We Can Hear Your PIN Drop: An Acoustic Side-Channel Attack on ATM PIN Pads

Kiran Balagani[2], Matteo Cardaioli[1,4], Stefano Cecconello[1], Mauro Conti[1], and Gene Tsudik[3]

[1] University of Padua, Padua, Italy
[2] New York Institute of Technology, New York
[3] University of California, Irvine (UCI)
[4] GFT Italy, Italy

Abstract. Personal Identification Numbers (PINs) are the most common user authentication method for in-person banking transactions at ATMs. The US Federal Reserve reported that, in 2018, PINs secured 31.4 billion transactions in the US, with an overall worth of US$ 1.19 trillion.

One well-known attack type involves the use of cameras to spy on the ATM PIN pad during PIN entry. Countermeasures include covering the PIN pad with a shield or with the other hand while typing. Although this protects PINs from visual attacks, acoustic emanations from the PIN pad itself open the door for another attack type. In this paper, we show the feasibility of an acoustic side-channel attack (called $\mathcal{P}inDrop$) to reconstruct PINs by profiling acoustic signatures of individual keys of a PIN pad. We demonstrate the practicality of $\mathcal{P}inDrop$ via two sets of data collection experiments involving two commercially available metal PIN pad models and 58 participants who entered a total of 5,800 5-digit PINs. We simulated two realistic attack scenarios: (1) a microphone placed near the ATM (0.3 meters away) and (2) a real-time attacker (with a microphone) standing in the queue at a common courtesy distance of 2 meters. In the former case, we show that $\mathcal{P}inDrop$ recovers 96% of 4-digit, and up to 94% of 5-digits, PINs. Whereas, at 2 meters away, it recovers up to 57% of 4-digit, and up to 39% of 5-digit PINs in three attempts. We believe that these results are both significant and worrisome.

Keywords: Keyboard eavesdropping · PIN security · ATM security.

1 Introduction

The Automatic Teller Machines Industry Association estimates that over 300 million ATMs are deployed worldwide [3]. In the US alone, over 10 billion ATM transactions are performed every year [19]. ATMs have now become an indispensable part of the self-service banking ecosystem. An ATM typically uses a unique physical card (which a customer possesses) along with a PIN (which a customer remembers) to form a two-factor authentication system, wherein the

card uniquely identifies the customer account and the PIN identifies the customer.

In recent years, there have been many attacks aimed at PINs and at information encoded on ATM cards. Such attacks are broadly referred to as skimming operations [25], whereby criminals usually install a card-reader-like device to trick customers into placing (or inserting) their cards and copy the information [7,18]. This is often done in tandem with installing a video camera on the ATM (or in its vicinity) at an angle that allows the criminal to record PIN entry [22]. Recently studied attacks on PINs (e.g., [5,8,26]) went one step further and showed that the attacker does not even have to see the PIN. These side-channel attacks use a recording device (e.g., a video camera [5], a microphone [8], or a thermal camera [26]) placed near the ATM to collect information and use it to infer customers' PINs.

In this paper, we present a new acoustic side-channel $\mathcal{P}inDrop$ attack on ATM PIN entry. Differently from [8], $\mathcal{P}inDrop$ leverages the entire audio track to profile each key on the PIN pad, leading to far more accurate results. Our attack consists of two steps: (1) the attacker builds an acoustic profile (a signature of click sounds) for each key on the target PIN pad, and (2) at PIN entry time, the attacker records audio emitted by each pressed key and compares them to the acoustic profile to infer the actual keys pressed, thereby learning the PIN. These two steps can be carried out in any order.

1.1 Intended Contributions

The main contributions of this work are:

1. We described a novel attack targeting PINs: $\mathcal{P}inDrop$, based on acoustic emanations from commodity ATM PIN pads. We demonstrated that $\mathcal{P}inDrop$ reconstructs up to 94% of 5-digit PINs and 96% of 4-digit PINs within three attempts. We showed that the threat posed by $\mathcal{P}inDrop$ is higher compared to state-of-the-art acoustic side-channel attacks on ATM PIN pads [8,14,20].
2. We evaluated $\mathcal{P}inDrop$ via extensive experiments on two commercially available ATM PIN pad models, collecting acoustic emanations for 5,800 5-digit PINs entered in a simulated ATM (though using real PIN pads) by 58 distinct participants. The resulting dataset is publicly available[5] to the research community. We believe it will be useful in studying the problem further and developing countermeasures.
3. We analyzed the performance of $\mathcal{P}inDrop$ with two recording distances: 0.3 and 2 meters away from the PIN pad. At the distances of 0.3 and 2 meters, up to 96% and 57% (respectively) of 4-digit PINs were correctly learned in three attempts.
4. We assessed the performance of $\mathcal{P}inDrop$ in noisy environments, considering different levels and sources of noise to simulate real-context scenarios. We showed that $\mathcal{P}inDrop$ is still an effective attack at 2 meters with

[5] Dataset link: `https://spritz.math.unipd.it/projects/PINDrop`

low/moderate noise, while it remains effective under any noise condition at 0.3 meters.

2 Related Work

This section overviews attacks based on acoustic emanations from user input devices. We first consider attacks targeting keyboards, followed by those targeting PIN pads. For a comprehensive discussion of keyboard side-channel attacks, we refer to [17].

Attacks on generic keyboards. The first extensive study on keyboard acoustic eavesdropping was conducted by Asonov and Agrawal [2]. It showed that each key can be identified by the unique sound that it emits when pressed. This work investigated the reasons for this behavior, demonstrating that it can be attributed to the placement of keys on the keyboard plastic plate. In particular, when different keys are pressed, the plate emits sounds with different timbers.

Subsequent efforts to infer key sequences from acoustic emanations are based on two types of approaches: (i) extraction of features that allow exploiting the uniqueness of acoustic emissions of pressed keys, and (ii) extraction of temporal information. The former tries to distinguish among keys by their characteristic sound, and relies on either supervised [2,10,11,16] and unsupervised [6,28] machine learning models, depending on the specific attack scenario. Supervised models exploit features, notably Fast Fourier Transform (FFT) coefficients and their derivatives, such as Mel-frequency cepstral coefficients (MFCCs). Supervised algorithms generally achieve better performance in identifying keystrokes. On the other hand, these models have a greater dependence on the keyboard used in training and the users' typing style. A further weakness of supervised algorithms is the need to collect a labeled dataset to be used as a training set. Indeed, the ground truth collection is not a trivial task and could significantly affect the attack's effectiveness. One possible solution is discussed in [1,9]. which take advantage of the audio recorded during a VoIP call to collect a ground truth dataset directly. In this scenario, the attacker can exploit the text typed by the victim in a shared medium (e.g., in the VoIP chat or an email sent to the attacker during the call) to label the keystroke sound.

