] model, which can improve channel reciprocity and achieve high data rate quantization simultaneously. Subsequently, we propose a transformer-based reconciliation model to correct the mismatches. Evaluation results demonstrate that InaudibleKey2.0 can achieve a high matching rate even for various types of IoT devices.
3) The third challenge involves enhancing the entropy of the extracted key. Conventional quantization methods typically employ a threshold for binary encoding [3]. However, such methods can generate recurring bit strings, which decrease the entropy of the produced keys. Moreover, using these keys directly for generating the final key enables powerful adversaries to acquire raw information through reverse engineering. To tackle this issue, we initially implement a novel Bloom filter-based approach to safeguard the generated keys against reverse engineering attacks. Subsequently, we utilize the Karhunen-Loeve Transform (KLT) to eradicate redundant information and improve randomness.

[1] In this paper, the channel prediction refers to predicting the channel frequency response of an acoustic channel.

[2] Quantization refers to the process of converting channel measurements (i.e., channel frequency response in this paper) into binary bits, namely 0s and 1s.

Authorized licensed use limited to: Technical University of Denmark DTU Library. Downloaded on July 01,2024 at 07:58:30 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

While numerous recent studies have utilized acoustic signals for pairing mobile devices [15], [16], [21], [22], [23], InaudibleKey2.0 exhibits a substantial enhancement in performance.[3] This paper presents the following contributions:

- **System Design.** We present InaudibleKey2.0, an inaudible acoustic signal-based key agreement protocol for mobile devices. Based on acoustic channel reciprocity, InaudibleKey2.0 uses the channel frequency response of OFDM symbols for key generation. InaudibleKey2.0 incorporates several innovative techniques to markedly enhance system performance, such as a deep learning-empowered channel prediction and quantization model, and a transformer-based reconciliation method.
- **System Implementation.** To demonstrate feasibility, we implement a prototype of InaudibleKey2.0 on both powerful devices (smartphones) and resource-limited devices (Raspberry Pi 4). Evaluation results indicate that InaudibleKey2.0 incurs low system costs and runs efficiently on these IoT devices. Additionally, we show that it is more energy-efficient than public key cryptography and authenticated D-H protocol on IoT devices.
- **System Evaluation.** We conduct comprehensive experiments in various real-world settings. In comparison to state-of-the-art works, InaudibleKey2.0 enhances key generation rates by 1.3–9.1 times, increases pairing distance by 3.2–44 times, and lowers the number of information reconciliation counts by 1.2–16 times.
- **Security Analysis.** Thorough analysis demonstrates that InaudibleKey2.0 exhibits resilience against several malicious attacks, including eavesdropping, imitation, and predictable channel attacks.

The remainder of this paper is structured as follows: Section II presents the system model, followed by a detailed explanation of the system design in Section III. We evaluate the system performance in Section IV and assess its security in Section V. Related works are discussed in Section VI, and we conclude the paper in Section VII.

II. System Model

Fig. 1 illustrates the system model of InaudibleKey2.0. We assume that two mobile devices, Alice and Bob, intend to agree on the same secret key for secure communication. Both devices are equipped with a speaker and a microphone and have InaudibleKey2.0 installed, but share no prior secrets. We assume an adversarial device, Eve, is positioned beyond a secure distance (10 cm in InaudibleKey2.0) from the legitimate devices. If Eve approaches within a safe distance, users can easily detect it, as mentioned in prior research [16]. Although Eve is even closer to Alice or Bob, the acoustic channel reciprocity guarantees that her channel measurements are totally different from that of the channel between Alice and Bob. A possible use case can be described as such: Imagine at a conference, Alice and Bob, meeting each other for the first time, wish to securely exchange their business cards. By using InaudibleKey2.0, they can merely shake their device (e.g., smartwatch) or execute a random gesture close to the device for a brief period. This guarantees that secure communication is set up between the devices, even when they are separated by a distance of 1–2 m. If there is enough randomness, such as moving individuals or objects surrounding the users, they may not even need to perform any actions (see our demo[4]).

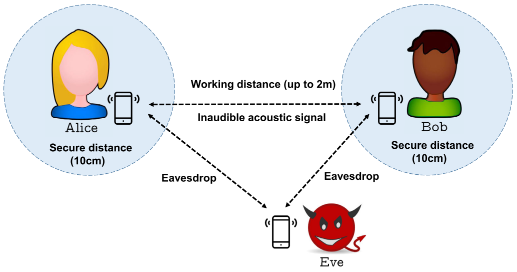


Fig. 1. System model.

We assume that Eve has complete knowledge of the key agreement protocol and can intercept, inject, and replay messages. However, as in numerous prior key generation research studies [1], [4], [10], [16], [21], although Eve can introduce messages into the public wireless channel, we assume Eve's goal is to acquire the secret key rather than disrupting their communications (i.e., Denial-of-Service (DoS) attack). In practice, a DoS attack against InaudibleKey2.0 could be carried out by obstructing inaudible acoustic signals in the environment, but such an attack can be identified using existing techniques in the literature [24]. Meanwhile, it should be noted that InaudibleKey2.0 is a key generation system rather than an authentication system, which means Eve can also initiate the pairing process and generate a key with Alice (say $K_{Alice-Eve}$) or Bob ($K_{Bob-Eve}$). But our system ensures $K_{Alice-Bob} \neq K_{Alice-Eve} \neq K_{Bob-Eve}$ because of the acoustic channel reciprocity. In this paper, we take into account three types of attacks frequently utilized in related works [1], [21]:

- Eavesdropping attack: Eve eavesdrops on all the messages transmitted in the public channel with the aim of extracting the same key.
- Imitating attack: Once Alice or Bob completes key extraction, Eve approaches the same site intending to generate an identical key as a legitimate user. For instance, Eve can initially observe how Alice or Bob uses their devices, such as their method of moving or shaking smartphones, then attempt to imitate their usage patterns and generate the same key.
- Predictable channel attack: Eve might intentionally change her position to create expected or predictable alterations in the communication channel between Alice and Bob.

III. System Design

Figure 2 illustrates InaudibleKey2.0's workflow. First, the two legitimate devices, Alice and Bob, exchange several

[3] InaudibleKey2.0 is an extended version of InaudibleKey [23]. In this manuscript, Sec. III-B.1 and Sec. III-C are new content. Additionally, all experiments have been re-performed with the new methods.

[4] https://www.youtube.com/watch?v=V8JSgOhairM [Online, accessed on July 19, 2023]

Authorized licensed use limited to: Technical University of Denmark DTU Library. Downloaded on July 01,2024 at 07:58:30 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

inaudible acoustic frames and compute the channel frequency response (CFR). Next, these two devices adhere to the pipelines outlined in Figure 2 to produce an identical cryptographic key. Ultimately, the generated secret key can be used to protect communication between them. The design details are as follows.

A. Transmitting Signal Design

During this phase, Alice and Bob exchange several acoustic frames by transmitting through their speakers and receiving them via their microphones to acquire channel measurements. InaudibleKey2.0 utilizes an inaudible audio frequency band ranging from 18 kHz to 22 kHz because most people cannot hear frequencies in this range.

To obtain fine-grained channel information, we utilize OFDM technology with acoustic signals based on the method employed in FingerIO [19]. We use the same parameter setting as FingerIO because it has been demonstrated to achieve high performance. Specifically, we partition the 18-22 kHz frequency range into 64 subcarriers, resulting in a width of 62.5 Hz for each subcarrier. The time-domain samples for transmission can be acquired by conducting an inverse Fast Fourier Transform (IFFT) on the transmitted data, while the receiver reconstructs the raw data bits using a Fast Fourier Transform (FFT). A speaker transmits vectors consisting of 64 real values derived from OFDM symbol creation. Another benefit of employing OFDM technology is that both devices can probe the channel within the channel coherence time without explicitly synchronizing the two mobile devices. In practice, it is unreasonable to assume that these two legitimate devices are synchronized upon encountering each other. We use the first S_{suf} of these values to create a cyclic suffix attached to the end of the OFDM symbol. This cyclic suffix assists in precisely estimating the start of the OFDM symbol. Even if Alice and Bob are not synchronized, they can still identify the beginning of the received symbol by calculating the correlation between the received signal and the known transmitting signal (refer to [19] for additional information).

The duration of the transmitted signal and the interval between transmissions are essential for two main reasons: 1) To guarantee that Alice and Bob acquire channel estimates within the coherence time, resulting in highly correlated CFRs; 2) The transmission interval should be longer than the coherence time to avoid consecutive CFRs from being correlated, which would diminish the key's randomness. Theoretically, the channel's change rate change is indicated by the Doppler frequency (f_d), and the channel's stability duration is referred to as channel coherence time (T_c). Coherence time serves as the time-domain equivalent of Doppler spread and helps to characterize the time-varying nature of the channel frequency. Assuming the motion speed of a subject or object is v, the channel frequency is f, and the velocity of the acoustic signal is c ($340\,\mathrm{m/s}$), the maximum Doppler frequency can be determined as $f_d = \frac{v \cdot f}{c}$ [25]. In practice, the channel coherence time in terms of the maximum Doppler frequency shift is $T_c = \sqrt{\frac{9}{16\pi f_d^2}}$ [25].

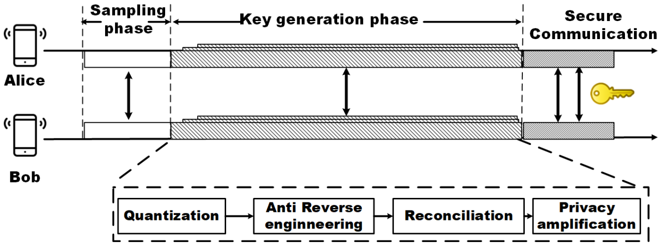


Fig. 2. System flowchart.

In InaudibleKey2.0, the acoustic signal ranges from $18\,\mathrm{kHz}$ to $22\,\mathrm{kHz}$, and the velocity of typical human movements ranges between $0.1\,\mathrm{m/s}$ and $2.7\,\mathrm{m/s}$ [26]. Consequently, the coherence time lies within the interval of $2\,\mathrm{ms}$ to $53\,\mathrm{ms}$. In InaudibleKey2.0, the cyclic suffix length S_{suf} is set to 26, signifying that the transmitted symbol consists of 90 samples. With a sampling rate of $48\,\mathrm{kHz}$, transmitting these 90 samples requires $1.9\,\mathrm{ms}$, which is shorter than the minimum coherence time. Concerning the transmission interval, Alice and Bob exchange acoustic signals every $100\,\mathrm{ms}$, surpassing the maximum coherence time.

