Key Generation with Ambient Audio

Bo-Rong Chen
UIUC

Hsin-Tien Chiang
Academia Sinica

Heng-Cheng Kuo
Academia Sinica

Yu Tsao
Academia Sinica

Yih-Chun Hu
UIUC

***Abstract*—Digital Contact Tracing (DCT) has been proposed to limit the spread of COVID-19, allowing for targeted quarantine of close contacts. The protocol is designed to be lightweight, broadcasting limited-time tokens over Bluetooth Low Energy (BLE) beacons, allowing receivers to record contacts pseudonymously. However, currently proposed protocols have vulnerabilities that permit an adversary to perform massive surveillance or cause significant numbers of false-positive alerts. In this paper, we present AcousticMask, which encrypts broadcast messages using a key derived from the audio signal present at each device *with sufficient security levels*. Our results show that a receiver sharing the same social space as a sender will hear all of the sender's ephemeral IDs (*EphIDs*) with Hamming distance at most 3, which can be decrypted at the rate of 10 Hz on a Raspberry Pi 4, while achieving a security factor of over 2^{108} against attackers in our testing set, showing AcousticMask is lightweight for DCT and provides sufficient security levels to protect user's privacy.**

I. Introduction

The recent outbreak of COVID-19 makes DCT attractive due to the possibility of replacing cumbersome manual contact tracing. Because of location privacy concerns, existing methods [3], [5], [9], [24] exchange temporary tokens, *i.e., EphIDs*, between users via BLE; these temporary tokens are designed so that two tokens used by the same user should be cryptographically unlinkable.

However, BLE-based DCT faces three issues. First, using *EphIDs* is not untrackable. Currently proposed plaintext *EphID* transmission potentially allows a receiver to use Radio Frequency (RF) tracking (*e.g.,* [4]) combined with human motion modeling [8], [10], [17] to track a user's physical movement across multiple transmissions of the same token; the speed and direction of travel could potentially be used to link a user across token changes, which we call **tracking attack**. Second, since the transmission range of RF signals is typically larger than the transmission distance for most diseases (6 feet or 1.8 meters), hearing an RF transmission is not a good proxy for being within the transmissibility region of a diseased individual. Finally, when an attacker can receive tokens from well outside social distance, that attacker has a wider range of tokens to replay. For privacy reasons, token broadcast is usually not location-constrained, allowing an attacker to use the same token across a wide geographical area, or to replay another users' token in the same way. We call this attack the **identifier-spoofing attack**.

These three vulnerabilities can be exploited by an adversary that covers a sufficiently large geographical region; the adversary can deploy several nodes (similar to the number of WiFi access points needed to cover a similar area), to perform massive surveillance or cause significant numbers of false-positive alerts by DCT. This can be addressed with the *single primitive* of authenticating physical proximity, but the existing co-location authentications are neither for broadcast communications [11], [14]–[16], [25] nor lightweight [21], sufficient for DCT, with practical security level.

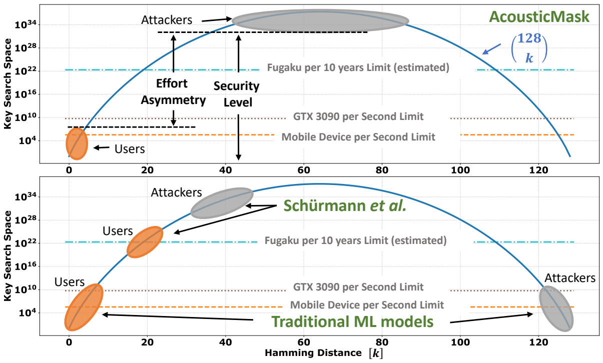


Fig. 1: AcousticMask provides much better effort asymmetry between users and attackers than Schürmann *et al.* [21] and traditional ML models. $\binom{128}{k}$ is the key search space for a Hamming distance of k with a 128-bit key.

In this paper, we present AcousticMask, a framework for securing information exchange with security guarantees, to address these issues. At a high level, AcousticMask aims to maximize the work needed to correct an attacker-key into a user-key, while ensuring the users have similar keys. The user-key is then used in encryption to reduce the ability of the out-of-space attacker to perform both tracking attack and identifier-spoofing attack. Fig. 1 shows that the best published approach, proposed by Schürmann *et al.* [21], does not allow users to have similar keys, whereas traditional Machine Learning (ML) models [12], [27] do not create sufficient effort asymmetry between users and attackers.

Contributions. We present a framework for securing information exchange with security guarantees, which enhances the user's privacy for current BLE-based plaintext DCT without using any network overhead for key exchange. In our evaluations using real and synthetic data, all adjacent (within 1.5 meters) users can receive an *EphID* encrypted with a key at a Hamming distance of at most 3, while achieving a security factor of $2^{108.6}$–$2^{118.4}$ against non-adjacent users in our testing set. Furthermore, AcousticMask produces reasonable performance directly on real-world data, despite being trained only on datasets created in controlled environments. This demonstrates the generalization capability of AcousticMask *without* the usage of real-world data at the training stage.