Unsupervised methods are used to group collected samples into unlabeled clusters. The label-cluster association is made by exploiting the characteristics of the input language. In particular, Zhuang et al. [28] perform labeling using letter frequency, while Berger et al. [6] make an association by selecting words from a dictionary that match specific constraints. Unsupervised approaches overcome the need for a ground-truth dataset. However, the scenarios where these attacks can be applied are limited by the strong assumptions on input text and therefore their performance drastically declines on random letter sequences.

The second approach involves the extraction of temporal features of pressed keystrokes. To this end, many efforts focused on analyzing the Time Difference of Arrival (TDoA) of the audio signal emitted by the keypress. They used one [13]

or more [27] microphones positioned around the input device to triangulate the position of the pressed key.

PIN pad-focused attacks. PIN pads are numeric keypads specifically designed for Point-of-Sale (PoS) terminals and ATMs, They facilitate users to enter their Personal Identification Numbers (PINs). Attacks on PIN pads tend to be different from those on regular keyboards. For instance, it is rather challenging to apply unsupervised techniques with PIN pads since the assumptions about the victim's language are no longer applicable. However, the other types of attacks, such as those based on the uniqueness of the acoustic emission and those based on the temporal information are usually applicable. PIN pads also prompt a new set of assumptions, usually dictated by the specific conditions under which they operate. This paves the way to new and more efficient side-channel attack scenarios. Below, we briefly discuss these attacks.

In [5], the authors demonstrate how to obtain PIN information by exploiting inter-keystroke timings. This information is leaked by recording the timing of appearance of masking symbols (e.g., asterisks) on the screen while the victim is entering the PIN. On a related note, [8], shows how inter-keystroke timing information can be inferred with higher accuracy from the feedback sound emitted by the PIN pad when a key is pressed. It also shows that combining multiple side-channel information (e.g., inter-keystroke timing and thermal residue) improve the probability of reconstructing a 4-digit PIN. Similarly, [14], proposes a user-independent attack based on inter-keystroke timing on a plastic PIN pad.

PIN pad acoustic emanations can also be used to improve security of PIN-based authentication systems. For example, [20] shows that inter-keystroke features obtained from PIN pad-emitted audio, can be used as an additional layer of authentication. The same work also showed how to perform a close-by attack (i.e., with the microphone placed a few centimeters from the PIN pad) on an arbitrary subset of keys. Exploiting the inter-keystroke features on this subset, a 60% accuracy in the identification of the pressed key can be reached. Acoustic information is also used in [24], where a Point-of-Sale (PoS) terminal is tampered with by inserting multiple microphones into it. This allows identifying the pressed key position using triangulation, reaching the average accuracy of 88% for a single key, on three PoS models. Although very effective, this approach requires full physical access to the PoS, thus reducing the attack's applicability and scalability.

3 $\mathcal{P}inDrop$ Attack

Assumptions: We assume that the victim interacts with a generic ATM, performing PIN-based authentication. The ATM is equipped with a PIN pad that emits a feedback sound when a key is pressed. The feedback sound (as perceived by the human ATM users) is the same for all keys. The attacker aims to learn the victim's PIN by placing a microphone near the ATM to record acoustic emanations of the PIN pad. The microphone stores recorded audio. How the microphone stores that audio is not relevant for $\mathcal{P}inDrop$, i.e., it can be stored locally

or off-loaded to a remote site. *PinDrop* attack relies only on that recorded audio.

Preliminaries: To set up *PinDrop*, the attacker must select a target ATM and hide a microphone nearby. The exact placement of the microphone can vary, though in the *PinDrop* setting the maximum distance form the PIN pad is 2 meters (just over 6′):

1. Concealed on the attacker's body, in case of a real-time attack. Albeit, strictly speaking, concealment is not required, since a regular smartphone microphone can be used, and it need not be hidden from view (as it is unlikely to arouse suspicion).
2. On any surface (walls, floor, ceiling) near the ATM. In this case, it might be in plain sight, especially, if its size and shape are inconspicuous enough not to be noticeable. It could also be partially hidden from view (e.g., behind a column or a light fixture), or even within or behind some normal-looking object, e.g., a vent, a light-switch or a garbage can.

As shown in Figure 1, *PinDrop* consists of four phases: 1) PIN Recording (Section 3.1), 2) Data Processing (Section 3.2), 3) Model Generation (Section 3.3), and 4) PIN Inference (Section 3.4),

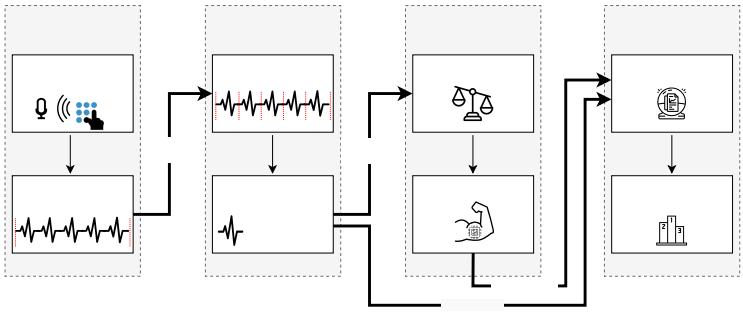


Fig. 1: *PinDrop* attack phases.

3.1 PIN Recording

The goal of this phase is to come up with two datasets (training and attack) with audio recordings of entered PINs. This takes two steps:

A.1 **Audio Recording** using a microphone placed near the ATM.

A.2 **PIN Extraction**, i.e., isolation of the sequences of feedback sounds emitted by the PIN pad, given the knowledge of the number of digits in the PIN, e.g., the beginning and the end of the 5-digit PIN entry.

To build the *training set*, the attacker must enter a set of PIN sequences on the target PIN pad. The sequences must be representative of all ten numeric keys. Once this step is completed, the attacker has a table of entered PINs and their

corresponding audio. The *attack set* consists of the audio recordings entered by the victim.

3.2 Data Processing

This phase is conducted on the data entered by both the attacker and the victim. It also consists of two steps: segmentation of the PIN audio signal into individual key-press sounds, and extraction of corresponding features.