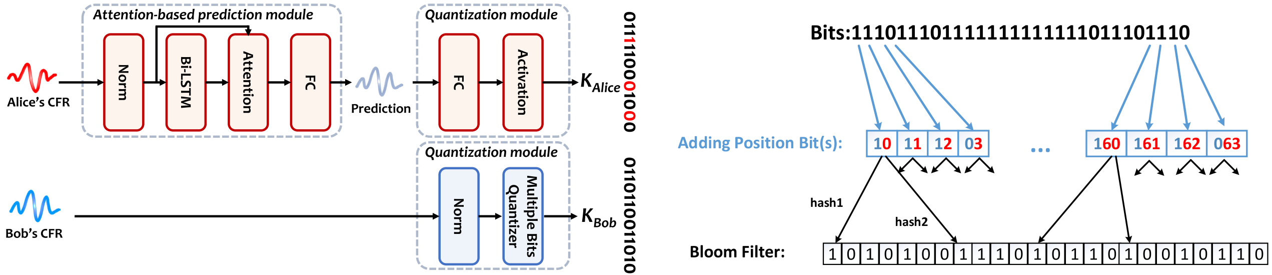
B. Quantization

1) Deep Learning-Empowered Quantization: As aforementioned in Section I-C, although the adoption of OFDM modulation is effective in improving the correlation results of the channel measurements collected by Alice and Bob, the channel conditions of the transmitted and received signals will still render slightly different due to the location difference, hardware diversity, and manufacture imperfections of the microphone and speaker. Consequently, it is desirable to improve the channel reciprocity for better performance. In response to this challenge, a novel deep learning-empowered model is proposed. As shown in Figure 3(a), the proposed model comprises two main components: a prediction module and a quantization module. The prediction module is responsible for predicting the channel measurements that will occur within the channel coherence time, while the quantization module converts the predicted channel measurements into a sequence of bits. The input to the model is a set of channel measurements observed by Alice, and the output is a generated sequence of bits. The motivation for integrating deep learning in the prediction and quantization processes stems from its superior capability to model complex, non-linear relationships. This ability is crucial for adapting to the diverse challenges posed by different locations and hardware imperfections, ensuring robust and accurate key generation. Below we will delve deeper into the details of this proposed model.

Prediction module. As aforementioned, the channel measurements captured by the two legitimate devices are not identical to each other due to the impaired channel reciprocity, which will result in a low matching rate between them. Our solution is to predict Bob's measurement based on Alice's measurement using an attention-based prediction module. The prediction module is composed of an attention-based BiLSTM layer and a fully connected layer. We chose to use BiLSTM because it is well-suited to handling time series data and its bidirectional learning process allows it to capture both the forward and reverse characteristics of the data [27].

Authorized licensed use limited to: Technical University of Denmark DTU Library. Downloaded on July 01,2024 at 07:58:30 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



(a) Attention-based prediction and quantization model

(b) Projecting the bits into Bloom filter space

Fig. 3. Illustration of quantization process.

However, standard BiLSTM may not be capable of capturing the features in a fine-grained manner. To resolve this problem, we attempt to focus on the key information of the input CFR characteristics by incorporating an attention mechanism. The attention mechanism simulates the way the human brain processes information by prioritizing important information and downplaying unimportant information. The proposed model employs an attention-based BiLSTM layer to extract important features from the input channel measurements observed by Alice. Following this, a fully connected layer is added to convert these extracted features into a predicted sequence of channel measurements, which are intended to closely resemble the actual measurements taken by Bob. One key advantage of using a single fully connected layer for this conversion is that the attention-based BiLSTM is capable of learning the overall patterns and dependencies within the input sequence, rather than just local features. Thus, a single fully connected layer is sufficient to capture the most important information from the input data and make accurate predictions. This allows the model to effectively and efficiently use the information present in the input sequence to make predictions.

Quantization module. Once the predictions have been made, the two legitimate devices need to convert the predicted CFRs into binary bit sequences in order to generate the secret keys. This step is traditionally done using a separate module called a quantization module [3]. In InaudibleKey2.0, to avoid the additional cost of using a separate module, the quantization process is integrated into the prediction network by adding a fully connected layer with a sigmoid activation function after the prediction module. The fully connected layer, which connects all the neurons within the layer to those in the previous layers to integrate the representation of the features, is utilized to transform the output generated by the prediction module into a binary vector containing values of either 0 or 1. For Alice, the utilization of the sigmoid function in this fully connected layer is a purposeful choice as it serves as a quantization function, which maps the predicted CFR values outputted by the prediction module, into a range between 0 and 1. The usage of this approach increases the performance of the quantization process because it helps to mitigate the impact of small errors in the predicted CFR values on the resulting binary sequence. On the other hand, Bob uses a multiple-bit quantizer proposed in [3]. This type of quantizer generates a greater amount of bits compared to singular threshold-based techniques. Overall, by integrating the quantization process into the prediction network, the proposed system is able to generate accurate and efficient binary sequences while reducing the overall cost of the system.

Combined training. Combining the prediction and quantization modules into a single neural network makes the training process more convenient and allows for the use of a combined loss function to optimize both modules together. This can yield benefits in terms of computational efficiency and accuracy. The designed combined loss function is represented by

$$\arg\min_{p} |\epsilon \times BCE(\hat{y}, y) + (1-\epsilon) \times MSE(\hat{z}, z)|^2, \quad (1)$$

where p represents the designed prediction and quantization neural network, $MSE(y, \hat{y})$ is the Mean Squared Error loss function for the prediction module, $BCE(z, \hat{z})$ is the Binary Cross Entropy loss function for the quantization module, y and z represent the ground-truth CFRs and binary bits of Bob which serve as the benchmark for evaluating the performance of the model. On the other hand, $\hat{y}$ and $\hat{z}$ represent the predicted CFRs and binary bits, which are the sequences outputted by the model under evaluation, and are being compared against the ground-truth sequences to assess its performance. Furthermore, a hyperparameter ϵ is used to determine the importance of the two loss functions (e.g., prediction loss MSE and the quantization loss BCE), which helps achieve the optimal overall network performance. This allows us to determine the optimal trade-off between the two modules and enhance the overall performance of the model. ϵ is empirically set to 0.1 through experiments.

The loss function employed in the model was specifically designed to serve a dual purpose - prediction and quantization. To ensure the model can accurately output CFRs of Bob, we use MSE loss function as it effectively measures the discrepancy between the predicted sequence and the actual sequence. Meanwhile, quantization is considered as a binary classification problem, where the output of prediction can be either 0 or 1. In such cases, the BCE loss function is a suitable choice as it is widely used for binary classification tasks. Therefore, to optimize the network for both objectives, a combination of the MSE and BCE loss functions is employed, with each function given a specific weighting to balance the contribution of both tasks in the final loss calculation.

The proposed network offers several benefits when compared to prior methods. The prediction module within the neural network architecture improves channel reciprocity by generating predictions of the sequence on the opposite side of the communication channel. Additionally, the quantization module of the network is seamlessly integrated into the overall architecture, eliminating the need for separate modules, and thus increasing the system's efficiency. Another significant advantage is that it only requires execution on a single device,

Authorized licensed use limited to: Technical University of Denmark DTU Library. Downloaded on July 01,2024 at 07:58:30 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

which reduces the computational demands on other devices, making the network more scalable and efficient. This allows for the network to be deployed in a variety of mobile devices, expanding its potential use cases.

2) Anti Reverse Engineering: Numerous previous key generation studies have directly employed quantized bits to derive the final secret key. However, this method is susceptible to attacks from Eve, who can attempt to extract the key from her own gathered data and the data shared between Alice and Bob. The Bloom filter has been used in a variety of privacy-preserving applications as part of encoding and perturbation techniques [28], [29]. However, conventional Bloom filter projections do not maintain order information. As a result, the discrepancies between the input key strings and output key strings might vary. In InaudibleKey2.0, we utilize a specially designed Bloom filter data structure that considers sequence/order information. This assists in projecting the key into the Bloom filter while preserving key distance information in a non-plaintext format. The primary objective of using the adapted Bloom filter is to safeguard this key distance information more securely.

The detailed procedure is depicted in Fig. 3(b). For example, consider a 64-bit key: each individual bit position is communicated using supplementary bit(s) before the Bloom filter hash-mapping. Subsequently, two hash functions are utilized to map each element in the added-position bits to the Bloom filtered space. The Bloom filter hash-mapping solely transforms ‘0’ to ‘1’ based on the hash calculation (refer to [30] for the original rationale). As a result, each ‘1’ in the Bloom filter uniquely corresponds to a specific bit (‘0’ or ‘1’) at a particular position in the initial key. Importantly, this modified Bloom filter data structure can also preserve the Jaccard distance between the raw data bits and the projected Bloom filter data bits. In other words, if $\hat{K}_{Alice}$ and $\hat{K}_{Bob}$ represent the Bloom filter outputs of K_{Alice} and K_{Bob}, and there are N_{mis} mismatches between K_{Alice} and K_{Bob}, then there will also be N_{mis} mismatches between $\hat{K}_{Alice}$ and $\hat{K}_{Bob}$. The proof can be found in [29]. Consequently, we can directly apply the subsequent information reconciliation approach to $\hat{K}_{Alice}$ and $\hat{K}_{Bob}$ since they retain the similarity information between K_{Alice} and K_{Bob}. It is crucial to notice that while the Bloom filter functions as an irreversible one-way process, if the input key's length is too short, Eve might still acquire the Bloom filter's output, such as through a brute-force attack. Thus, it is vital to guarantee the entropy of the Bloom filter input. In InaudibleKey2.0, we concatenate the bits produced from each quantization module into a key string and further split it into consecutive segments, with each segment comprising 128 bits. Considering that Eve is unaware of the number and location of incorrect bits, it becomes computationally unfeasible (2^{128} guesses) to obtain K_{Alice} and K_{Bob}. The modified Bloom filter, which depends solely on hash functions and limits temporary storage (for storing Bloom filter results), adds low overhead to the mobile system.