The overall structure of this paper is as follows: we make our system assumptions and the threat model in Sec. II. We present AcousticMask in Sec. III. Sec. IV discusses our evaluation; in Sec. V, we perform security analysis. We make

978-1-6654-3540-6/22/$31.00 © 2022 IEEE

our conclusion in Sec. VI.

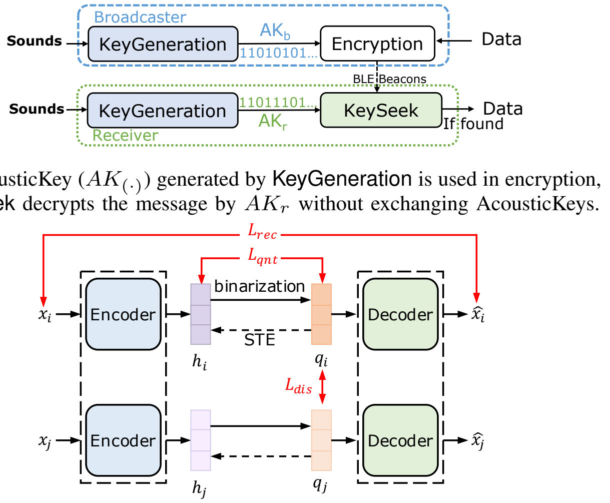
II. Threat Model and Assumption

Threat Model. Our work is based on the idea that building a global audio adversary is significantly more difficult than building a global RF adversary. Because building users typically value sound insulation more highly than RF insulation (in fact, RF insulation can be a negative because of its impact on cellular service), it is common for buildings to provide much less RF attenuation than audio attenuation; for example, 50 dB audio attenuation is common for buildings, while 50 dB RF attenuation would likely create gaps in cellular coverage. As a result, building a global RF adversary requires many fewer listening points. Hence, we assume an adversary that *does not have global acoustic coverage*, either as a sender or a receiver. An attacker in our model can distribute hundreds or even thousands of listening devices or loudspeakers throughout a city, and will be considered to be part of those social spaces, but since our social spaces are small, each listening device will compromise privacy only over a few square meters, and a loudspeaker that is not obnoxiously loud will likewise compromise privacy only over a few square meters. We assume the attacker can use a powerful antenna to increase transmission range and signal strength. The attacker can eavesdrop on the wireless channel from different social spaces. In addition, the attacker can survey public spaces to obtain an acoustic signature, but the attacker should not be considered to be in the same social space if that survey occurs at a different time.

Because we assume that the adversary is not global, we allow any user within the social space to receive *EphIDs*, relying on the adversary's blind-spots to provide privacy through protocols that switch *EphIDs*. We also assume that the user's phone is not compromised (otherwise location can be accessed directly), so an attacker that can continuously play sounds through the victim's phone [23], or listen to their microphone are not part of our threat model.

We do not mean to say that an attacker that can ever hear a user's phone or play a sound through a user's phone can compromise the user's privacy. Rather, our mechanism is designed to ensure that reasonably-constrained attackers cannot deploy a global adversary that allows tracking from *EphID* to *EphID*. Once the attacker loses contact with the victim across one *EphID* change, the attacker can no longer definitively track one *EphID* to the next. Because a gap in coverage breaks the chain of locations. The goal of the attacker is to track users by continuously eavesdropping to violate users' privacy, or spoof the physical proximity with *EphIDs* of positive cases, causing users to get alerts by DCT.

System Assumptions. Each device is loosely time synchronized, such as through the Network Time Protocol (NTP). In addition, packets are transmitted wirelessly, such as over BLE. Packets might be dropped, but the same information will be transmitted multiple times, allowing acceptable performance for solutions that find keys on the scale of minutes. We assume our protocol runs under rich ambient audio environments, and must work in indoor environments, since indoor contagion [22] is more severe in COVID-19.



(a) AcousticKey ($AK_{(.)}$) generated by KeyGeneration is used in encryption, KeySeek decrypts the message by AK_r without exchanging AcousticKeys.

(b) KeyGeneration: AE-based binary key generative model.

Fig. 2: (a) System architecture (b) KeyGeneration is trained including reconstruction loss (L_{rec}), quantization loss (L_{qnt}) and discriminative loss (L_{dis}). During the binarization process, STE is used to directly copy gradients in backpropagation.

III. Proposed Approach

A. *AcousticMask Overview*

To mitigate attacks that spread *EphIDs* to create false positives for contacts and enhance user's privacy that prevents attackers from tracking users due to repeated *EphID* transmissions, we propose AcousticMask, which encrypts *EphID* transmissions using AcousticKeys (AKs) generated by KeyGeneration. Because neighboring devices will generate slightly different AcousticKeys, we develop KeySeek, which searches for and validates keys used for DCT. AcousticMask provides three advantages: (i) it reduces the effectiveness of an attacker that sends long-range contact tracing messages using BLE with powerful antennas; (ii) it provides more precise social-space estimates, especially in the presence of protective shields, and (iii) it decreases the attacker's ability to collect and spread large numbers of *EphIDs* using only Bluetooth devices. Fig. 2a illustrates our system architecture.