B.1 **Segmentation:** The attacker uses the feedback sound emitted by the PIN pad as a signal that a key has been pressed. This can be achieved via the characteristic frequency of the feedback sound, as in [8]. The attacker segments the signal, using time windows centered at the detected key-press. The window size is chosen to comprise the entire audio segment related to a single key-press.

B.2 **Feature Extraction:** The attacker extracts features descriptive of a key-press sound. Prior results show that short-term power spectrum can be used for this type of a classification problem. In particular, [9] shows that mel-frequency cepstral coefficients (MFCC) [15] achieve the best performances for discriminating among the sounds of different keys. This step yields two feature sets: (1) a labeled training, and an (2) unlabeled attacker.

3.3 Model Generation

This phase is applied to the labeled training set in order to train a classifier.

C.1 **Down-sampling:** Since we make no assumptions about how often a victim uses a specific digit in the PIN, it may be necessary to down-sample the data by classes before proceeding with training. The down-sampling mitigates over-fitting and leads to a balanced dataset where each class (i.e., each digit) has the same number of samples.

C.2 **Model Training.** The attacker trains a multi-class classifier to predict the digit based on its emitted key-press sound. The class labels output by the classifier are the keys (digits) of the PIN pad. Together with the predicted digit, classifiers also output the prediction probability of each class.

3.4 PIN Inference

In this phase, the attacker utilizes the trained classifier to guess a victim's PIN. The output is a sequence of all possible PINs ordered by probability. This ordering allows the attacker to minimize the number of attempts to guess the PIN. In a real-life setting, ATM cards are usually blocked after three failed attempts. This phase involves two steps:

D.1 **Prediction:** The attacker reconstructs the PIN entered by the victim applying the classifier trained in the previous phase to the attack set. As input to the classifier, the attacker feeds the features of a single key of the victim's PIN. This is repeated for each digit of the PIN.

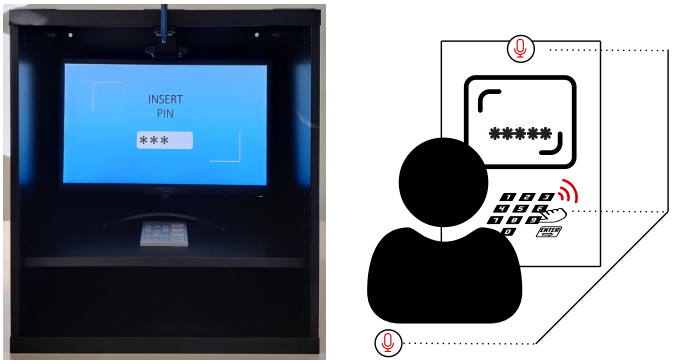
D.2 **PIN Ranking.** The classifier yields a probability for each digit to be the one actually pressed by the victim. Combining the probability set of each input, the attacker builds a ranking of the most likely PINs. The probability assigned to a PIN is the product of the probability of each digit in that PIN.

4 Experimental Setting

To assess the feasibility of *PinDrop*, we collected a large dataset of keystroke sounds, as detailed in this section.

4.1 Data collection

We performed two separate data collection efforts on two commercially available (commodity) metal PIN pads: DAVO LIN Model *D-8201 F* (Figure 3a) [6] and Model *D-8203 B*(Figure 3b) [7]. For clarity's sake, we refer to *D-8201 F* as *PAD-1* and *D-8203 B* as *PAD-2*. For usability reasons, both pads emit a specific feedback sound (the same for all keys) when any key is pressed. In all experiments, we embedded each PIN pad into a simulated ATM (Figure 2a).



(a) *The simulated ATM.* (b) *The testbed configuration used in the experiments.*

Fig. 2: *PinDrop* experimental setup.

The simulated ATM size is based on a real ATM [12]. It is 0.6m wide, 0.64m high, and 0.4m deep. At 0.15m above the ATM base, we inserted a shelf upon

[6] https://www.davochina.com/4x4-ip65-waterproof-industrial-metal-keypad-stainless-steel-keyboard-for-access-control-atm-terminal-vending-machine-p00103p1.html

[7] https://www.davochina.com/4x4-ip65-stainless-steel-numeric-metal-keypad-with-waterproof-silicone-cover-p00126p1.html

(a) *PAD-1: DAVO LIN Model D-8201 F*

(b) *PAD-2: DAVO LIN Model D-8203 B*

Fig. 3: Two commodity metal PIN pads we used.

which we placed the PIN pad and the monitor. The keyboard is 1.1m above the ground. To record keystroke sounds, we used the microphones of two *Logitech HD C920 Pro* webcams: one placed on the ATM's chassis 0.3m above the PIN pad, and another microphone 2m in front of the ATM, as shown in Figure 2b.

The first data collection effort involved 38 participants (23 male and 15 female, average age 38.97 ± 11.36), while the second involved 20 participants (11 male and 9 female, average age 29.50 ± 5.74). Together, that makes the total of 58 participants who entered $5,800$ 5-digit PINs. Participants were university employees and students who participated voluntarily without compensation. The average duration of an experiment was 15 minutes. We used both these data collections to obtain datasets of 4-digit PINs by removing the last key entered by the participants from each 5-digit PIN. Since the attack takes advantage of the sound emitted when a key is pressed, shortening the PIN does not affect the reliability of the dataset. We selected 4 and 5-digits PINs to be comparable with the works [5,8]. After being informed about the study's goals and the confidentiality and anonymity of the data, all participants provided written informed consent for their volunteer participation. At the University of Padua, where the experiments were carried out, a formal review process for research involving human participants was not required, so such ethical considerations were considered based on the authors' past experience with similar experiments. During the experiments, participants were asked to stand in front of the simulated ATM, and remain silent for the duration. A participant's task consisted of typing 100 5-digits PINs randomly generated, divided into four batches of 25 PINs. This split was made to allow for short breaks between batches in order to lower fatigue. PINs were displayed one at a time on the ATM screen: once a PIN is entered, the participant presses the Enter button to proceed to the next PIN.

Regardless of the individual's typing behavior and familiarity (or lack thereof) with a given PIN or the PIN pad, we decided to randomize the order of PINs, rather than ask users to enter the same PIN multiple times. This approach gen-

eralizes the $\mathcal{P}inDrop$ attack, which is actually applicable to both mnemonic PINs and One Time Passwords (OTPs). We also collected the key logs of the PIN pad via the USB interface to create ground truth. In particular, for each pressed key, we collected both the "key-down" (press) and "key-up" (release) events. Moreover, we synchronized the recordings with the timestamp of these key events. We found no significant differences in synchronizing recordings using logs or the feedback sound as suggested in [8]. All recordings were done with a sampling frequency of $44,100$Hz and then saved in the 32-bit WAV format.