C. Transformer-Based Reconciliation

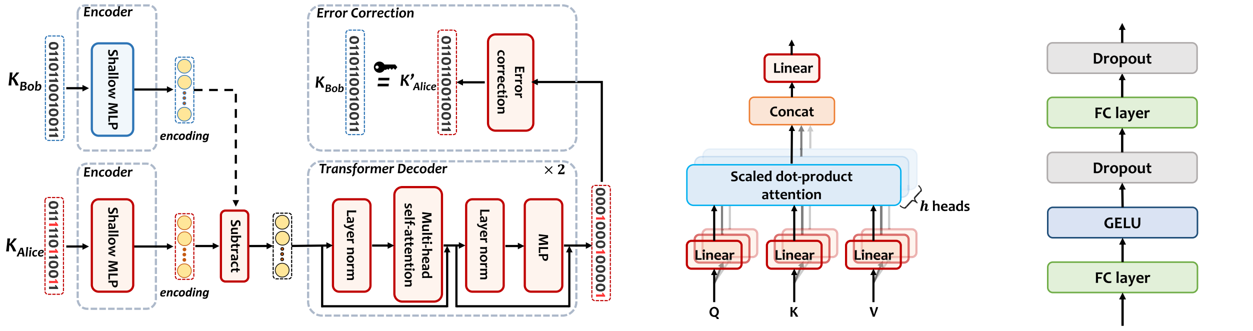
Due to noise, the obtained keys K_{Alice} and K_{Bob} from Alice and Bob, respectively, are not identical but approximately equal, i.e., $K_{Alice} \approx K_{Bob}$. To this end, it is indispensable to use the reconciliation procedure to correct their mismatches. In InaudibleKey2.0, we propose a novel transformer-based reconciliation framework that utilizes deep learning to correct errors between K_{Alice} and K_{Bob}. The design of the transformer-based reconciliation method is motivated by ECCT [31], which employs transformer models to correct the errors in physical communication layers. Unlike ECCT's focus on error correction, our approach extends the utility of transformers to the domain of key reconciliation, targeting the efficient correction of mismatches between binary sequences, K_{Alice} and K_{Bob}. Our approach leverages the powerful capabilities of deep learning to efficiently correct mismatches between the two binary sequences and outperform traditional methods with respect to accuracy.

The proposed transformer-based reconciliation model, as illustrated in Fig.4(a), consists of a two-input transformer that utilizes the keys of Alice and Bob to correct mismatches between the two binary sequences. The proposed method utilizes a lightweight encoder to transform the initial keys of Alice and Bob into encoding representations with lower-dimension. This process is performed separately for each party, as Alice and Bob are distinct D2D communication objects. Once the keys have been encoded, the difference of the lower-dimensional vectors of their keys is used as input for the transformer decoder. The decoder then decodes the key mismatches between the two parties. Bob only needs to perform the encoding process and transmit the resulting encoded vector to Alice. Alice, on the other hand, is responsible for executing the encoding process, subtraction, and decoding process using the transformer decoder to complete the reconciliation process. The steps executed by Alice are represented in the red section of Fig.4(a) while the steps executed by Bob are represented in the blue section. The methodology of the proposed reconciliation technique will be detailed further in the following.

Encoder. Assuming that the two legitimate devices have independently generated the initial keys. These keys are inputs to two distinct encoders, each comprising of a shallow Multi-layer Perceptron (MLP) with two fully connected layers. Specifically, the keys K_{Bob} and K_{Alice} are passed through MLP f_1 and MLP f_2, respectively, to derive the encoding vectors $y_{Bob} = f_1(K_{Bob})$ and $y_{Alice} = f_2(K_{Alice})$, where y_{Alice} and y_{Bob}, respectively, both of which reside in R^M. Here, y_{Alice} and y_{Bob} are high-dimensional condensed representations of their bit sequences. Bob then sends his encoding y_{Bob} to Alice via a public communication channel such as Bluetooth broadcasting. Assuming that upon receipt of the encoding vector y'_{Bob} sent by Alice, it has been corrupted by noise, that is $y'_{Bob} = f_1(K_{Bob}) + e$. Upon receipt of the noisy y'_{Bob}, Alice calculates the difference between Bob's noisy encoding vector and her own: $h = y'_{Bob} - y_{Alice}$. It is worth noting that the vector h can be considered as an approximation of the condensed representation of the mismatches between K_{Alice} and K_{Bob}. On Bob's side, transmitting h instead of K_{Bob} has several advantages. First, the condensed representation is safe to transmit over an unauthenticated channel. Secondly, the length of the condensed vector h is smaller than K_{Bob}, reducing the amount of data transmitted over the communication channel, thereby

Authorized licensed use limited to: Technical University of Denmark DTU Library. Downloaded on July 01,2024 at 07:58:30 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



(a) Model framework. (Bob solely executes the task of the blue section, and then communicates the resulting encoding to Alice.)

(b) Multi-head self-attention

(c) Multi-layer perception

Fig. 4. Transformer-based reconciliation method.

conserving energy and enhancing the overall efficiency of the system.

Decoder. Once Alice has obtained h, she feeds it into a transformer-based decoder g to obtain the mismatches between K_{Alice} and K_{Bob}: $\Delta x = K_{Alice} \oplus K_{Bob}$. The transformer-based decoder is designed as follows. The transformer-based encoder is comprised of alternating layers of multi-head self-attention (MSA) and MLP, in which layer normalization [32] is applied between each layer. This structure allows the encoder to effectively capture and process the condensed encoding representation of the mismatch information between K_{Alice} and K_{Bob}. The multi-head self-attention block is shown in Fig. 4(b). This block allows the decoder to focus on different positions of the input by attending to information from multiple representation subspaces simultaneously. This helps the network learn more robust and fine-grained features of the mismatch information, improving the performance of reconciliation. We apply multi-head attention with 2 heads, where the self-attention function is calculated twice. The self-attention function uses trainable query, key, and value matrices $\boldsymbol{W}^Q \in \mathbb{R}^{D\times D}$, $\boldsymbol{W}^K \in \mathbb{R}^{D\times D}$, and $\boldsymbol{W}^V \in \mathbb{R}^{D\times D}$, respectively, which are first multiplied with the input $\boldsymbol{R}$ to obtain the query, key, and value matrices $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$. Then, the query, key, and value matrices are linearly projected 2 times with different, learned linear projections $\boldsymbol{W}_i^Q$, $\boldsymbol{W}_i^K$, and $\boldsymbol{W}_i^V \in \mathbb{R}^{D\times \frac{D}{h}}$ $(1 \leq i \leq h)$, respectively, to generate linear projected queries, keys, and values. These projections allow the self-attention function to capture different aspects of the input patches in parallel. The projections are concatenated and multiplied by a trainable matrix $\boldsymbol{W}^O \in \mathbb{R}^{D\times D}$ to produce the final output of the self-attention layer:

$$\boldsymbol{Q}_i = \boldsymbol{Q}\boldsymbol{W}_i^Q, \quad \boldsymbol{K}_i = \boldsymbol{K}\boldsymbol{W}_i^K, \quad \boldsymbol{V}_i = \boldsymbol{V}\boldsymbol{W}_i^V. \tag{2}$$

Then, the attention of each head is calculated for each group of $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ using the following equation:

$$\boldsymbol{head}_i = \text{Attention}(\boldsymbol{Q}_i, \boldsymbol{K}_i, \boldsymbol{V}_i) = \text{softmax}\left(\frac{\boldsymbol{Q}_i\boldsymbol{K}_i^\top}{\sqrt{D/h}}\right)\boldsymbol{V}_i. \tag{3}$$

By concatenating all output sequences $\boldsymbol{head}_i$ of each head, the final output of the MSA block can be obtained. After layer normalization, this output is then fed to the MLP block, as shown in Fig. 4(c). The MLP block contains two fully connected layers, two dropout layers, and one GELU (Gaussian error linear unit) layer, which allows the model to learn non-linear relationships between the input and output. The output of the MLP block is then residually connected with the output of the MSA block to acquire the final output of the transformer decoder. This output is fed to the next transformer decoder layer, or it can be used as the final representation of the input if this is the last decoder layer. In the error correction module, as shown in Fig. 4(a), Alice can rectify the mismatches in the keys by simply calculating $K'_{Alice} = K_{\text{Alice}} \oplus \Delta x = K_{Bob}$, where $\oplus$ denotes the bitwise XOR operation and Δx is the mismatch between Alice's and Bob's keys. This allows an agreement between Alice and Bob on the same key, which ensures secure communication.

Training. We design a loss function to optimize the model during the training phase. The loss function is defined as:

$$\underset{f_1, f_2, g}{\arg\min} |\Delta x - (K_{Bob} \oplus K_{Alice})|^2, \tag{4}$$

where Δx denotes the decoded mismatches between K_{Alice} and K_{Bob}, f_1, f_2, and g are the neural network components representing the encoder and the decoder, respectively. The objective of this loss function is to minimize the discrepancy between Δx and $K_{Bob} \oplus K_{Alice}$ by adjusting the parameters of the neural networks.

As discussed in Sec II, Eve possesses the ability to alter, insert, and replay messages. Consequently, she can execute two prevalent attacks during the reconciliation process: MITM and replay attack. Eve can initiate MITM by impersonating Alice or Bob during the key generation process to modify or insert her own messages. To address this issue, we implement the message authentication code (MAC) technique to ensure message integrity and authenticity [5]. Bob adds an extra MAC message with y_{Bob}, making the total message sent to Alice $L_{Bob} = \{y_{Bob}, MAC(K_{Bob}, y_{Bob})\}$. Upon receiving L_{Bob}, Alice calculates K'_{Alice} using Eq. 2 and verifies its identity. If $MAC(K'_{Alice}, y_{Bob}) \neq MAC(K_{Bob}, y_{Bob})$, Alice is aware that the message compromise could be due to modification by an adversary Eve, or a malfunction in the key generation process. In either scenario, Alice can alert the user to the unsuccessful device pairing, prompting the user to undertake appropriate measures, such as retrying the pairing process. If $MAC(K'_{Alice}, y_{Bob}) = MAC(K_{Bob}, y_{Bob})$, Alice can

Authorized licensed use limited to: Technical University of Denmark DTU Library. Downloaded on July 01,2024 at 07:58:30 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

ascertain that the message truly originated from Bob. To identify replay attacks, we can employ standard approaches such as nonces, timestamps, or labeling each message with a session ID [33].

