Heat of the Moment: Characterizing the Efficacy of Thermal Camera-Based Attacks

Keaton Mowery
UC San Diego

Sarah Meiklejohn
UC San Diego

Stefan Savage
UC San Diego

Abstract

In this paper, we examine the potential of using a thermal camera to recover codes typed into keypads in a variety of scenarios. This attack has the advantage over using a conventional camera that the codes do not need to be captured *while* they are being typed and can instead be recovered for a short period afterwards. To get the broadest sense of how effective such an attack might be, we consider a number of variables: the material of the keypad, the user entering the code, the distance from the camera to the keypad, and the possible methods used to analyze the data. First, we present code recovery results from human review of our test data set; this provides us with a baseline for the overall effectiveness of thermal camera-based attacks. Second, using techniques from computer vision we automatically extract the code from raw camera data, thus demonstrating that this attack has the potential to scale well in practice.

As we will see, both human and automated attacks are by and large successful in recovering the keys present in the code, even a full minute after they have been pressed; both methods are also able to determine the exact code (i.e., including the order in which the keys were pressed) for a smaller fraction of codes. Even without ordering, however, the search space of possible keys is still vastly reduced by knowing the keys pressed; for example, the search space is reduced from 10,000 possible codes to approximately 24 for a 4-digit code. In large-scale attacks involving many unique codes, such as on ATM PINs, our success rate indicates that an adversary can correctly recover enough codes to make such an attack economically viable.

1 Introduction

When deploying access control systems, designers must balance technical security issues with the practical concerns of cost and user acceptance. For these latter reasons, keypad-based systems — in which an authorized user is challenged to enter a secret numeric code — are extremely popular for applications ranging from bank ATMs to safes and car doors. Since such codes are typically entered in public view, however, they are open to optical capture, either via "shoulder surfing" [9, 1] or carefully placed cameras. Indeed, in recent years there have been a rash of attacks (typically focused on bank ATMs and gas stations) in which the user's payment card stripe is acquired via a "skimmer" while a pinhole camera is used to capture the associated PIN as it is entered [6]. In principle, the same style of attack could be used to defeat any keypad-based access control system.

Although optical attacks have proved to be quite effective in practice, they do have two clear limitations: they require real-time capture and they are vulnerable to occlusion. For example, if the user's body or hand blocks the camera's view of the keypad, then the attacker will be unable to recover the PIN code.[1] In 2005, however, Michał Zalewski [12] described how thermal imaging could be used to bypass both limitations. Demonstrating this technique using a keypad lock on an industrial safe, he found that body heat from the user transferred to the individual keys pressed and the resulting thermal residue could persist over long durations (between five and ten minutes, according to Zalewksi). Consequently, an attacker could potentially approach the keypad *after* the code was entered (and the legitimate user had left the scene) and recover the code by viewing the keypad with a thermal camera.

In this work, we examine the effectiveness of this attack in more depth and in regards to three separate considerations. First, we consider the impact of materials. Two types of keypads are commonly used in practice: plastic keypads (usually with rubber or rubberized keys), and metal keypads; we wondered if the material used would affect the results, and if so by how much. Next, we considered the role of individual differences between people using the keypads, such as variations in body heat and keypress "technique." In both of these cases, our underlying concern was to understand their impact on the window of vulnerability — how long an attacker can wait before approaching the keypad — since this ultimately determines the efficacy of the thermal capture approach. Finally, we also considered whether or not such an attack could be scaled efficiently. While a single high-value PIN (e.g., for a door or safe) may be recovered by visually inspecting thermal images, for a high-volume attack such as one against an ATM, we presume that an attacker would prefer not to classify hours of such footage by hand. We

[1] Indeed, a number of manufacturers now make keypad shrouds designed to occlude most view angles.

therefore consider whether the process of determining the code can be automated, and if so if it is more or less effective (e.g., accurate) than manual visual inspection.

For each of these underlying questions, we document that the answer is in fact "yes". In particular, we observed that the material of the keypad has a tremendous impact: the high thermal conductivity of metal keypads rendered them virtually impervious to the attack, while we obtained qualitatively similar results to Zalewksi using the plastic keypad (although in our measurements the thermal residue persisted for far shorter). Similarly, individual differences of the keypad operators plays a determining role as well. Some people were quite a bit more warm-blooded than others, and some were more forceful in pressing the keypad; for the people with colder hands or a lighter touch, the thermal results faded significantly more quickly. Finally, we developed an algorithm to completely automate the extraction of a code using a single post-hoc frame from the thermal camera footage, thus demonstrating that the attack has the potential to scale.

In summary, while we document that post-hoc thermal imaging attacks are feasible and automatable, we also find that the window of vulnerability is far more modest than some have feared and that there are simple countermeasures (i.e., deploying keypads with high thermal conductivity) that can shrink this vulnerability further still.

2 Attack Scenarios

As mentioned in the introduction, thermal cameras have a clear advantage over conventional cameras for the purposes of capturing codes: conventional cameras need to film the code as it is being typed, whereas thermal cameras can recover the code for some time afterwards. There are of course prevention methods that a user might in turn take against thermal camera-based attacks (for example, continuing to press the keypad even after he has entered the code, or simply resting his whole hand on the keypad); nevertheless, we expect that all but the most paranoid of users do not take them (at least not at present), and so the advantage over conventional cameras is still meaningful. We outline two main categories in which the advantage is most useful below, and also discuss the differences between the required attacks.