4.2 Classification Methods

To identify the key pressed by the victim, we experimented with four well-known and popular classifiers: Support Vector Classification (SVC), k Nearest Neighbors (KNN), Random Forests (RF), and Logistic Regression (LR). We applied a repeated nested crossfold validation to evaluate the performance of our approach. The pipeline varies on the number of attackers (i.e., a single attacker or a group) included in the training set.

In the outer loop, we randomly selected the attacker(s) among the participants. This procedure was repeated 10 times generating 10 groups of attackers. The inner loop consists of a k-fold cross-validation, where k depends on the number of attackers. If the training set contains samples from a single attacker, we used 5-fold cross-validation, since a user-independent split is not applicable. If samples from at least two attackers are present in the training set, we use a k-fold cross-validation user-independent where k is the number of attackers.

We varied hyper-parameters by using the grid search on all four considered classifiers. For SVC, we considered a linear kernel and varied C among: $[10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}]$. For KNN, we varied the number of neighbors to among: [1, ..., 20]. For RF, we considered from 10 to 100 estimators (steps of 10 and extremes included) and a max depth from 6 to 31 (steps of 5 and extremes included). Finally, LR was evaluated for ℓ_1 and ℓ_2 penalties, with C ranging from 10^{-4} to 10^{4}.

5 Experimental Results

We evaluated $\mathcal{P}inDrop$ in different scenarios, showing its performance in the different conditions in which the attacker may find himself. Section 5.1 describes how we evaluated different classifiers and consequently selected the best for our purpose. Sections 5.2 and 5.3 report the results for our algorithms on the key classification task and PIN classification task, respectively. Finally, Section 5.4 compares the performance of $\mathcal{P}inDrop$ with the results obtained in the state-of-the-art.

5.1 Model evaluation

To assess the performance of our classifiers, we evaluated different attack scenarios. In particular, we considered two settings: (i) number of distinct attackers

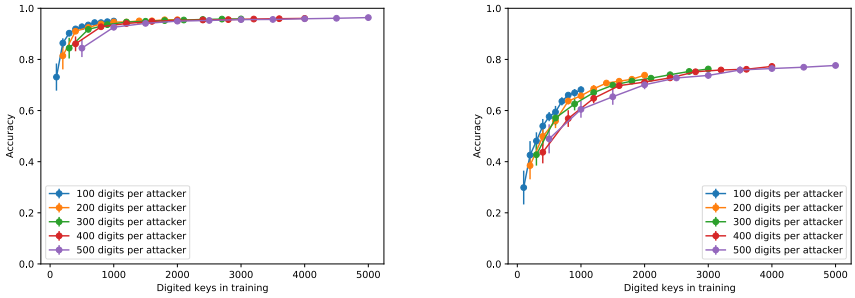
and (ii) the number of digits entered by each attacker. We varied the number of attackers included in the training set between 1 and 10. This range has been selected to reflect a realistic attack scenarios. We varied the number of digits entered by each attacker in increments of 100, i.e., 100, 200, 300, 400, or 500. The performance of our attack was evaluated on all possible combinations between the number of attackers and the number of digits entered by each attacker.

To select the best classifier, we compared the PIN validation accuracy of all the classifiers across different scenarios (i.e., PIN pads, and distances) and settings (i.e., number of digits per attacker, and number of attackers). SVC and LR achieved comparable performance, outperforming KNN and RF. In particular, LR achieved higher validation accuracy on *PAD-1*, while SVC showed better performance on *PAD-2*. In Appendix 8.1, Table 2 reports a comparison of the validation accuracies for all the investigated classifiers, considering five attackers that train the classifiers with 500 digits each (i.e., training size = 2500 digits).

5.2 Single Key Inference

We report the LR classifier performance for the *PAD-1* and the SVC classifier performance for the *PAD-2* based on the validation results. In Figure 4 we show single key accuracy comparison for all the considered settings (i.e., the number of attackers and the number of digits entered by each attacker) in our four scenarios (see Figure 8, and Appendix 8.2 for PAD-2 results). Each graphic depicts how the accuracy varies in the considered scenario as the number of entered keys included in the training set varies. Further, each graphic shows five curves representing the number of digits entered by the attackers, while the bullets of a curve represent the number of attackers included in the training set. The bullets have an increasing value from left to right: the first bullet (from left) of each curve indicates the result obtained when only one attacker has been included in training, the second indicates the result obtained when two attackers were included in training, and so on. Therefore, the number of numeric keys included in the training set varies according to the number of attackers and the number of digits entered by each attacker. We note that the accuracy is significantly affected by the training set's size (i.e., entered keys in training) and the distance. Interestingly, with the same number of entered keys in training, the accuracy improves due to the number of attackers. For example, if we set the number of entered keys in training at 400, we can see that in all scenarios, the accuracy obtained by four attackers typing 100 keys each (i.e., 20 5-digit PINs per attacker) is significantly higher than a single attacker typing 400 keys (i.e., 80 5-digits PINs). This may depend on the variability of the data used to train the classifiers. Each person has a slightly different typing style [20] (e.g., pressure, typing speed), and adding more attackers would introduce higher variance in the training set, helping our classifiers to generalize improving their classification performance over a test set.

Appendix 8.2 provides experiments where we analyzed how our classifiers mis-classify the true key to investigate how spatial locality interferes in the classifiers' predictions.



(a) *PAD-1, 0.3m* (b) *PAD-1, 2m*

Fig. 4: Key accuracy on the testing set for the best classifiers.

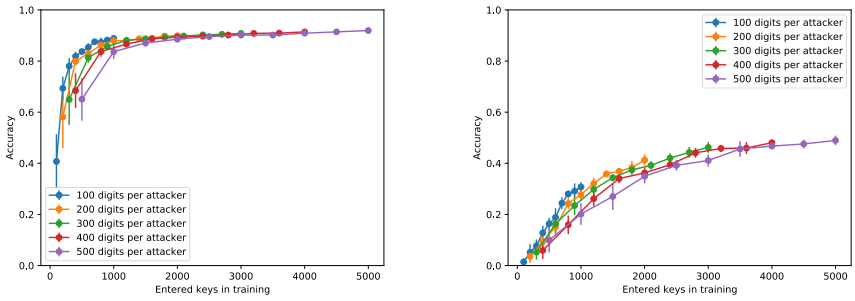
5.3 PIN inference

In a realistic context, an attacker generally has three attempts to guess the victim's PIN (i.e., the max number of incorrect PIN entries allowed before blocking the card). In this section, we report on the performance of our approach in PIN reconstruction in TOP 3-accuracy, i.e., only the three most probable PIN predictions. In Figure 5 we show the performance of the classifiers in the reconstruction of 4-digit and 5-digit PINs according to the different settings (i.e., PIN pad and distances). Further, similar to Figure 4, each graphic reports the performance for all settings on PAD-1 (see Figure 10, and Appendix 8.2 for PAD-2 results).