D. Privacy Amplification

Although the quantization method can produce a greater number of bits from each sample, but it may also lead to duplicated bits. Using a key with such duplications could undermine the system's security. Although the Bloom filter-based technique described in Sec III-B.2 offers protection against reverse engineering attacks, it does not improve entropy. To address this limitation, we apply the Karhunen-Loeve Transform (KLT) to decrease the correlation in the bit sequence after reconciliation.

Assume that the key generated by Alice after reconciliation is $\bar{K}_{Alice} = (k_1, k_2, \cdots, k_L)^T$, where k_i represents the i-th bit and L denotes the key's length. The initial step in the KLT process is to compute the auto-correlation matrix $R = E(KK^T)$. Following this, we calculate its eigenvalues λ_i and eigenvectors ϕ_i so that $R\phi = \lambda_i \phi_i \quad (i = 1, 2, \cdots, L)$. It's important to note that R is Hermitian and its eigenvectors ϕ_i are orthogonal. By arranging the eigenvalues in descending order, $\lambda_1 > \lambda_2 > \cdots > \lambda_L$, we can construct a unitary matrix Φ that diagonalizes R: $\Phi = [\phi_1, \cdots, \phi_L]$. This matrix, referred to as the KLT matrix, can be used to decorrelate the bit sequences K. By selecting the largest S eigenvectors to construct Φ, we can obtain a decorrelated key string through $K''_{Alice} = \Phi^T \bar{K}_{Alice}$. Similarly, Bob can generate a decorrelated key sequence $K''_{Bob} = K''_{Alice}$.

While reconciliation enhances the reliability of a key generation protocol, it discloses some information to Eve. Privacy amplification serves as a typical method to eliminate the exposed information from the generated secret key sequence. This is generally achieved through the extractor, universal hashing functions, and cryptographic hash functions [34]. In InaudibleKey2.0, we employ the widely used dual universal hash function [35] to produce the final key. Subsequently, the key can be utilized by encryption algorithms like AES-128 to safeguard their communication.

IV. Evaluation

A. Goals, Metrics and Methodology

Experimental Setup. We implement InaudibleKey2.0 on a Samsung S10 smartphone, equipped with a microphone and speaker. The inaudible acoustic signal's frequency range is set between 18 and 22 kHz, with a sampling rate of 48 kHz. The speaker volume is maximized, resulting in a sound pressure level (SPL) of 82 dB for the acoustic signal. Bluetooth Low Energy broadcast serves as the public channel for exchanging reconciliation data. Most modern mobile devices can support these settings. Our comprehensive experiments involve four smartphones: Alice, Bob, Eve, and David (Eve's partner). In line with advanced studies [4], [21], we gather data in four distinct settings: **A** - indoor stationary, **B** - outdoor stationary, **C** - indoor moving, and **D** - outdoor moving. In the moving scenarios, users either shake Alice and Bob or carry them while walking. In stationary scenarios, Alice and Bob remain still, with individuals moving around them. Indoor experiments are conducted in a student laboratory, while outdoor experiments take place on a campus pathway. To assess the influence of distance, we alter the distance between Alice and Bob from $10\,\mathrm{cm}$ to $300\,\mathrm{cm}$ in each setting. Eve and David are placed at least $10\,\mathrm{cm}$ away from the legitimate devices.

Unless mentioned otherwise, the whole dataset is randomly partitioned into three sections: a training set (70%), a validation set (15%), and a test set (15%). It should be noted that we do not train an individual model for different environments. Instead, the training, validation, and test sets consist of the data from four different environments. We set the initial learning rate at 0.0001 and report results after 200 epochs. The model, which is implemented in PyTorch, is first trained offline on a desktop PC featuring an Intel i7-10700 CPU, 64 GB RAM, and an RTX 3080 GPU. The trained model is then deployed on the smartphone.

Metrics. In this study, we focus on the following four metrics that are widely used in the literatures [10], [36], and [37]. We present the results for each of these metrics as mean values accompanied by their respective standard deviations.

- **Key agreement rate:** it represents the percentage of bits matching in the secret keys generated by two parties. This metric evaluates the potential of Alice and Bob agreeing on the same key.
- **Key generation rate:** it denotes the average number of bits generated per unit time and is usually measured in bits per second (bps). This metric evaluates how fast Alice and Bob can generate a shared secret key.
- **Entropy:** it is the measure of uncertainty or randomness associated with the generated bit strings. The entropy of a binary bit string varies in the range $[0, 1]$, and larger entropy indicates more randomness of the bit string.
- **Reconciliation count:** it refers to the count of reconciliation steps required to achieve 100% key agreement rate.

B. Effectiveness of Channel Prediction

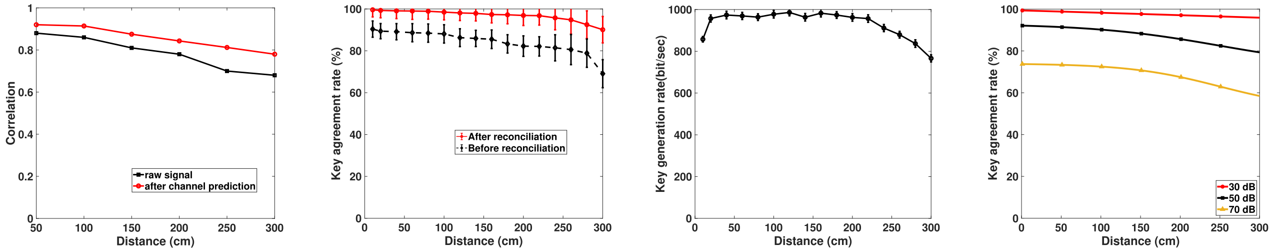
In this experiment, we assess the effectiveness of the proposed channel prediction module. Since the channel prediction module is used to improve the similarity of Alice's channel measurement and Bob's channel measurement, we use the Pearson correlation coefficient as a metric to quantify the similarity. The Pearson correlation coefficient measures the linear correlation between two continuous variables (i.e., Alice's channel measurement and Bob's channel measurement), and it is known as the best method of measuring the relationship between variables of interest because it is based on the method of covariance. Fig. 5(a) displays the Pearson correlation coefficient at various distances, revealing that our method indeed improves the correlation. This improvement is particularly noticeable at greater distances, where acoustic signals exhibit a low signal-to-noise ratio (SNR). As we will demonstrate in Sec. IV-H, increased correlation leads to a higher key agreement rate.

C. Effectiveness of Reconciliation

Next, we evaluate the effectiveness of the proposed reconciliation method by comparing the key agreement rate before

Authorized licensed use limited to: Technical University of Denmark DTU Library. Downloaded on July 01,2024 at 07:58:30 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



(a) Channel prediction (b) With and without reconciliation (c) Impact of distance (d) Theoretical boundary

Fig. 5. Evaluation results.

and after reconciliation. As depicted in Fig. 5(b), the key agreement rate before reconciliation declines as the distance increases and fails to reach 100%. Upon applying the proposed reconciliation technique, the key agreement rate experiences a boost of 4–16%, depending on the distance. A comprehensive comparison with traditional reconciliation methods is provided in Sec. IV-H.

Although the results of Sec. IV-B and Sec. IV-C demonstrate the superior performance of the deep learning-powered prediction and reconciliation module, there still exist several limitations. First, deep learning is known to require extensive data to train a good model. Secondly, deep learning models require more computation resources. As will be seen in Sec. IV-J, compared to the system overhead of InaudibleKey [23], the processing time and energy consumption increase by 2.6× and 2.8×, respectively.

D. Impact of Distance

We now evaluate the influence of the distance between Alice and Bob on the system's performance. Fig. 5(b) reveals a gradual decrease in the key agreement rate as the distance between Alice and Bob expands from 20 cm to 300 cm. This is due to the exponential audio signal attenuation resulting from path loss as distance increases [38]. Fig. 5(c) demonstrates that the key generation rate initially rises when the distance grows from 10 cm to 20 cm, stabilizes between 20 cm and 230 cm, and then drops sharply. The reason for this is that the Line-of-Sight (LoS) channel dominates the signal when the devices are in close proximity, limiting the randomness available for use. However, as the communication distance becomes too large, more environmental noise interferes with the received signal, lowering the SNR and leading to greater discrepancies. By analyzing Fig. 5(b) and Fig. 5(c), we can conclude that a distance range of [20 cm, 220 cm] is a practical pairing range for achieving both a high agreement rate and bit generation rate. Compared to FREE [21] and TDS [4], InaudibleKey2.0 extends the pairing distance by 3.2 times and 44 times, respectively. Fig. 5(d) presents an analysis of the key agreement rate under varying noise levels across distances, using an acoustic signal propagation model with Monte Carlo simulation. For the low noise condition of 30 dB, the agreement rate stays relatively high up to 100 cm but gradually lessens beyond this range. At 50 dB, the rate drops more noticeably with distance, particularly past 150 cm, highlighting moderate noise sensitivity. High noise at 70 dB shows a steep decline in agreement rate, intensifying with distance. Typically, ambient noise levels in office environments range around 30 dB to 40 dB [39], where our system shows robust performance.

E. Impact of Different Environments

In this subsection, we examine the performance of InaudibleKey2.0 under various environments, utilizing data within the range of [20 cm, 220 cm]. Fig. 6(a) displays the key generation rate and key agreement rate across distinct scenarios. As one might expect, the key agreement rate in outdoor environments (i.e., B and D) is slightly lower than in indoor environments (i.e., A and C). This can be attributed to the reduced multi-path effects and increased environmental noise in outdoor settings [6]. When considering the generation rate, it becomes apparent that mobile scenarios (i.e., C and D) produce more bits than static scenarios (i.e., A and B). This occurs because mobile contexts result in enhanced channel diversity and randomness.