B. *KeyGeneration Design*

Binary Key Generative Model. KeyGeneration is an AE-based binary key generative model for audio hashing (Fig. 2b). The encoder of the AE model is formed with a series of convolutional layers followed by a fully connected layer. The first and second convolutional layers consist of 4 kernels of size 4×4, and the third and last convolutional layers have 8 kernels of size 4×4; the dilation rates are 1, 4, 16 and 64 for these four layers, respectively. Batch normalization and ReLU are placed after each layer except for the last fully connected layer, where a hard-tanh activation function is used instead. The hard-tanh function is defined as: hard-tanh $(x) = \max(\min(x, 1), -1)$.

After passing through the last hard-tanh layer, a one-dimensional vector, h, is obtained. A threshold value of 0 is applied on h to obtain a binary vector q, with element values either -1 (when the value of the original element is smaller than the threshold) or 1 (when the value of the original element is greater than the threshold). We choose the values of binary codes to be {-1,1} instead of {0,1} to facilitate an easier training process, which will be described below. The decoder of AE with the binary vector q as the input then passed through a fully connected layer along with batch normalization and ReLU, followed by a series of deconvolutional layers symmetric to the convolutional ones in the encoder. Finally, the reconstructed data is obtained at the output. The encoder and decoder are jointly trained with a reconstruction loss, denoted as L_{rec}, which minimizes the l_1 distance between the input and the output of the AE model. During training, since the binarization process is a non-differentiable estimation, the straight-through estimator (STE) [6] is utilized to copy the gradient of q directly to h on backward-pass and disregards the non-differentiable portion [20]. In order to reduce the error caused by the STE, quantization loss L_{qnt} is applied to minimize the l_2 loss between h and q. In addition, when performing key generations, we then change the value from {-1,1} to {0,1} to facilitate the Hamming distance computation.

To further improve performance, a discriminative loss L_{dis} is introduced to increase the separation between positive (contact) and negative samples. Here, we apply the cosine similarity in calculating L_{dis}, which is formulated as:

$$L_{dis} = -\mathbb{E}_i \left\{ \log \frac{\exp(sim(q_i,q_i^+))}{\exp(sim(q_i,q_i^+))+\sum \exp(sim(q_i,q_i^-))} \right\} \quad (1)$$

where sim is the cosine similarity, q_i is the binary code of the i-th microphone; q_i^+ and q_i^- denote the positive sample (contact) and negative samples of the i-th microphone, respectively. Here, q_i^- includes the binary codes of non-contact microphones during the same time interval and the binary code of i-th microphone itself at other time intervals.

Objective Function for Security Level. From Eq. 1, by optimizing L_{dis}, we accordingly increase the separation between $sim(q_i, q_i^+)$ and $sim(q_i, q_i^-)$. As a result, L_{dis} aims to encourage the Hamming distance between all positive pairs to converge to 0 and the Hamming distance between negative pairs to converge to K. For a pattern recognition task, a clear separation between positive and negative samples is considered a desirable property. For an encryption task, on the other hand, an excessive separation may not be ideal. With a Hamming distance very close to K, an attacker can find the victim's key simply by inverting its own key. To alleviate this potential risk, we modify L_{dis} to specify the Hamming distance between negative pairs to be close to $\frac{K}{2}$ (instead of K, since $\binom{K}{K} = 2^0$, but $\binom{K}{\frac{K}{2}} \approx 2^K$). The modified L_{dis} is:

$$L_{dis} = -\mathbb{E}_i \left\{ \log \frac{\exp(sim(q_i,q_i^+))}{\exp(sim(q_i,q_i^+))+\sum \exp(|sim(q_i,q_i^-)|)} \right\} \quad (2)$$

where the Hamming distance between q_i and q_i^- is close to $\frac{K}{2}$ when $\arg\min(|sim(q_i, q_i^-)|)$. This modification makes KeyGeneration create sufficient security levels between positive and negative examples, thus allowing AcousticMask to *provide security guarantees* against the out-of-space attacker.

Besides, to further emphasize positive pairs, L_{pos} is introduced to measure the distance of the two positive binary vectors by: $L_{pos} = 1 - sim(q_i, q_i^+)$. Finally, the overall objective function is a summation of four losses is: $L = L_{rec} + L_{qnt} + L_{dis} + \lambda L_{pos}$, where λ is the weight for determining the magnitude of the L_{pos}.

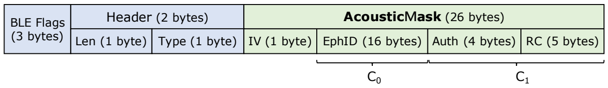


Fig. 3: BLE beacon format.

Fields	Definition	Purpose
IV (1 byte)	Initialization Vector	encrypted *EphIDs* that change from packet to packet
EphID (16 bytes)	$C_0 := SK_{i_0} \oplus EphID$	encrypted *EphID*
Auth (4 bytes)	$C_1 := SK_{i_1}[0:9] \oplus (Auth\|\|RC)$	TESLA [19] authenticator defined by BlindSignedID [7]
RC (5 bytes)		special encryption scheme [18], where $RC := C_0[0:5]$

TABLE I: KeySeek design details, where StreamKeys SK_i are generated by AcousticKeys from KeyGeneration.