The results show that the effectiveness of the attack in each scenario. In particular, at 0.3m away, we can reconstruct correctly within three attempts up to 94% 4-digit PINs for *PAD-1* and up to 96% PINs for *PAD-2*. Although the performance worsens by increasing the distance at which the microphone is placed, *PinDrop* manages to reconstruct within three attempts up to 57% of the 4-digit PINs for *PAD-1* and up to 50% for *PAD-2* at 2m away. At 0.3m, the accuracy graphs reach a plateau at around 1500 digits in training. On the contrary, at 2m, the accuracy seems not to reach the plateau even with a training of 10 attackers and 500 digits per attacker (i.e., 5000 digits in training). This behavior is particularly marked in *PAD-2*, where the increase appears almost linear also with a high number of digits in training. This could be partially due to the classifier used in the specific scenario (i.e., LR for *PAD-1* and SVC for *PAD-2*) in addition to the physical differences between the two PIN pads.

Comparing the performance on two PIN pads (fixing the number of attackers and entered keys per attacker), the accuracy on *PAD-1* appears generally higher than the one on *PAD-2*. This applies to both distances. The number of attackers significantly affects performance with the same number of entered keys in training. For example, in *PAD-1* at 0.3m, the threshold of 80% of 4-digit PINs reconstructed in three attempts is reached with three attackers whom enter 100

digits each (i.e., 300 total digits), or two attackers whom enter at least 200 digits each (i.e., at least 400 total digits).



(a) *PAD-1 and microphone placed at 0.3m* (b) *PAD-1 and microphone placed at 2m*

Fig. 5: 5-digit PINs inference performance within 3 attempts for the best classifiers.

5.4 Comparison with the state-of-the-art

To evaluate $\mathcal{P}inDrop$, we compare its with that of state-of-the-art attacks exploiting acoustic emanations of PIN pads [8,14,20,24]. Table 1 summarizes the results (with 10 attackers entering 500 digit each) in terms of key accuracy and PIN reconstruction accuracy within three attempts.

Both [14] and [8], exploit inter-keystroke timing. Although in [14] the distance at which the acoustic information is collected is unspecified, such attacks can be carried out from a distance over one meter, as demonstrated in [8]. The distance significantly decreases the risk of the attacker being detected. However, the reported performance is rather poor, since the PINs correctly reconstructed within three attempts were less than 1% for both attacks. However, from a greater distance (i.e., 2m) $\mathcal{P}inDrop$ outperform [14,8] achieving the accuracy of 44% and 54% on 5-digit and 4-digit PINs, respectively. Most effective attacks are those carried from a significantly shorter distance. In particular, [20] records acoustic emanations with a microphone placed at 0.05m from the PIN pad. This work obtains 60% key accuracy on a sub-set of keys (i.e., 6 on 10). Since we can not estimate the real accuracy considering all the 10 digits we decided for fairness, to leave this upper-bound. Under this assumption, we derived that this attack may achieve 4-digit and 5-digit PIN accuracies of 27.36% and 16.42%, respectively. Comparing these results with the performance of $\mathcal{P}inDrop$, we can see how $\mathcal{P}inDrop$ achieves better accuracy for both 0.3m and 2m.

The last method we consider was proposed by De Souza [24]. This attack assumes that two microphones are placed inside a PoS under the PIN pad.

Unlike other methods, it uses the time of arrival of the acoustic signals. The performance achieved by the De Souza is slightly better to $\mathcal{P}inDrop$ from 2m. However, $\mathcal{P}inDrop$ has better performance from 0.3m (i.e., a 26% increase in 4-digit PINs and a 33% increase in 5-digit PINs). Moreover, $\mathcal{P}inDrop$ differs from [24] in that it does not require physical tampering with the device, even if the attack is performed from 0.3m away.

	Key Accuracy	4-digit PINs	5-digit PINs	Recording Distance
Liu [14]	NA	0.26% *	0.11% *	NA
Cardaioli [8]	NA	0.72%	NA	$1.50m$
Panda [20]	60.00%	27.36% **	16.42% **	$\sim 0.05m$
De Souza [24]	87.60%	68.40% **	59.92% **	$0.00m$ ***
$\mathcal{P}inDrop$	95.84%	94.64%	92.79%	$0.30m$
$\mathcal{P}inDrop$	74.58%	53.75%	43.99%	$2.00m$

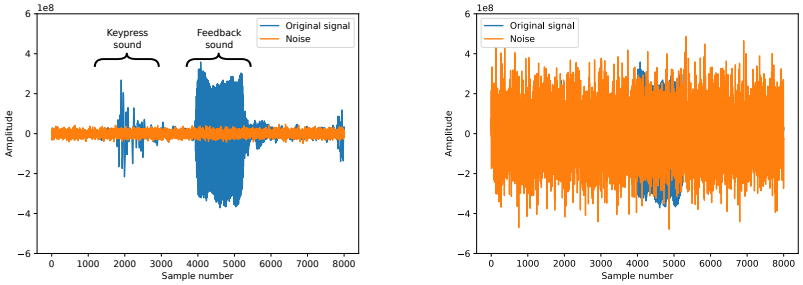
* Performance derived from the proportion of human-chosen PINs and the accuracy of each PIN strength level reported in the paper.
** Performance estimated from reported key accuracy, assuming the prediction error to be equally distributed.
*** Multiple microphones are integrated in the device.

Table 1: Comparison between $\mathcal{P}inDrop$ and the state-of-the-art results on single key accuracy and percentage of guessed PINs within three attempts. If the score cannot be derived from the reference paper, we report N/A.