We further conduct an experiment to evaluate the performance of InaudibleKey2.0 in unseen environments. Specifically, we train InaudibleKey2.0 using the data from one environment but test it using data from other unseen environments. In this experiment, the model trained in environment A is selected as the base model (the results of the other models are similar but are not included due to space limitation). Before applying the model directly to the new environment, we fine-tune the base model with different percentages of training data from the new environment. Transfer-10% means that 10% of the new data is used to fine-tune the new model on the base model. The result is obtained by testing the fine-tuned model and traditional trained model on the testing set of the new scenarios.

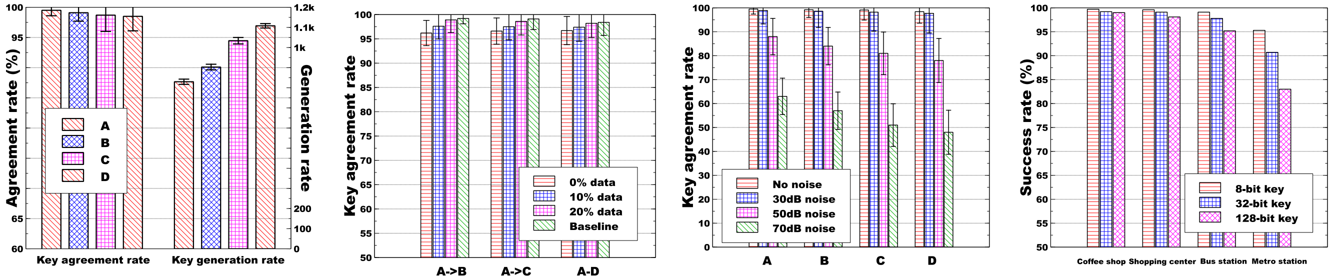
From the results in Fig. 6(b), we can see that when no data of the new environment is used in training, the key agreement rate of InaudibleKey2.0 drops by approximately 3%. When only 10% of the new data is used in training, the performance approaches that of using all the data. To sum up, these results demonstrate the proposed model has good generalization ability and can quickly adapt to new scenarios with limited training data.

F. Impact of Background Noise

Although our analysis demonstrates that InaudibleKey2.0 consistently achieves high agreement rates, it is important to note that these experiments were conducted solely on campus. Real-world environments are more complex and might contain various types of noise. To account for this, we investigate the

Authorized licensed use limited to: Technical University of Denmark DTU Library. Downloaded on July 01,2024 at 07:58:30 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



(a) Impact of different environments (b) Impact of new environments (c) Impact of background noise (d) Performance in real environments

Fig. 6. Impact of environment and noise.

degradation in matching rate as background noise increases. We manually introduce random Gaussian noise within the 18–22 kHz range at different intensities (30, 50, and 70 dB). As shown in Fig. 5(c), the key agreement rate experiences only a minor decline when the noise level is at 30 dB. However, a significant drop in the key agreement rate occurs as the noise level is further increased. In reality, environmental noise within the 18–22 kHz range is minimal in standard office and street settings. Such noise is typically encountered in factories and metro stations [20].

To further validate our findings, we carried out an additional experiment across four common settings: a coffee shop, shopping center, bus station, and metro station. In these tests, Alice and Bob are positioned approximately 1 m apart, and we collect 30 minutes of data for each environment. All measurements are taken between 9 AM and 6 PM. In this experiment, we utilize the success rate (the likelihood of generating the same key) as a metric rather than the matching rate, as we wanted to determine the number of trials users would need to successfully pair two devices in real-world environments. Fig. 6(d) presents the success rate for each environment. Our results indicate that InaudibleKey2.0 achieves over 97% success rates in coffee shops and shopping centers. While a slight decrease in success rate is observed at bus stations, it remains above 95%. We find a significant drop in the success rate at metro stations, which can be attributed to the noise levels in subway stations reaching over 100 dB according to previous studies [40], [41]. Moreover, metro stations feature more noise within the inaudible frequency range. However, when using an 8-bit key, the success rate remains above 95%.

G. Impact of Device Diversity

We now evaluate InaudibleKey2.0's performance when Alice and Bob employ different types of devices. Fig. 7 displays the CFRs of various mobile devices. We observe that when Alice and Bob both use Samsung S10 devices, their CFRs closely align. While there are larger differences when Bob uses an HTC smartphone or HUAWEI watch, the CFR pattern across the frequency band still remains fairly similar. Tab. I presents the key agreement rate for different device combinations. As anticipated, InaudibleKey2.0 attains the highest success rate when Alice and Bob utilize the same kind of devices. The success rate decreases by 2.5–10% when Alice and Bob use different devices. This can be attributed to the varying impacts that distinct microphone and speaker types have on the transmission and recording of acoustic signals. In particular, the Raspberry Pi exhibits the lowest matching rates with other devices due to the use of a low-cost microphone and speaker module, as discussed in Sec. IV-J.

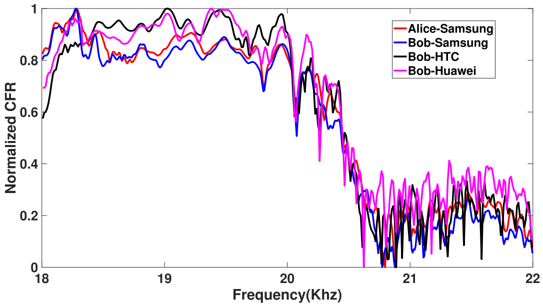


Fig. 7. CFR of different devices.

TABLE I

SUCCESS RATE OF DIFFERENT PAIRS

	Samsung	HTC	Huawei	Raspberry Pi
Samsung	**99.4%**	96.5%	96.3%	93.3%
HTC	96.5%	**98.9%**	95.2%	92.4%
Huawei	96.3%	95.2%	**99.1%**	89.4%
Raspberry Pi	93.3%	92.4%	89.4%	**96.9%**

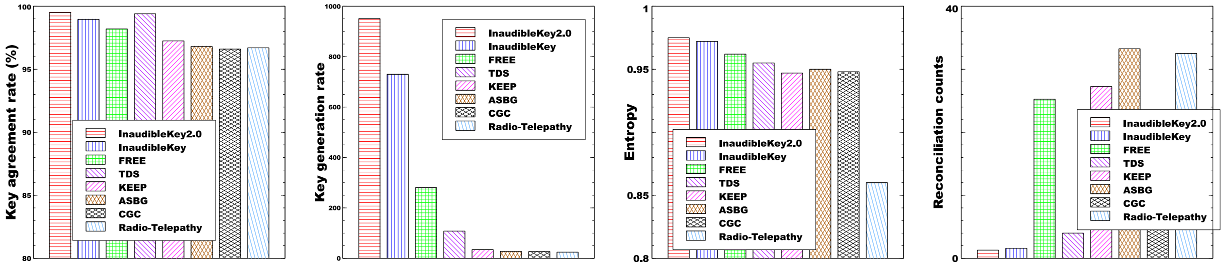
H. Comparison With State-of-the-Arts

In this subsection, we contrast InaudibleKey2.0 with several notable key agreement techniques for mobile networks, including InaudibleKey [23], KEEP [1], ASBG [3], TDS [4], Radio-telepathy [5], CGC [6], and FREE [21]. Note that audio-based device pairing systems fall into two categories: proximity-based and channel reciprocity-based. Since our method belongs to the second category, we compare it with the competing methods in this category only and exclude several classical methods such as [42] and [43] despite the fact they are also based on the audio signal. To ensure a fair comparison, we adjust the parameters of other methods to optimize their performance. Specifically, we use the default settings for InaudibleKey as detailed in our conference version [23]. For FREE, we set the distance between Alice and Bob to 80 cm and the block size to 30. For ASBG, KEEP, and CGC, we establish the guard band ratio and fragment size at 0.35 and 50, respectively. For TDS, we set the block size to 5 and position Alice and Bob at a 4 cm distance, as recommended by the authors. We then compare the key agreement rate, key generation rate, entropy, and reconciliation counts across the various methods.

Fig. 8 displays the performance of various approaches. As seen in Fig. 8(a), the key agreement rate of InaudibleKey2.0 surpasses that of other methods. Although

Authorized licensed use limited to: Technical University of Denmark DTU Library. Downloaded on July 01,2024 at 07:58:30 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



(a) Key agreement rate (b) Key generation rate (c) Entropy (d) Reconciliation counts

Fig. 8. Comparison with state-of-the-arts.

TDS achieves the second-highest agreement rate, it only functions at a short distance (4 cm), which is impractical in real-world scenarios. Fig. 8(b) reveals that InaudibleKey2.0's key generation rate is significantly greater than those of previous works. Specifically, InaudibleKey2.0's key generation rate is 1.3× faster than InaudibleKey [23], 3.9× faster than FREE [21], and 9.1× faster than TDS [4] on average. Several factors contribute to this improvement. Firstly, the audio signal's sampling rate (i.e., 48kHz) is considerably higher than radio channel probing. Secondly, the channel frequency response offers more channel information compared to the channel tap used in FREE [21]. Lastly, the channel prediction module enhances correlation, and the transformer-based reconciliation method recovers more mismatches, as demonstrated in Sec. IV-B and Sec. IV-C.

Fig. 8(c) illustrates the entropy of the extracted keys. By employing KLT to decorrelate the bit sequences, InaudibleKey2.0 achieves higher entropy than other methods. When compared to InaudibleKey, InaudibleKey2.0 attains slightly higher entropy, as the multi-bit quantizer used in InaudibleKey may generate duplicated bit strings, while the proposed quantization model mitigates this issue. Fig. 8(d) presents the information reconciliation counts for different approaches. InaudibleKey2.0 necessitates the fewest information reconciliation counts. To successfully generate the same key, Alice and Bob only need to exchange reconciliation messages an average of 1.3 times using InaudibleKey2.0. In contrast, InaudibleKey requires 1.6 rounds, TDS necessitates 4 pass checks [4], and FREE demands an average of 25 reconciliation information messages [21]. In other terms, InaudibleKey2.0 reduces information reconciliation counts by 1.2–16 times. The findings demonstrate that InaudibleKey2.0 significantly enhances the key generation rate and entropy while substantially reducing reconciliation counts in comparison to the state-of-the-art approaches.