C. *KeySeek Design*

We propose KeySeek to allow the receiver to find the AcousticKey used by the broadcaster. Since BLE beacons are limited to 31 bytes [1], we design the AcousticMask payload as shown in Fig. 3. During period t, both broadcaster and receiver record an audio clip and use KeyGeneration to generate their AcousticKeys, AK_b and AK_r, respectively. The broadcaster then uses a 1-byte IV together with her AcousticKey AK_b, using a counter PRG with AES-128 counter mode to encrypt 2 blocks, which operate as StreamKeys SK_i: $SK_{i_0}||SK_{i_1} := PRG(IV) := AES_{AK_i}(IV, ctr)||AES_{AK_i}(IV, ctr + 1)$, where $i = b$, to encrypts *EphID*, *Auth*, and RC as C_0 and C_1. We summarize each field in Table I, where the 1-byte IV is changed each packet to ensure that SK values are not duplicated between packets, thus removing that source of trackability. When the receiver receives the message, she first generates a set of candidate keys, starting from her AcousticKey AK_r, then by flipping a single bit of AK_r, increasing the number of flipped bits up to a threshold T. If the decrypted RC field matches the first 5 bytes of the C_0, she has found the sending key AK_b.

IV. Evaluation

A. *Qualification Metrics*

Security Level. We define two Hamming distances: $\mathtt{d}_{\text{user}}$ and $\mathtt{d}_{\text{atk}}$, where $\mathtt{d}_{\text{user}}$ is the upper bound that a user can decrypt due to computational limitations on mobile devices ($\mathtt{d}_{\text{user}} = 3$ works for most mobile devices and is sufficient for DCT according to our testing), and $\mathtt{d}_{\text{atk}}$ is the Hamming distance between the user's AcousticKey and the AcousticKeys obtained by attackers. We define the probability ($\mathtt{P}_{\text{user}}$ and $\mathtt{P}_{\text{atk}}$) of obtaining AcousticKeys below $\mathtt{d}_{\text{user}}$ and $\mathtt{d}_{\text{atk}}$ by users and attackers, respectively. To interpret the performance of KeyGeneration, we define baseline requirements as follows: for each *EphID*'s 5-minute broadcast interval, we have 30 AcousticKeys (10-second for an AcousticKey). Our goal is that a user in the same social space gets at least 1 AcousticKey within Hamming distance $\mathtt{d}_{\text{user}}$ with probability at least 99%; this requires that each AcousticKey have a probability $\mathtt{P}_{\text{user}} \geq 0.1424$. This baseline is acceptable, since Centers for Disease Control and Prevention (CDC) [2] defines *close contact* in contact tracing as: it is someone who

Recording Scenarios	Type	Room 1 ($4.88\times3.7\times2.7$ meters^3)	Room 2 ($9.97\times6.68\times2.7$ meters^3)
$\mathcal{S}1$	Real	(Sp1, N1)	(Sp2, N2)
$\mathcal{S}2$	Real	(Sp1, N1)	(Sp3, speech)
$\mathcal{S}3$	Synthetic	(Sp1, N1)	(Sp2, N2),(Sp3, speech)
$\mathcal{S}4$	Synthetic	(Sp1, N1)	(Sp2, N2),(Sp3, speech)
$\mathcal{S}5$	Synthetic	(Sp1, N1)	(Sp2, N2),(Sp3, speech)

TABLE II: Datasets used in evaluations, where N1 and N2 are different noise types. (Sp$_i$,$Signal_i$) donates Sp$_i$ played $Signal_i$.

was within 6 feet of an infected person for a cumulative total of 15 minutes or more over a 24-hour period. AcousticMask broadcasts each *EphID* for 5 minutes as a broadcasting interval for each *EphID*, so getting at least 1 AcousticKey within the 5-minute broadcasting interval to have the *EphID* satisfies the definition. At the same time, an attacker from a different social space should not be able to acquire 2 AcousticKeys within Hamming distance $\mathtt{d}_{\text{atk}}$; that is, $\mathtt{P}_{\text{atk}} \leq 0.0149$. We allow an attacker to find a single AcousticKey because a single *EphID* broadcast gives no information to the attacker; the *EphID* and corresponding authenticator are designed to leak no identity information, and all identity risk comes from the repetition of the same *EphID*. For spoofing, the attacker has even more significant difficulty due to the small search space explored by legitimate devices and the relatively low message rate allowed by BLE.

***EphID* Delivery Rate.** To enhance user's privacy without degrading the performance of DCT, we need to consider *EphID* delivery rate ($\mathtt{R}_{EphID}$): during the broadcasting interval, what proportion of *EphIDs* are delivered to users within the same social space. We aim for a delivery rate of almost 100%. As a result, the goal of AcousticMask is to achieve $\mathtt{P}_{\text{user}}$ over at least the probability of 0.1424 to have the keys within Hamming distance of 3, which we call $\mathtt{d1}$ group, and maximizes the Hamming distance of $\mathtt{d}_{\text{atk}}$ that corresponds to $\mathtt{P}_{\text{atk}}$ below the probability of 0.0149, which we call $\mathtt{d2}$ group. Table III summarizes the baseline metrics.