6 Impact of Noise on $\mathcal{P}inDrop$

In Section 5 we demonstrated the effectiveness of $\mathcal{P}inDrop$ in a noise-controlled environment. This scenario can be traced back to ATM rooms commonly found in banks or city centers. To evaluate the effectiveness of $\mathcal{P}inDrop$ in other contexts (e.g., external ATMs), we simulated two different noise sources: i) road noise produced by urban traffic and ii) Gaussian noise. We modulated the two sources to obtain four levels of SNRs (Signal to Noise Ratios): very low noise (SNR 10dB), low noise (SNR 5dB), high noise (SNR -5dB), and very high noise (SNR -10dB). In Figure 6, we show the comparison between the audio emitted the sound emitted by a key press (with the corresponding feedback sound) and two amplitude levels of the modulated Gaussian noisy signal. Following the procedure described in Section 4, for each considered SNR, we trained and tested $\mathcal{P}inDrop$ with the perturbed signals obtained from the sum of the original signal with the corresponding modulated noise.

To simulate the noise produced by urban traffic, we extracted a set of urban noises from the *AudioSet* [23] dataset made available by Google. Accordingly to the four considered SNRs levels, we modulated the noises, and we added them to



(a) *SNR 10 dB* (b) *SNR -10 dB*

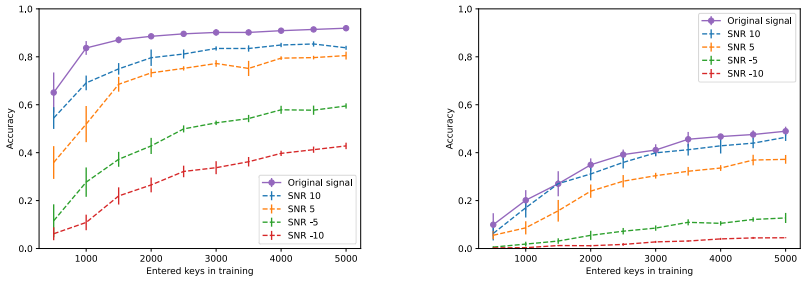
Fig. 6: Comparison between very-low and very high levels of Gaussian noise with the original sound signal of a keypress.

the original signal. In particular, 99% of the power of the considered set of urban noises ranges between 125Hz and 2500Hz, in line with the literature [4,21]. Similarly, to evaluate if the addition of a noise that covers all frequencies affects the performance of $\mathcal{P}inDrop$, we perturbed the original signal with four modulated Gaussian noises amplitude, according to the four SNRs considered.

Figure 7 shows the results of $\mathcal{P}inDrop$ trained on the perturbed PAD-1 dataset (configuration 500 digits per attacker) in inferring 5-digit PINs within three attempts. The graphs suggest that both at 0.3m and at 2m distance regardless the source of noise, $\mathcal{P}inDrop$ remains very effective when low noisy signals are added (i.e., SNR 10dB and 5dB). Further, Figure 7 highlights how the addition of low noises has a greater impact on the performance of $\mathcal{P}inDrop$ at 0.3m than at 0.2m. This difference in performance can be related to the more significant background noise component already present in the original signal recorded at 2m, making the algorithm more robust at low perturbation levels. Figure 11 in Appendix 8.2 reports the results for PAD-2.

For higher noise levels (i.e., SNR -5dB and -10dB), $\mathcal{P}inDrop$ still manages to reconstruct a significant percentage of PINs when the attack is performed from 0.3m (e.g., up to 59% with SNR -5dB and up to 43% with SNR -10dB). However, the performance obtained at 0.3m by $\mathcal{P}inDrop$ on sounds perturbed by Gaussian noise are slightly lower than those obtained with urban traffic perturbation. This difference can be reconducted to the range of frequencies perturbed by the two sources of noise: Gaussian noise affects the entire spectrum, while urban noise has a limited frequency band. At 2m, the performance of $\mathcal{P}inDrop$ degrades significantly with high-noisy perturbation, suggesting that the information contained in the original signal is no longer sufficient to make the attack effective in a very noisy environment. In contrast to the attack scenario at 0.3m, at a distance of 2m we do not notice significant differences between accuracies of PINs reconstructed from audio perturbed with urban noise and those reconstructed from audio perturbed with Gaussian noise. This suggests that the high-frequency

component (i.e., above 2500Hz) is less effective in the reconstruction of the PINs at 2m compared to 0.3m scenario.



(a) *Urban noise and microphone placed at 0.3m*

(b) *Urban noise and microphone placed at 2m*

Fig. 7: Impact of noise source and SNR in the inference of 5-digit PINs within three attempts for PAD-1 and 500 digits per attacker.

7 Potential Countermeasures & Future Work

The relatively high accuracy of $\mathcal{P}inDrop$ highlights its danger and the importance of robust countermeasures. Barring wholesale replacement of PINs with other login means, we consider the following possibilities:

- *PIN Pad noise reduction*: This idea is simple, though challenging to deploy. It consists of masking the noise emitted by the PIN pad by covering it with soundproofing material. This approach could help in reducing the effectiveness of longer-range attack.
- *Noise emanation*: This countermeasure involves the emission of white noise by the ATM when entering the PIN. As shown in Section 6, high noise levels negatively affect attack performance.
- *On-screen PIN pad*: An effective countermeasure could be to virtualize the PIN pad using a touch screen. (This is in fact already done on some ATMs). This countermeasure would also allow dynamic rearrangement of digits, making it much more challenging to implement $\mathcal{P}inDrop$-like attacks. On the other hand, on-screen keypads are generally less user-friendly and can pose a problem for visually impaired users;
- *Feedback distortion*: If removing the characteristic sound emitted by each key is not possible, an alternative is to add noise that does not allow individual keys to be profiled. By emitting a masking sound at each key-press, $\mathcal{P}inDrop$ can be made more difficult, especially, its training phase;

- *Personal PIN pad*: Another possible countermeasure is to use a trusted device, such as a smartphone, to replace the physical PIN pad. The PIN could then be transmitted to the ATM using a wireless medium (e.g., NFC);
- *Behavioral biometrics layer*: An additional layer of security might be possibly via behavioral biometrics. One possibility is to involve user authentication based on keystroke dynamics. While this method can yield a high rate of false positives, it is completely transparent to the user (until or unless, a false positive occurs).

Possible future directions range from improving applicability of $\mathcal{P}inDrop$ to exploring its effectiveness on other kinds of PIN pads. An interesting direction might be to apply more sophisticated (e.g., parabolic) microphones. Such a microphone could significantly extend the effective recording distance of $\mathcal{P}inDrop$. Another direction is looking at $\mathcal{P}inDrop$ in the context of screen-based PIN pads that are fairly common on modern ATMs. This setting is more complicated due to lack of physical keys the sound of which can be profiled. However, it would be interesting to study whether sounds emitted by the touchscreen still allow the attacker to infer information about keys pressed. Finally, it would be interesting to evaluate $\mathcal{P}inDrop$ in a noisy real-world environment to assess the robustness of our approach, overcoming the actual experimental constraints.