I. Key Randomness

To assess the randomness of the extracted keys, we use the widely applied NIST Test suite [44]. The NIST Test Suite is a collection of 15 statistical examinations designed to evaluate the randomness within binary sequences of any length. Each test in the suite targets different potential patterns of predictability or structured deviations from randomness within a sequence. For instance, the serial test specifically assesses the occurrence rates of all conceivable overlapping patterns of m-bits throughout the sequence. It aims to determine whether the number of occurrences of the 2^m m-bit overlapping patterns is approximately the same as expected for a random sequence. For the details of other statistical tests, readers are encouraged to refer to [44]. The outcomes of the NIST statistical tests are p-values for various test processes, which determine whether a key is random. If the p-value exceeds 1%, the key is considered random. As seen in the results presented in Tab. II, all p-values are greater than 1%, indicating that the extracted keys exhibit a high quality of randomness.

TABLE II
RESULTS OF NIST TEST

NIST TEST	p-value
Serial	0.661
FFT Test	0.548
Longest Run	0.426
Monobit Frequency	0.698
Linear Complexity	0.864
Block Frequency	0.437
Cumulative Sums	0.653
Approximate Entropy	0.535
Non Overlapping Template	0.724

J. System Implementation

To demonstrate the practicality of InaudibleKey2.0 across a range of IoT devices, we develop a prototype of InaudibleKey2.0 for both a Samsung S10 smartphone and a Raspberry Pi 4.

The Samsung S10 features a Snapdragon CPU clocked at 2.84 GHz and runs on the Android 9.0 operating system. It comes with a stereo speaker and dual dedicated microphones with an active noise cancellation feature. Only the bottom microphone is utilized, as it is situated near the speaker. The system is developed in Java, and the MAC algorithm described in Section III-C is based on SHA256 (HMAC-SHA256). In InaudibleKey2.0, the transmitted OFDM signal is saved as a Waveform Audio (WAV) file in a 16-bit Pulse Coded Modulation (PCM) format, which the speaker plays. To minimize the anticipated response time, we implement InaudibleKey2.0 using multiple threads. After launching InaudibleKey2.0, two threads are created. One thread handles the transmission of the WAV file, while the other records the audio signal from another phone after the smartphone transitions into listening mode. The deep learning models are implemented using the PyTorch mobile framework.

First, we compare InaudibleKey2.0 to public key cryptography and the D-H key exchange protocol. For public key cryptography, we use the widely adopted RSA as a benchmark.

Authorized licensed use limited to: Technical University of Denmark DTU Library. Downloaded on July 01,2024 at 07:58:30 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE III

SYSTEM OVERHEAD

	Samsung S10			Raspberry Pi		
	InaudibleKey	RSA	ECDHE-RSA	InaudibleKey	RSA	ECDHE-RSA
Total Time (ms)	321	361	347	499	894	1,027
Energy Consumption (mJ)	308	391	354	486	641	849

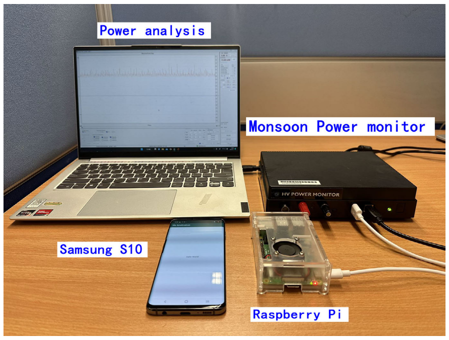
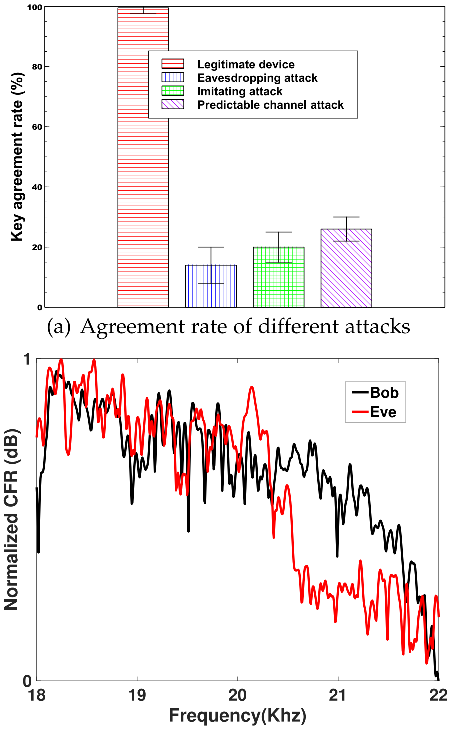


Fig. 9. Experimental setup of energy consumption.

Alice can employ RSA to encrypt a 128-bit key, which Bob can then decrypt. Subsequently, Alice and Bob can secure their communication using AES-128. In this experiment, we utilize a 2048-bit key for RSA, as recommended by NIST [45]. The traditional Diffie-Hellman protocol is vulnerable to MITM attacks and is seldom used in practice. Hence, we employ the widely adopted Elliptic Curve Diffie Hellman Ephemeral with RSA signature (ECDHE_RSA), which is used in Transport Layer Security (TLS). We implement these algorithms on the Samsung S10 and measure their processing time and energy consumption. The cryptographic algorithm implementations are based on the Chilkat library.[5] We obtain the computation time from the development environment console (Android Studio) and average the results from 30 tests. The smartphone's energy consumption is calculated by reading the battery's voltage and current levels, which can be accessed via the Android API. As shown in Table III, RSA requires $361\,\mathrm{ms}$ to complete a round of encryption and decryption using a 2048-bit key. ECDHE_RSA takes approximately $347\,\mathrm{ms}$ to generate a 128-bit key. However, InaudibleKey2.0 needs only $91\,\mathrm{ms}$ to generate a 128-bit key. Thus, InaudibleKey outperforms public key cryptography and the D-H protocol in key distribution on mobile devices.

Second, to assess the feasibility of InaudibleKey2.0 on less powerful IoT devices, we implement our system on a Raspberry Pi 4. In contrast to the powerful CPU in the Samsung S10, the Raspberry Pi 4 features a Broadcom BCM2711 microcontroller clocked at 1.5GHz. Since the board does not include a default speaker and microphone, we connect additional speaker and microphone modules to it. To measure the energy consumption on the Raspberry Pi, we link it to a Monsoon power monitor. Fig. 9 displays the details of the experimental setup. Table III presents the processing time and energy consumption. As the results indicate, although the system overhead of InaudibleKey2.0 on the Raspberry Pi 4 is significantly higher than on the smartphone, it remains substantially more efficient than RSA and ECDHE-RSA.



(a) Agreement rate of different attacks

(b) Bob vs predictable channel attacker

Fig. 10. Security analysis.

We now examine the impact of energy consumption on IoT devices. The battery capacity of the Samsung S10 is 3,400 mAh ($42.8\,\mathrm{kJ}$). Consequently, the energy cost of InaudibleKey2.0 represents $0.8e^{-5}$ of the total energy supply. Assuming a smartphone with a targeted lifespan of one day, this corresponds to an energy budget of $1.75\,\mathrm{kJ}$ per hour. With just 1% of the battery budget ($17.5\,\mathrm{J}$), InaudibleKey2.0 can operate approximately 181 times per hour, meaning it can run continuously every 20 seconds. Similarly, we can estimate that with a $9\,\mathrm{V}$ battery (500 mAh) and 1% of the battery budget, InaudibleKey2.0 can operate every 6 minutes on a Raspberry Pi, or roughly 10 times per hour. These results show that InaudibleKey2.0 has a low system overhead and is more efficient than the public key scheme.

V. SECURITY ANALYSIS

In this section, we investigate whether an attacker can obtain the same key by executing the three attacks described in Sec. II. Specifically, Eve can carry out the following three types of attacks to generate a key K_{Eve} that closely resembles K_{Bob}, with the objective of recovering K_{Bob} using the eavesdropped y_{Bob}.

Against Eavesdropping Attack. In this attack, Eve can intercept all communication traffic in the public channel. However, since she is located outside the safe distance ($\geq 10\,\mathrm{cm}$), her received channel response will be significantly different. As shown in Fig. 10(a), the agreement rate for the eavesdropping attack ranges between 10% and 18% with an average agreement rate of 14%. Thus, if Eve is beyond the safe distance, she cannot deduce the same key due to the different multipath fading channels.

Against Imitating Attack. During this attack, Eve has the ability to monitor the procedure of Alice and Bob generating

[5] Chilkat library: http://www.chilkatsoft.com/ [Online, accessed on July 19, 2023].

Authorized licensed use limited to: Technical University of Denmark DTU Library. Downloaded on July 01,2024 at 07:58:30 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

secret keys. After Alice and Bob leave the location, Eve has her accomplice, David, mimic the movements of Alice and Bob to reproduce the same key. Previous research indicates that merely replicating a user's shaking or walking motions is insufficient to generate the same key for accelerometer-based authentication systems [10], [11], [46]. Similarly, Fig. 10(a) demonstrates that an imitation attack can achieve a higher agreement rate than an eavesdropping attack. Nevertheless, it only reaches an average agreement rate of 20%.

Against Predictable Channel Attack. The predictable channel attack is a straightforward yet effective strategy for compromising key agreement protocols, especially in the case of RSSI-based approaches [3], [6]. In this attack, Eve can purposely move through the LoS between Alice and Bob to produce predictable channel measurements. We evaluate this attack by placing Alice and Bob 100 cm apart with LoS and having a person intentionally walk between them. Subsequently, following the key generation, we replace Alice and Bob with Eve and David and request the same individual to replicate the process. We then compare Eve's key with Bob's key to ascertain whether Eve can produce an identical key. As shown in Fig. 10(a), a predictable channel attack achieves the highest matching rate among the three attack types. However, it still only attains an average 24% matching rate. Fig. 10(b) displays the extracted CFRs of Bob and Eve when the same individual obstructs the LoS signal. While the channel responses of Bob and Eve appear similar at specific frequencies, a significant discrepancy persists at other frequencies, resulting from time-varying channels and hardware differences. Notably, we observed that Eve can produce similar channel responses in lower frequency ranges, yet fails to do so for higher frequency ranges. Two reasons contribute to this result. First, the microphone functions as a low-pass filter with a 22 kHz cutoff frequency [47], causing the acoustic signal to be slightly attenuated in the higher frequency range and resulting in more mismatches. Additionally, Zhou et al. [20] discovered that the performance of different speakers varies more significantly at higher frequency ranges. If Eve employs more advanced hardware, their attack capabilities might improve. However, this remains an open question warranting further investigation.