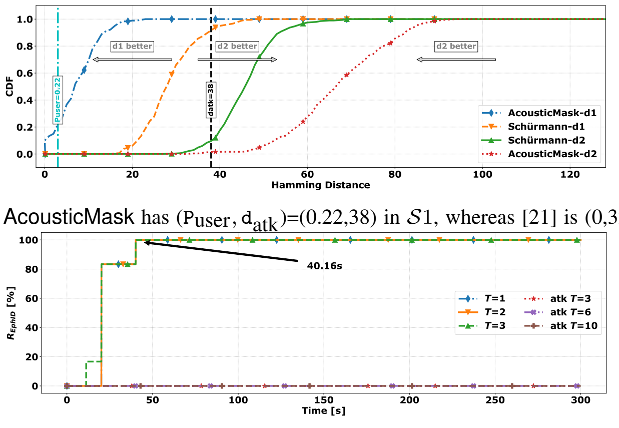
B. *AcousticMask Evaluations*

Dataset Collection. Our audio recordings were made in two meeting rooms side by side. Six microphones (M1-M6) of the same brand (AKG P420) were placed at two positions (M1, M2) in room 1 and four positions (M1-M4) in room 2. Each microphone has one other microphone placed 1.5 meters away in the same room. There were three speakers; one speaker (Sp1) was placed in room 1 and another two speakers (Sp2, Sp3) were placed in room 2.

We prepared two datasets to evaluate the proposed system: real and simulated datasets, which we list in Table II. $\mathcal{S}1$ and $\mathcal{S}2$ were real datasets containing noise and/or speech signals, whereas $\mathcal{S}3$–$\mathcal{S}5$ were generated by N2 and speech signals at three SNR levels: 0, 10 and 50 dB. The contents of speech signals for Sp3 were from the Taiwan Mandarin Hearing in Noise Test (TMHINT) sentences [13]. This dataset includes recorded speech utterances of eight speakers (four male and four female), with each speaker pronouncing 320 utterances. There was no overlap between the training and testing language-speakers and speech contents. The noise signals were obtained from the 100 noises dataset [26]. For N1 and N2, we selected 20 noise types for each during training.

Recording Scenarios	Performance Metrics			
	$\mathtt{P}_{\text{user}}\{\mathtt{d1} \leq \mathtt{d}_{\text{user}} = 3\}$	$\mathtt{d}_{\text{atk}}$	Security Factor	$\mathtt{R}_{EphID}$ (%)
$\mathcal{S}1$ (Noise, real)	0.2222	38	$2^{108.6}$	100%
$\mathcal{S}2$ (Noise+Speech, real)	0.7111	48	$2^{118.4}$	100%
$\mathcal{S}3$ (SNR 0 dB, synthetic)	0.5444	47	$2^{117.6}$	100%
$\mathcal{S}4$ (SNR 10 dB, synthetic)	0.6722	47	$2^{117.6}$	100%
$\mathcal{S}5$ (SNR 50 dB, synthetic)	0.7056	48	$2^{118.4}$	100%
Baseline	≥ 0.1424	$\mathtt{P}_{\text{atk}}\{\mathtt{d2} \leq \mathtt{d}_{\text{atk}}\}$ ≤ 0.0149	-	$\approx 100\%$
Library	0.2083	9	$2^{44.12}$	100%
Office	0.0083	18	$2^{71.63}$	0%
Laboratory	0.2117	12	$2^{54.40}$	100%
Restaurant	0.2929	10	$2^{47.69}$	100%
Train	0.2028	8	$2^{40.38}$	91.67%
Conference Room-1	0.2944	13	$2^{57.55}$	100%
Conference Room-2	0.6583	12	$2^{54.40}$	100%
Game Room	0.4583	13	$2^{57.55}$	100%
Karaoke	0.2708	12	$2^{54.40}$	100%

TABLE III: Performance in testing set ($\mathcal{S}1$–$\mathcal{S}5$) and real-world.



(a) AcousticMask has ($\mathtt{P}_{\text{user}}, \mathtt{d}_{\text{atk}}$)=(0.22,38) in $\mathcal{S}1$, whereas [21] is (0,32).

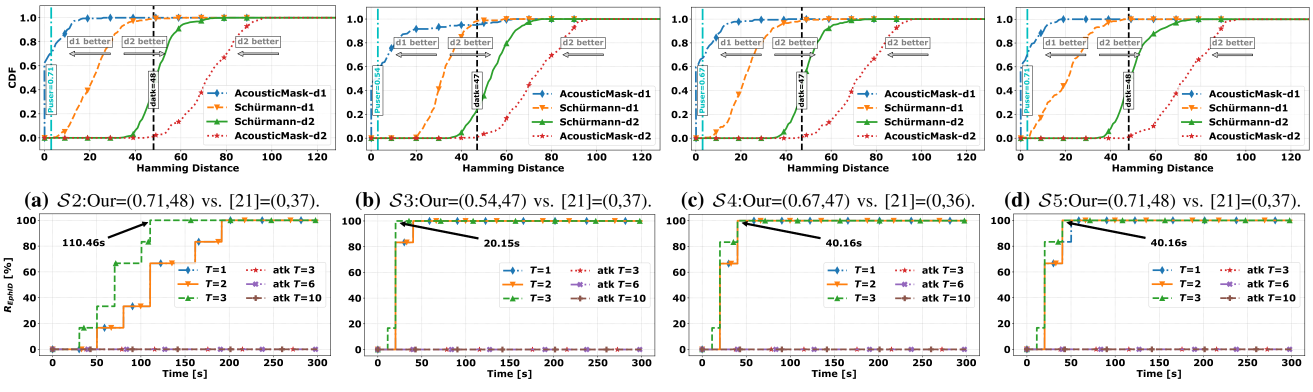
(b) $\mathtt{R}_{EphID}$ for varying thresholds T ($\mathtt{R}_{EphID} = 100\%$ when $T = 3$).