8 Conclusions

This paper demonstrated $\mathcal{P}inDrop$, a highly accurate acoustic side-channel attack on PIN pads. It takes advantage of acoustic emanations produced by ATM users entering their PINs into the commodity ATM's metal PIN pads. These emanations can be surreptitiously recorded and used to accurately profile all PIN pad keys, allowing $\mathcal{P}inDrop$ to yield the victim's PIN with high probability. Specifically, this work shows that $\mathcal{P}inDrop$ is effective when applied from a very short (and perhaps not always realistic) distance away from the PIN pad (0.3m) as well as from a rather safe and inconspicuous distance (2m).

We demonstrated the effectiveness and robustness of $\mathcal{P}inDrop$ by conducting extensive experiments that involved a total of 58 participants and two commodities (commercially available) metal ATM PIN pads. We experimented with $\mathcal{P}inDrop$ in several configurations, showing how its performance can be optimized based on the training set size and the number of attackers.

$\mathcal{P}inDrop$'s accuracy reaches 93% and 95% in reconstructing 5-and 4-digit PINs, respectively, within three attempts, from 0.3 meters away. Also, at 2m away, $\mathcal{P}inDrop$ outperforms state-of-the-art results, reaching over 44% accuracy. This translates into an average accuracy improvement of 44% and 53% in 5-digit and 4-digit PINs, respectively. Finally, we proved that $\mathcal{P}inDrop$ is effective at 2 meters with low/moderate noise, reaching a lower-bound accuracy of 37%, while it remains effective under any noise condition at 0.3 meters. We believe that, due to its real-world applicability and performance, this work significantly advances the state-of-the-art in acoustic side-channel attacks.

References

1. Anand, S.A., Saxena, N.: Keyboard emanations in remote voice calls: Password leakage and noise (less) masking defenses. In: Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy. pp. 103–110 (2018)
2. Asonov, D., Agrawal, R.: Keyboard acoustic emanations. In: IEEE Symposium on Security and Privacy, 2004. Proceedings. 2004. pp. 3–11. IEEE (2004)
3. ATM Industry Association: , https://www.atmia.com
4. Bakowski, A., Radziszewski, L., Dekỳš, V., Świetlik, P.: Frequency analysis of urban traffic noise. In: 2019 20th International Carpathian Control Conference (ICCC). pp. 1–6. IEEE (2019)
5. Balagani, K., Cardaioli, M., Conti, M., Gasti, P., Georgiev, M., Gurtler, T., Lain, D., Miller, C., Molas, K., Samarin, N., et al.: Pilot: Password and pin information leakage from obfuscated typing videos. Journal of Computer Security **27**(4), 405–425 (2019)
6. Berger, Y., Wool, A., Yeredor, A.: Dictionary attacks using keyboard acoustic emanations. In: Proceedings of the 13th ACM conference on Computer and communications security. pp. 245–254 (2006)
7. Bond, M., Choudary, O., Murdoch, S.J., Skorobogatov, S., Anderson, R.: Chip and skim: cloning emv cards with the pre-play attack. In: 2014 IEEE Symposium on Security and Privacy. pp. 49–64. IEEE (2014)
8. Cardaioli, M., Conti, M., Balagani, K., Gasti, P.: Your pin sounds good! augmentation of pin guessing strategies via audio leakage. In: European Symposium on Research in Computer Security. pp. 720–735. Springer (2020)
9. Cecconello, S., Compagno, A., Conti, M., Lain, D., Tsudik, G.: Skype & type: Keyboard eavesdropping in voice-over-ip. ACM Transactions on Privacy and Security (TOPS) **22**(4), 1–34 (2019)
10. Halevi, T., Saxena, N.: A closer look at keyboard acoustic emanations: random passwords, typing styles and decoding techniques. In: Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security. pp. 89–90 (2012)
11. Halevi, T., Saxena, N.: Keyboard acoustic side channel attacks: exploring realistic and security-sensitive scenarios. International Journal of Information Security **14**(5), 443–456 (2015)
12. Hyosung, N.: cmax7600ta installation manual. http://www.tetralink.com/core/media/media.nl/id.46617/c.4970910/.f?h=d919934a85943438b8fe (2015), [Online; accessed 30-December-2020]
13. Liu, J., Wang, Y., Kar, G., Chen, Y., Yang, J., Gruteser, M.: Snooping keystrokes with mm-level audio ranging on a single phone. In: Proceedings of the 21st Annual International Conference on Mobile Computing and Networking. pp. 142–154 (2015)
14. Liu, X., Li, Y., Deng, R.H., Chang, B., Li, S.: When human cognitive modeling meets pins: User-independent inter-keystroke timing attacks. Computers & Security **80**, 90–107 (2019)
15. Logan, B., et al.: Mel frequency cepstral coefficients for music modeling. In: Ismir. vol. 270, pp. 1–11 (2000)
16. Martinasek, Z., Clupek, V., Trasy, K.: Acoustic attack on keyboard using spectrogram and neural network. In: 2015 38th International Conference on Telecommunications and Signal Processing (TSP). pp. 637–641. IEEE (2015)

17. Monaco, J.V.: Sok: Keylogging side channels. In: 2018 IEEE Symposium on Security and Privacy (SP). pp. 211–228. IEEE (2018)
18. Murdoch, S.J., Drimer, S., Anderson, R., Bond, M.: Chip and pin is broken. In: 2010 IEEE Symposium on Security and Privacy. pp. 433–446. IEEE (2010)
19. NationalCash Systems: ATM Statistics, http://www.nationalcash.com/statistics/
20. Panda, S., Liu, Y., Hancke, G.P., Qureshi, U.M.: Behavioral acoustic emanations: Attack and verification of pin entry using keypress sounds. Sensors **20**(11), 3015 (2020)
21. Rochat, J.L., Reiter, D.: Highway traffic noise. Acoust. Today **12**(4), 38 (2016)
22. Sean Kelly: Cell Phone Cameras Hidden Inside ATMs Cause Rise In Fraud (2018), https://www.opposingviews.com/category/cell-phone-cameras-hidden-inside-atms-cause-rise-fraud-throughout-britain
23. Sound and Video Understanding teams pursing Machine Perception research at Google: AudioSet: Traffic noise, roadway noise, https://research.google.com/audioset/dataset/traffic_noise_roadway_noise.html
24. de Souza Faria, G., Kim, H.Y.: Differential audio analysis: a new side-channel attack on pin pads. International Journal of Information Security **18**(1), 73–84 (2019)
25. United States Attorney's Office, District of Massachussets: Bulgarian National Pleads Guilty to ATM Skimming (2021), https://www.justice.gov/usao-ma/pr/bulgarian-national-pleads-guilty-atm-skimming
26. Wodo, W., Hanzlik, L.: Thermal imaging attacks on keypad security systems. In: SECRYPT. pp. 458–464 (2016)
27. Zhu, T., Ma, Q., Zhang, S., Liu, Y.: Context-free attacks using keyboard acoustic emanations. In: Proceedings of the 2014 ACM SIGSAC conference on computer and communications security. pp. 453–464 (2014)
28. Zhuang, L., Zhou, F., Tygar, J.D.: Keyboard acoustic emanations revisited. ACM Transactions on Information and System Security (TISSEC) **13**(1), 1–26 (2009)