While the above attacks can achieve an approximate 24% matching rate, Eve cannot identify which bit is correct due to the time-varying nature of the channels. The success rate of an attacker is zero in all the evaluations. Eve's matching rate can be further diminished by reducing the speaker's volume. Given Eve's matching rate, a 225-bit key in our system is equivalent to a 128-bit AES symmetric key. According to the results in Sec. IV-D, generating such a key takes about 0.23 s.

InaudibleKey2.0 assists in the pairing process between two devices but does not verify their identities. As with numerous prior studies [4], [16], our assumption is that devices within pairing distance are legitimate. However, since InaudibleKey2.0 attains a significantly longer working distance compared to earlier systems, it is feasible for an attacker to approach the user and masquerade as a legitimate device to initiate key generation. In such scenarios, we must either involve the user in the process or employ conventional authentication methods, such as a pre-shared key or token-based approaches.

VI. Related Work

Proximity-based methods. Proximity-based approaches facilitate device pairing by leveraging the fact that devices in close physical proximity can measure similar physical information. Researchers have proposed numerous systems that explore various location-sensitive features, including RSSI [7], [8], [48], CSI [4], audio [42], and illumination [43]. Amigo [7] was the pioneering work utilizing a shared radio environment to verify physical proximity, employing the D-H protocol for key establishment. Proximate [8] eliminates the dependency on the Diffie-Hellman protocol but has a low bit generation rate (1-3.5 bit/sec). These approaches share a common limitation: the distance between two legitimate devices must be extremely short, e.g., 1.25 cm in Proximate [8] and 5 cm in TDS [4]. Although the distance constraint is relaxed in [43], it requires a considerable amount of time to complete the pairing process. Device pairing systems for heterogeneous IoT devices are also proposed by exploring the user's physical operation [49] and ambient context information [50]. Still, these methods are orthogonal to the application scenario of this paper.

Channel reciprocity-based methods. In recent years, the field of physical layer key generation has gained popularity among researchers. They have examined key agreement methods for a variety of wireless technologies, such as ZigBee [3], LoRa [51], [52], [53], [54], and Wi-Fi [4], [5]. Among these, RSSI-based key generation techniques are prone to predictable channel attacks and exhibit low bit generation rates. While CSI-based key generation strategies can increase the bit generation rate, the majority of systems depend on specialized hardware to gather CSI information. More recently, investigators have also considered using distinct body channels to connect two mobile devices [17], [55]. Roeschlin et al. [17] used a unique body channel to facilitate secure communication between two devices equipped with electrodes. Yang et al. [55] made use of electromyogram signals originating from human muscle contractions for generating keys. Nevertheless, these techniques necessitate the use of specific sensors, such as electrodes [17] and electromyogram sensors [55].

Activity-based methods. Several studies have explored pairing two devices by leveraging common activities they observe, such as shaking and walking. Lars Erik et al. [56] initially proposed shaking two devices together for pairing. Rene et al. [46] followed the same concept but incorporated secure authentication. Walkie-Talkie [10] utilized a common human activity, walking, to pair IoT devices on the same body. Researchers in [37] and [57] pursued the same idea but employed different techniques. To reduce the need for user intervention, researchers have also proposed using heartbeat signals to generate keys [58], [59], [60]. However, these approaches necessitate that the devices be closely attached to or inside users' skin (i.e., implantable devices) to detect heartbeats.

Acoustic signal-based methods. Acoustic signals have been employed for pairing mobile devices as well

Authorized licensed use limited to: Technical University of Denmark DTU Library. Downloaded on July 01,2024 at 07:58:30 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

[15], [16], [21], [22], [42]. Proximity-based strategies like [16], [42] are impractical due to social distancing limitations. Two recent works [21], [22] share similarities with our system. FREE [21] utilized channel tap, while another work [22] leveraged sound pressure as channel characteristics. However, these metrics offer only a rough estimation of the acoustic channel. In comparison, we employ OFDM technology to modulate the audio signal, enabling fine-grained channel estimation, and propose an optimization algorithm to enhance reconciliation performance. Consequently, we can attain a significantly higher generation rate.

VII. Conclusion

In this work, we introduce InaudibleKey2.0, a novel key generation system for mobile devices using inaudible acoustic signals. We propose several innovative techniques to enhance the key agreement rate and key generation rate. Comprehensive evaluation results demonstrate that InaudibleKey2.0 substantially surpasses state-of-the-art methods. To showcase its feasibility, we implement InaudibleKey2.0 on both powerful and resource-limited IoT devices. Additionally, we evaluate the security of InaudibleKey2.0 against common attacks. The evaluation results reveal that InaudibleKey2.0 is a fast, practical, and efficient key generation protocol for mobile devices, capable of functioning in a variety of environments.

References

[1] W. Xi et al., “KEEP: Fast secret key extraction protocol for D2D communication,” in *Proc. IEEE 22nd Int. Symp. Quality Service*, May 2014, pp. 350–359.

[2] W. Xu, J. Zhang, S. Huang, C. Luo, and W. Li, “Key generation for Internet of Things: A contemporary survey,” *ACM Comput. Surveys*, vol. 54, no. 1, pp. 1–37, 2021.

[3] S. Jana, S. N. Premnath, M. Clark, S. K. Kasera, N. Patwari, and S. V. Krishnamurthy, “On the effectiveness of secret key extraction from wireless signal strength in real environments,” in *Proc. 15th Annu. Int. Conf. Mobile Comput. Netw. (MOBICOM)*, 2009, pp. 321–332.

[4] W. Xi et al., “Instant and robust authentication and key agreement among mobile devices,” in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2016, pp. 616–627.

[5] S. Mathur, W. Trappe, N. Mandayam, C. Ye, and A. Reznik, “Radio-telepathy: Extracting a secret key from an unauthenticated wireless channel,” in *Proc. 14th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, San Francisco, CA, USA, Sep. 2008, pp. 128–139.

[6] H. Liu, Y. Wang, J. Yang, and Y. Chen, “Fast and practical secret key extraction by exploiting channel response,” in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 3048–3056.

[7] A. Varshavsky, A. Scannell, A. LaMarca, and E. De Lara, “Amigo: Proximity-based authentication of mobile devices,” in *Proc. Int. Conf. Ubiquitous Comput.*, 2007, pp. 253–270.

[8] S. Mathur, R. Miller, A. Varshavsky, W. Trappe, and N. Mandayam, “ProxiMate: Proximity-based secure pairing using ambient wireless signals,” in *Proc. 9th Int. Conf. Mobile Syst., Appl., Services*, Bethesda, MD, USA, Jun. 2011, pp. 211–224.

[9] N. Karapanos, C. Marforio, C. Soriente, and S. Capkun, “Sound-proof: Usable two-factor authentication based on ambient sound,” in *Proc. 24th USENIX Secur. Symp.* Berkeley, CA, USA: USENIX Association, 2015, pp. 483–498.

[10] W. Xu, G. Revadigar, C. Luo, N. Bergmann, and W. Hu, “Walkie-talkie: Motion-assisted automatic key generation for secure on-body device communication,” in *Proc. 15th ACM/IEEE Int. Conf. Inf. Process. Sensor Netw. (IPSN)*, Apr. 2016, pp. 1–12.

[11] Y. Shen, F. Yang, B. Du, W. Xu, C. Luo, and H. Wen, “Shake-n-shack: Enabling secure data exchange between smart wearables via handshakes,” in *Proc. IEEE Int. Conf. Pervasive Comput. Commun.*, Mar. 2018, pp. 1–10.

[12] Q. Wang, H. Su, K. Ren, and K. Kim, “Fast and scalable secret key generation exploiting channel phase randomness in wireless networks,” in *Proc. Conf. Comput. Commun.*, Shanghai, China, Apr. 2011, pp. 1422–1430.

[13] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, “Tool release: Gathering 802.11 n traces with channel state information,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 1, p. 53, Jan. 2011.

[14] M. Schulz, J. Link, F. Gringoli, and M. Hollick, “Shadow Wi-Fi: Teaching smartphones to transmit raw signals and to extract channel state information to implement practical covert channels over Wi-Fi,” in *Proc. 16th Annu. Int. Conf. Mobile Syst., Appl., Services*, Jun. 2018, pp. 256–268.

[15] J. Han et al., “Do you feel what I hear? Enabling autonomous IoT device pairing using different sensor types,” in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2018, pp. 836–852.

[16] P. Xie, J. Feng, Z. Cao, and J. Wang, “GeneWave: Fast authentication and key agreement on commodity mobile devices,” *IEEE/ACM Trans. Netw.*, vol. 26, no. 4, pp. 1688–1700, Aug. 2018.

[17] M. Roeschlin, I. Martinovic, and K. B. Rasmussen, “Device pairing at the touch of an electrode,” in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2018, pp. 18–21.

[18] W. Wang, L. Yang, and Q. Zhang, “Touch-and-guard: Secure pairing through hand resonance,” in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, Sep. 2016, pp. 670–681.

[19] R. Nandakumar, V. Iyer, D. Tan, and S. Gollakota, “FingerIO: Using active sonar for fine-grained finger tracking,” in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, San Jose, CA, USA, May 2016, pp. 1515–1525.

[20] Z. Zhou, W. Diao, X. Liu, and K. Zhang, “Acoustic fingerprinting revisited: Generate stable device ID stealthily with inaudible sound,” in *Proc. ACM Conf. Comput. Commun. Security*, 2014, pp. 429–440.

[21] Y. Lu, F. Wu, S. Tang, L. Kong, and G. Chen, “FREE: A fast and robust key extraction mechanism via inaudible acoustic signal,” in *Proc. 20th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, Jul. 2019, pp. 311–320.

[22] D. Q. Bala and B. Raman, “PHY-based key agreement scheme using audio networking,” in *Proc. Int. Conf. Commun. Syst. Netw.*, Jan. 2020, pp. 129–136.