Fig. 4: Evaluation results of $\mathcal{S}1$.

Since we intend to test performance under unseen noise types, we selected another 10 noise types for each during testing, and there is no overlap between the training and testing noise types. Recordings were done with a sampling frequency of 48 kHz. Each scenario had 90 minutes data for training and 10 minutes for testing.

Preprocessing. For any audio signal from the six microphones, we first cut each set of signals into segments of 10 seconds and then downsampled to 8 kHz. Time-frequency features were extracted for each data using a 512-point short time Fourier transform (STFT) with a Hamming window size of 64 milliseconds and a hop size of 32 milliseconds. This resulted in a generated 257-point STFT log-power spectra feature vector. The feature vector was normalized with range of 0 to 1 using min-max normalization and used as the input to the AE model.

Controlled Environments. Table III shows the results of all the scenarios ($\mathcal{S}1$–$\mathcal{S}5$) in the testing set, where $\mathtt{P}_{\text{user}}\{\mathtt{d1} \leq \mathtt{d}_{\text{user}} = 3\}$ and $\mathtt{d}_{\text{atk}}$, which corresponds to $\mathtt{P}_{\text{atk}}\{\mathtt{d2} \leq \mathtt{d}_{\text{atk}}\} \leq 0.0149$, were reported for performance comparison. In all scenarios, the results of $\mathtt{P}_{\text{user}}$ outperformed the baseline, which suggests the users can acquire more than 1 AcousticKey in 30 chances with high probability. In addition, KeyGeneration keeps $\mathtt{d}_{\text{atk}}$ above 38, and this shows that the attacker rarely has AcousticKeys within Hamming distance 38. Even when the attacker gets one AcousticKey, she still needs to break another AcousticKey from the same sender and the same *EphID* to gain any privacy-compromising information, and even then, the leakage is only two locations that are known



(a) $\mathcal{S}2$:Our=(0.71,48) vs. [21]=(0,37). **(b)** $\mathcal{S}3$:Our=(0.54,47) vs. [21]=(0,37). **(c)** $\mathcal{S}4$:Our=(0.67,47) vs. [21]=(0,36). **(d)** $\mathcal{S}5$:Our=(0.71,48) vs. [21]=(0,37).

(e) $\mathcal{S}2$:$\mathtt{R}_{EphID} = 100\%$ when $T = 3$. **(f)** $\mathcal{S}3$:$\mathtt{R}_{EphID} = 100\%$ when $T = 3$. **(g)** $\mathcal{S}4$:$\mathtt{R}_{EphID} = 100\%$ when $T = 3$. **(h)** $\mathcal{S}5$:$\mathtt{R}_{EphID} = 100\%$ when $T = 3$.

Fig. 5: (a)-(d): (·,·) is ($\mathtt{Puser}, \mathtt{d}_{\mathtt{atk}}$), compared with Schürmann *et al.* [21]; (e)-(h): $\mathtt{R}_{EphID}$ for varying thresholds T.

to be the same individual. Fig. 4a and 5a–5d show the results of scenarios $\mathcal{S}1$–$\mathcal{S}5$. We compare against the audio fingerprint proposed by Schürmann *et al.* [21], which uses 9 frequency bands to obtain a 128 bit fingerprint. The figure shows that each scenario, 22.22%, 71.11%, 54.44%, 67.22%, and 70.56% of Hamming distances were within 3. This indicated that for pairing codes, our model created AcousticKeys with Hamming distance strongly concentrated below 3. We can also observe that in our five scenarios, $\mathtt{d}_{\mathtt{atk}}$ were 38, 48, 47, 47, and 48, showing a good separation of d1 and d2. As a result, AcousticMask outperformed Schürmann *et al.*'s method in terms of feasibility of d1 and security levels in the testing set.

Meanwhile, the largest Hamming distances achieved are around 96, 97, 98, 97 and 97 for scenarios $\mathcal{S}1$–$\mathcal{S}5$, respectively, showing that modifying L_{dis} in Eq. 2 successfully prevented the Hamming distance from converging to $K = 128$, and demonstrating that searching around the inverse of the attacker's key is not significantly more efficient than a brute-force search of the entire key space (since $\binom{128}{98} \approx 2^{96.9}$).