Appendix

8.1 Validation Results

Table 2 reports the results on the validation set for four different ML models. Results show that LR and SVC obtain the best results on PAD-1 and PAD-2, respectively.

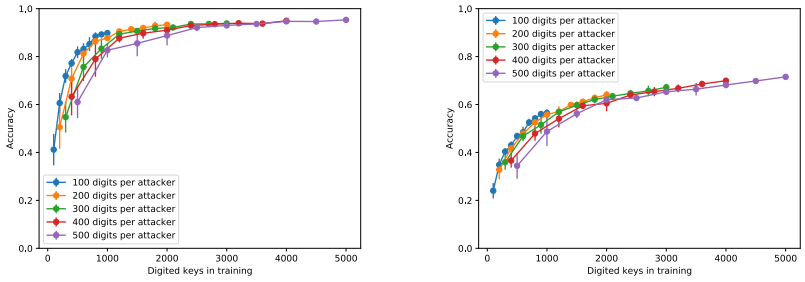
8.2 Additional Results

In Figure 8, we report the key accuracy results for PAD-2 (from both 0.3 m and 2 m). The results refer to the SVC model that achieved better performances on PAD-2.

In Figure 9, we report an example for the digit "3" for all the four scenarios. All the other keys show similar behavior, highlighting no significant inter-class differences. Interestingly, we note a different distribution of classification errors between *PAD-1* and *PAD-2*. In the first case, the error is uniformly distributed

	PAD-1		*PAD-2*	
	Distance 0.3 m	Distance 2 m	Distance 0.3 m	Distance 2 m
SVC	0.90±0.04	0.35±0.12	**0.86±0.06**	**0.21±0.07**
LR	**0.92±0.04**	**0.40±0.11**	0.85±0.06	0.19±0.04
KNN	0.65±0.07	0.13±0.07	0.17±0.05	0.02±0.01
RF	0.78±0.07	0.10±0.06	0.31±0.06	0.02±0.00

Table 2: PIN accuracies on the validation set for the investigated classifiers. The training set includes samples from five distinct attackers. The results show that for *PAD-1* the best performing model is the Logistic Regression (LR), while for *PAD-2* the best model is the SVC.



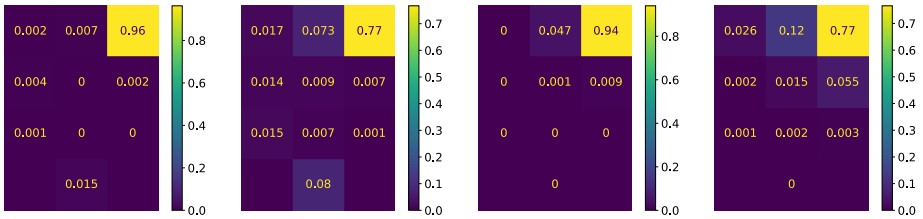
(a) *PAD-2, 0.3m* (b) *PAD-2, 2m*

Fig. 8: Key accuracy on the testing set for the best classifiers.

over all digits, in the second case, a higher concentration of errors is prominent around the true digit (i.e., digits 2, 5, and 6).

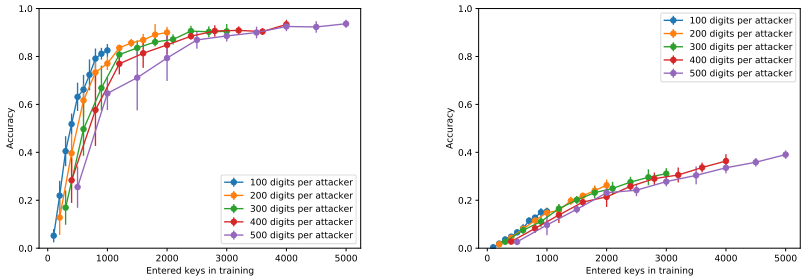
Figure 10 reports the PIN inference results within 3 attempts for PAD-2 and SVC model.

Figure 11 shows the results of $\mathcal{P}inDrop$ trained on the perturbed PAD-2 dataset (configuration 500 digits per attacker) in inferring 5-digit PINs within three attempts. The graphs report results similar to those obtained on PAD-1.



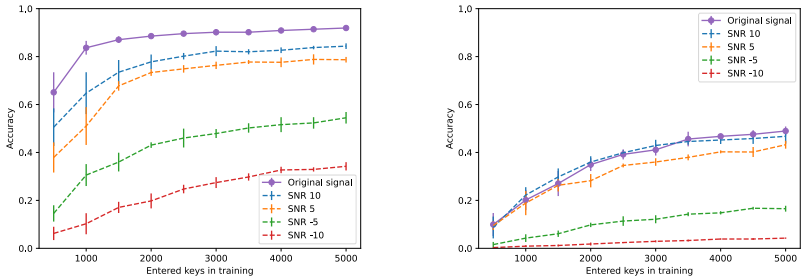
(a) *PAD-1 at 0.3m* (b) *PAD-1 at 2m* (c) *PAD-2 at 0.3m* (d) *PAD-2 at 2m*

Fig. 9: Digit "3" prediction heat maps for the four considered attack scenarios (the PIN pad layout is reported in Figure 3). We reported the results for the experiment with 5 attackers and 500 digits entered per attacker.



(a) *PAD-2 and microphone placed at 0.3m* (b) *PAD-2 and microphone placed at 2m*

Fig. 10: 5-digit PINs inference performance within 3 attempts for the best classifiers.



(a) *Gaussian noise and microphone placed at 0.3m* (b) *Gaussian noise and microphone placed at 2m*

Fig. 11: Impact of noise source and SNR in the inference of 5-digit PINs within three attempts for PAD-2 and 500 digits per attacker.