[23] W. Xu et al., “InaudibleKey: Generic inaudible acoustic signal based key agreement protocol for mobile devices,” in *Proc. 20th Int. Conf. Inf. Process. Sensor Netw.* New York, NY, USA: Association for Computing Machinery, May 2021, pp. 106–118.

[24] J. P. Walters, Z. Liang, W. Shi, and V. Chaudhary, “Wireless sensor network security: A survey,” *Secur. Distrib., Grid, mobile, Pervasive Comput.*, vol. 1, p. 367, 2007.

[25] T. S. Rappaport, *Wireless Communications: Principles and Practice*, vol. 2. Upper Saddle River, NJ, USA: Prentice-Hall, 1996.

[26] K. Zhang, P. Werner, M. Sun, F. X. Pi-Sunyer, and C. N. Boozer, “Measurement of human daily physical activity,” *Obesity Res.*, vol. 11, no. 1, pp. 33–40, Jan. 2003.

[27] A. Graves and J. Schmidhuber, “Framewise phoneme classification with bidirectional LSTM and other neural network architectures,” *Neural Netw.*, vol. 18, nos. 5–6, pp. 602–610, Jul. 2005.

[28] Ú. Erlingsson, V. Pihur, and A. Korolova, “RAPPOR: Randomized aggregatable privacy-preserving ordinal response,” in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2014, pp. 1054–1067.

[29] W. Xue, D. Vatsalan, W. Hu, and A. Seneviratne, “Sequence data matching and beyond: New privacy-preserving primitives based on Bloom filters,” *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 2973–2987, 2020.

[30] B. H. Bloom, “Space/time trade-offs in hash coding with allowable errors,” *Commun. ACM*, vol. 13, no. 7, pp. 422–426, 1970.

[31] Y. Choukroun and L. Wolf, “Error correction code transformer,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 38695–38705.

[32] J. Lei Ba, J. Ryan Kiros, and G. E. Hinton, “Layer normalization,” 2016, *arXiv:1607.06450*.

[33] S. Malladi, J. Alves-Foss, and R. B. Heckendorn, “On preventing replay attacks on security protocols,” in *Proc. Int. Conf. Secur. Manag.*, 2002, pp. 1–8.

[34] J. Zhang, T. Q. Duong, A. Marshall, and R. Woods, “Key generation from wireless channels: A review,” *IEEE Access*, vol. 4, pp. 614–626, 2016.

[35] M. Hayashi and T. Tsurumaru, “More efficient privacy amplification with less random seeds via dual universal hash function,” *IEEE Trans. Inf. Theory*, vol. 62, no. 4, pp. 2213–2232, Apr. 2016.

Authorized licensed use limited to: Technical University of Denmark DTU Library. Downloaded on July 01,2024 at 07:58:30 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

[36] W. Xu, C. Javali, G. Revadigar, C. Luo, N. Bergmann, and W. Hu, "Gait-key: A gait-based shared secret key generation protocol for wearable devices," *ACM Trans. Sensor Netw.*, vol. 13, no. 1, pp. 1–27, Feb. 2017.

[37] D. Schürmann, A. Brüsch, S. Sigg, and L. Wolf, "BANDANA—Body area network device-to-device authentication using natural gait," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. (PerCom)*, Mar. 2017, pp. 190–196.

[38] (2019). *Attenuation of Sound Waves*. [Online]. Available: https://www.nde-ed.org/EducationResources/CommunityCollege/Ultrasonics/Physics/attenuation.htm

[39] *Decibel Examples: Noise Levels of Common Sounds*. Accessed: Apr. 5, 2024. [Online]. Available: https://lexiehearing.com/us/library/decibel-examples-noise-levels-of-common-sounds.

[40] D. Lee, G. Kim, and W. Han, "Analysis of subway interior noise at peak commuter time," *J. Audiology Otology*, vol. 21, no. 2, pp. 61–65, Jul. 2017.

[41] R. R. M. Gershon, R. Neitzel, M. A. Barrera, and M. Akram, "Pilot survey of subway and bus stop noise levels," *J. Urban Health*, vol. 83, no. 5, pp. 802–812, Aug. 2006.

[42] D. Schürmann and S. Sigg, "Secure communication based on ambient audio," *IEEE Trans. Mobile Comput.*, vol. 12, no. 2, pp. 358–370, Feb. 2013.

[43] 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 *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2014, pp. 880–891.

[44] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, and E. Barker, *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*. McLean, VA, USA: Booz Allen Hamilton, 2001.

[45] *Recommendation for Key Management*. Accessed: Apr. 5, 2024. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57Pt3r1.pdf

[46] R. Mayrhofer and H. Gellersen, "Shake well before use: Intuitive and secure pairing of mobile devices," *IEEE Trans. Mobile Comput.*, vol. 8, no. 6, pp. 792–806, Jun. 2009.

[47] Y. He et al., "Canceling inaudible voice commands against voice control systems," in *Proc. 25th Annu. Int. Conf. Mobile Comput. Netw.*, 2019, pp. 1–15.

[48] J. Zhang, Z. Wang, Z. Yang, and Q. Zhang, "Proximity based IoT device authentication," in *Proc. IEEE Conf. Comput. Commun.*, May 2017, pp. 1–9.

[49] X. Li, Q. Zeng, L. Luo, and T. Luo, "T2Pair: Secure and usable pairing for heterogeneous IoT devices," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2020, pp. 309–323.

[50] H. Farrukh, M. O. Ozmen, F. Kerem Ors, and Z. B. Celik, "One key to rule them all: Secure group pairing for heterogeneous IoT devices," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2023, pp. 3026–3042.

[51] W. Xu, S. Jha, and W. Hu, "LoRa-key: Secure key generation system for LoRa-based network," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6404–6416, Aug. 2019.

[52] H. Yang et al., "Vehicle-key: A secret key establishment scheme for LoRa-enabled IoV communications," in *Proc. IEEE 42nd Int. Conf. Distrib. Comput. Syst.*, Sep. 2022, pp. 787–797.

[53] J. Gao et al., "A novel model-based security scheme for LoRA key generation," in *Proc. 20th Int. Conf. Inf. Process. Sensor Netw.*, May 2021, pp. 47–61.

[54] H. Yang et al., "Chirpkey: A chirp-level information-based key generation scheme for LoRA networks via perturbed compressed sensing," in *Proc. IEEE Conf. Comput. Commun.*, Jun. 2023, pp. 1–10.

[55] L. Yang, W. Wang, and Q. Zhang, "Secret from muscle: Enabling secure pairing with electromyography," in *Proc. 14th ACM Conf. Embedded Netw. Sensor Syst.*, 2016, pp. 28–41.

[56] L. E. Holmquist, F. Mattern, B. Schiele, P. Alahuhta, M. Beigl, and H. W. Gellersen, "Smart-its friends: A technique for users to easily establish connections between smart artefacts," in *Ubicomp 2001: Ubiquitous Computing*. Berlin, Germany: Springer, 2001, pp. 116–122.

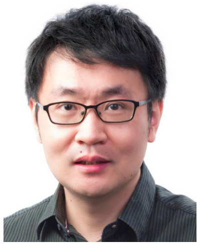
[57] Y. Sun, C. Wong, G.-Z. Yang, and B. Lo, "Secure key generation using gait features for body sensor networks," in *Proc. IEEE 14th Int. Conf. Wearable Implant. Body Sensor Netw. (BSN)*, May 2017, pp. 206–210.

[58] M. Rostami, A. Juels, and F. Koushanfar, "Heart-to-heart (H2H): Authentication for implanted medical devices," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2013, pp. 1099–1112.

[59] F. Xu, Z. Qin, C. C. Tan, B. Wang, and Q. Li, "IMDGuard: Securing implantable medical devices with the external wearable guardian," in *Proc. IEEE INFOCOM*, Apr. 2011, pp. 1862–1870.

[60] Q. Lin et al., "H2B: Heartbeat-based secret key generation using piezo vibration sensors," in *Proc. 18th ACM/IEEE Int. Conf. Inf. Process. Sensor Netw. (IPSN)*, Apr. 2019, pp. 265–276.

Huanqi Yang (Graduate Student Member, IEEE) received the bachelor's degree from the University of Electronic Science and Technology of China in 2021. He is currently pursuing the Ph.D. degree with the Department of Computer Science, City University of Hong Kong. He is supervised by Dr. Weitao Xu. His research interests include smart sensing, the IoT security, the IoT+AI, and wireless networks.



Zhenjiang Li (Member, IEEE) received the B.E. degree from Xi'an Jiaotong University, Xi'an, China, in 2007, and the M.Phil. and Ph.D. degrees from The Hong Kong University of Science and Technology, Hong Kong, in 2009 and 2012, respectively. He is currently an Associate Professor with the Department of Computer Science, City University of Hong Kong. His research interests include wearable and mobile computing, smart health, deep learning, and distributed computing.

Chengwen Luo received the Ph.D. degree from the School of Computing, National University of Singapore (NUS), Singapore. He is currently an Associate Professor with the College of Computer Science and Software Engineering, Shenzhen University (SZU), China. Before joining SZU, he was a Post-Doctoral Researcher in CSE with The University of New South Wales (UNSW), Australia. His research interests include mobile and pervasive computing and security aspects of the Internet of Things.

Bo Wei received the Ph.D. degree in computer science and engineering from the University of New South Wales, Australia, in 2015. He was a Lecturer at Lancaster University and a Post-Doctoral Research Assistant at the University of Oxford. He is currently a Senior Lecturer (Associate Professor) with the School of Computing, Newcastle University. His research interests include edge AI, the Internet of Things, and cyber security.

Weitao Xu (Senior Member, IEEE) received the Ph.D. degree from The University of Queensland in 2017, under the supervision of Prof. Neil Bergmann and Dr. Wen Hu. He is currently an Assistant Professor with the Department of Computer Science, City University of Hong Kong. Before that, he was a Post-Doctoral Research Associate with the School of Computer Science and Engineering (CSE), UNSW, from June 2017 to August 2019. His research interests include mobile computing, sensor networks, and the IoT.

Authorized licensed use limited to: Technical University of Denmark DTU Library. Downloaded on July 01,2024 at 07:58:30 UTC from IEEE Xplore. Restrictions apply.