Threshold T and $\mathtt{R}_{EphID}$. We evaluated the threshold of KeySeek using a Nexus 5X and a Raspberry Pi 4 to broadcast BLE beacons at 10 Hz. The receiver accepted only successfully-decrypted beacons, and buffered the most recent incoming beacon during the decryption. We split the AcousticKeys from the 10 minutes testing set into 2 5-minute blocks, since BlindSignedID [7] uses each *EphID* for 5 minutes. We set the threshold T to 3 because the computational limitations on the Raspberry Pi 4 make recovering from 4 errors infeasible. During these 5-minute slots, the *EphID* was received, since only one beacon needs to be successfully decrypted, as shown in Fig. 4b and 5e–5h, where the *EphID* delivered rate reaches 100% within 110.46 seconds with Hamming distance 3, whereas the attacker with different capabilities of breaking keys had $\mathtt{R}_{EphID}$ of 0%. All scenarios, $\mathcal{S}1$–$\mathcal{S}5$, have the *EphID* of 100%, as shown in Table III. This means AcousticMask can still be used effectively on DCT while enhancing users' privacy.

AcousticMask in Real World. We further evaluated AcousticMask by the recordings in the real environments from 4 mobile phones, including a pair of iPhone 12 and two heterogeneous Android phones. We placed each pair of phones within the same social space, and recorded each scenario for about 1 hour. The scenarios included library, office, laboratory, restaurant, train, conference rooms, game room, and karaoke. The range of loudness was from 34–75 dB, which included most noise levels excluding harmful levels. Since we did not have negative cases, we chose the first key of each scenario as the attacker's key. Table III reported the performance AcousticMask in all real-world scenarios. For all scenarios having $\mathtt{R}_{EphID}$ of 100%, AcousticMask has $\mathtt{Puser}$ over the baseline and $\mathtt{d}_{\mathtt{atk}}$ from 9–13, which corresponds to a security factor of $2^{44.1}$–$2^{57.5}$. The results first show that KeyGeneration produces reasonable performance directly on real-world data, despite being trained only on the original datasets, thus demonstrating the generalizability of KeyGeneration *without* the usage of real-world data at the training stage. In addition, for the attacker, privacy compromise still difficult, because she needs to break at least 2 AcousticKeys, each with Hamming distance over 9, and because she cannot determine which packets belong to which user, she must perform between $2^{44.1}$–$2^{57.5}$ key explorations *per packet*. We found AcousticMask did not perform well in the office scenario, because we put phones in a silent, empty office. The train scenario was likewise sometimes quiet due to the train's maglev operation and the cultural expectation of silence on the train, leaving little audio signal to create similar keys at each site.

V. Discussion

In this section, we perform analysis of AcousticMask and discuss the limitations of our system.

Security Analysis. Several existing methods [3], [5], [9], [24] neither mitigate Denial-of-Service attack nor prevent from identifier-spoofing attack. An attacker can easily send a large amount of fake *EphIDs* to consume users' phone storage, or replay valid packets with the high power antenna. To our knowledge, only BlindSignedID [7] prevents such attacks by only accepting *EphIDs* with a valid authenticator. However, it still uses plaintext BLE transmissions, so users can be tracked by the attacker. AcousticMask extends this work by using ambient audio to derive an encryption key, preventing malicious users in different social space (*i.e.*, those who cannot access to the common ambient audio) from claiming the physical proximity and from tracking users.

Privacy Analysis. The current methods [3], [5], [9], [24] provide no privacy in terms of trackability among adjacent transmission packets. The *EphIDs* are the same across packets during the same operating time slot. Through human mobility prediction and RF tracking, the attacker can still potentially track the user's location and violate her privacy. However, AcousticMask has 2^8 IVs per AcousticKey, and each IV is used at most once per AcousticKey in a 10-second slot. This allows AcousticMask to send 4–10 packets per second (128 IVs $>$ 100 packets in 10 seconds at 10 Hz) while ensuring that each encrypted packet is distinct. As a result, attackers that cannot decrypt the messages cannot use header information to track users, thereby enhancing the user's privacy.

Without Access to Audio. Mobile phones often already listen continuously to support applications such as Google Assistant and Siri. To further protect users' privacy, AcousticMask KeyGeneration can be made into an OS-level API, so that DCT applications need not access privacy-sensitive audio data. For example, Google Assistant and Siri already have APIs for extending their functionality to cover third-party applications.

Quiet Environments. Table III shows that AcousticMask offers a lower security factor in the library scenario, where the loudness was around 34.1–36.4 dB. Most sounds were whispers, and noise from outside traffic and indoor equipment, such as fans, did not vary sufficiently over time. Although AcousticMask achieved R_{EphID} of 100%, it did not provide a large $\mathsf{d}_{\mathsf{atk}}$. However, we consider the loss of privacy in quiet environments to be an acceptable trade-off, since quiet environments tend to be sparsely populated and tend to lack motion, so mere presence provides significant tracking cues even if the encryption is perfect. In the office environment (44.8–44.9 dB), AcousticMask could not generate common AcousticKeys, because KeyGeneration relies on rich audio signals to generate AcousticKeys, which does not work in a silent environment; however, DCT is unnecessary in such environments due to a lack of social activity.

VI. Conclusion

In this paper, we presented AcousticMask, which explores an AE-based key generative model for generating common keys through audio at each device; these keys can then be used for encryption. Though two nearby devices may not immediately share the same key, over time they will likely have a pair of keys that differ in only a small number of bits, and a receiver can determine a sender's key by searching nearby keys. Moreover, in any 10 second time period of our test set, our model has a 22.22–71.11% probability of acquiring common keys within Hamming distance 3; over the 300-second duration of an *EphID*, all adjacent (within 1.5 meters) users can receive an *EphID* with Hamming distance at most 3 at the rate of 10 Hz, while achieving a security factor of over $2^{108.6}$ against non-adjacent users in our testing set. Finally, AcousticMask can produce reasonable performance directly on real-world data, despite being trained only on the original datasets, thus demonstrating the generalizability of AcousticMask without the usage of real-world data at the training stage. Our framework can both authenticate physical proximity to prevent from identifier-spoofing attack, and enhance user's privacy by encrypting BLE beacons.

References

[1] "Altbeacon protocol specification v1.0," https://github.com/AltBeacon/spec, accessed: 2021-07-22.

[2] "Cdc contact tracing - appendices," https://www.cdc.gov/coronavirus/2019-ncov/php/contact-tracing/contact-tracing-plan/appendix.html, accessed: 2021-07-20.

[3] "Google/apple privacy-preserving contact tracing," https://www.apple.com/covid19/contacttracing, accessed: 2021-07-09.

[4] P. Bahl et al., "Radar: An in-building rf-based user location and tracking system," in *IEEE INFOCOM*, vol. 2. IEEE, 2000, pp. 775–784.

[5] J. Bay et al., "Bluetrace: A privacy-preserving protocol for community-driven contact tracing across borders," *Government Technology Agency-Singapore, Tech. Rep*, 2020.

[6] Y. Bengio et al., "Estimating or propagating gradients through stochastic neurons for conditional computation," *arXiv:1308.3432*, 2013.

[7] B.-R. Chen et al., "Mitigating denial-of-service attacks on digital contact tracing," in *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, 2020, pp. 770–771.

[8] Z. Du et al., "Inter-urban mobility via cellular position tracking in the southeast songliao basin, northeast china," *Scientific data*, vol. 6, no. 1, pp. 1–6, 2019.

[9] J. C. et al., "Pact: Privacy sensitive protocols and mechanisms for mobile contact tracing," *arXiv preprint arXiv:2004.03544*, 2020.

[10] M. C. Gonzalez et al., "Understanding individual human mobility patterns," *nature*, vol. 453, no. 7196, pp. 779–782, 2008.

[11] T. Halevi et al., "Secure proximity detection for nfc devices based on ambient sensor data," in *European Symposium on Research in Computer Security*. Springer, 2012, pp. 379–396.

[12] J. Han et al., "Physical layer secret key generation based on autoencoder for weakly correlated channels," in *ICCC*. IEEE, 2020, pp. 1220–1225.

[13] M. Huang, "Development of taiwan mandarin hearing in noise test," *Department of speech language pathology and audiology, National Taipei University of Nursing and Health science*, 2005.

[14] A. Kalamandeen et al., "Ensemble: cooperative proximity-based authentication," in *ACM MobiSys*, 2010, pp. 331–344.

[15] N. Karapanos et al., "Sound-proof: Usable two-factor authentication based on ambient sound," in *USENIX Security*, 2015, pp. 483–498.

[16] D. Liu et al., "Secure pairing with wearable devices by using ambient sound and light," *Wuhan University Journal of Natural Sciences*, vol. 22, no. 4, pp. 329–336, 2017.

[17] X. Lu et al., "Approaching the limit of predictability in human mobility," *Scientific reports*, vol. 3, no. 1, pp. 1–9, 2013.

[18] R. Pass et al., "A course in cryptography," Online textbook. [Online]. Available: https://www.cs.cornell.edu/courses/cs4830/2010fa/lecnotes.pdf

[19] A. Perrig et al., "The tesla broadcast authentication protocol," *Rsa Cryptobytes*, vol. 5, no. 2, pp. 2–13, 2002.

[20] J. Ramapuram et al., "Improving discrete latent representations with differentiable approximation bridges," in *IJCNN*. IEEE, 2020, pp. 1–10.

[21] D. Schürmann et al., "Secure communication based on ambient audio," *IEEE Trans. Mobile Comput.*, vol. 12, no. 2, pp. 358–370, 2011.

[22] V. Senatore et al., "Indoor versus outdoor transmission of sars-cov-2: environmental factors in virus spread and underestimated sources of risk," *Euro-Mediterranean journal for environmental integration*, vol. 6, no. 1, pp. 1–9, 2021.

[23] B. Shrestha et al., "The sounds of the phones: Dangers of zero-effort second factor login based on ambient audio," in *ACM CCS*, 2016, pp. 908–919.

[24] C. Troncoso et al., "Decentralized privacy-preserving proximity tracing," *arXiv preprint arXiv:2005.12273*, 2020.

[25] A. Varshavsky et al., "Amigo: Proximity-based authentication of mobile devices," in *International Conference on Ubiquitous Computing*. Springer, 2007, pp. 253–270.

[26] Y. Wang et al., "Towards scaling up classification-based speech separation," *IEEE Trans. Audio, Speech, Language Process.*, vol. 21, no. 7, pp. 1381–1390, 2013.

[27] Y. Wu et al., "Auto-key: Using autoencoder to speed up gait-based key generation in body area networks," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, no. 1, pp. 1–23, 2020.