ATM PINs. When combined with a card skimmer, conventional cameras installed at ATMs have already proved to be quite effective in stealing people's account information. Using a thermal camera instead provides an attacker the ability to recover the code even in the cases where, for example, a user's body is blocking the keypad throughout the transaction, or he just covers the keypad with his hand as he types in the PIN. Attackers therefore gain an extra degree of flexibility in terms of camera placement, as it is no longer essential that the camera have an unobstructed view of the keypad at all times.

In an ATM scenario, one could easily imagine an attacker whose goal is to obtain as many PINs as possible. In this type of attack, an automated code extraction process would be highly beneficial; if the attacker simply installed the camera (and presumably a skimmer as well) and then used it to film the ATM keypad for a full day, using an automated process would save him the trouble of sifting through this entire day's worth of footage. In addition, the accuracy of the code extraction is not so essential in this scenario. Even if the attacker does not recover every single code, any non-trivial fraction of the PINs entered in a full day's worth of ATM usage would still be quite valuable.

Door codes. Doors (or gates or elevators) may act as access control points, in which entry to a given room, building, etc. is meant to be restricted to authorized users. Authorized users could share a special key or ID card, have their biometric data stored in the system for fingerprint or optical scans, or, in many cases, enter a password. In this last case, an attacker using a camera to capture an authorized user entering his code would be able to gain entry to the restricted area of his choice. Again, thermal cameras present a number of advantages here. Just as with the ATMs, users may block the keypad (either intentionally or unintentionally) in the process of entering the code, in which case an installed conventional camera would be rendered useless. Additionally, if the keypad is protected by a hood or shroud, an attacker would have trouble installing a conventional camera angled in such a way that the whole keypad could be seen. Thermal cameras, on the other hand, are able to overcome this problem; in fact, a thermal camera would not necessarily even need to be installed full-time. After an authorized user has entered the correct password, an attacker can simply walk up with the camera and film the keypad; provided he does this soon enough after the code has been entered (and that he knows no one will be walking by!), he can safely recover the code.

In many ways, this attack is quite different from the attack on ATMs. Beyond not even having to install a camera, the vast majority of users will be entering the same code (modulo frequent password changes); compared with ATM keypads then, in which each user enters a different code, the keypad for a door password will be much less noisy. On the other hand, the accuracy of the code extraction becomes essential here: if the attacker does not recover the one single correct code, then he has earned nothing; in other words, the "fraction" of the codes which he now needs to recover is just 1. In addition, if the attacker has to walk up with a camera every time he films the keypad then the automation of the attack becomes less relevant, as he might as well also look

Figure 1: The Dynasystems brushed metal keypad, model 00-101088-008B.

Figure 2: The Diebold plastic ATM keypad with rubber keys, model 19-019062-001M REV1.

at the footage.

3 Experiment Design

We can break our experiment design into two main categories: data collection and data analysis; we discuss them both here.

3.1 Data Collection

We used an A320 FLIR camera running at 9Hz with the built-in lens and the standard ExaminIR software for the camera (see [5] for the full camera specifications). The monthly rental rate for this camera is $1950 and the cost to buy is about $17,950. None of us had any expertise in the area of thermal cameras before this project began; in fact, we were all completely new to both the hardware and software involved.

For the keypads, we purchased one that was brushed metal and one that was plastic with rubber keys. Both were purchased from eBay and were (at least according to the sellers) used in real ATMs. The metal keypad can be seen in Figure 1 and the plastic one in Figure 2.

To conduct the experiments, we first placed the keypads in a vise to allow users to press the keypad without having to steady it with their hand. We then placed the camera on a tripod, first at a distance of 14 inches and then at a distance of 28 inches from the keypad; as we will see in the next section, this difference in distance had little effect on our results, and so we posit that the camera would have to be moved quite a lot further away before results began to degrade. For each distance, we had 21 people press 27 different codes chosen at random; seven of the codes contained at least one duplicate (e.g., 2227 or 0510) and the other twenty contained four unique digits. Everyone pressed the keys in a way such that while they were pressing the buttons their hand and arm almost completely obscured the keypad in the frame, although no one attempted to shield the keypad with their other hand. As mentioned in the introduction, we found in earlier trials that people reacted with the keypad in very different ways: some had a light touch while others were almost forceful in pressing the keys; similarly, some people were very warm-blooded while others transferred barely any heat to the keypad. We therefore chose to use such a wide variety of testers as a way of eliminating any of these potential human biases. We also ran the full set of tests on the plastic keypad only; as mentioned in the introduction, even filming the metal keypad was problematic and so we performed only a few runs on it.

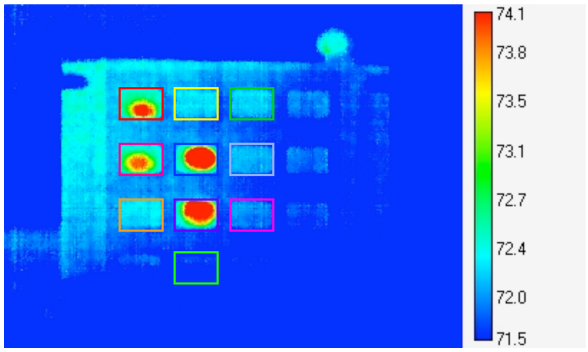

Figure 3: A frame as captured by the camera on one of our runs, with the ten regions of interest indicated by the colored boxes and the temperatures shown on the right. This is the image captured immediately after the hand no longer obscured the keypad from view; we can clearly see that the four digits pressed were 1, 4, 5, and 8, and furthermore that the 1 and 4 were likely pressed before the 5 and 8 (the real code was in fact 1485).

To begin collecting our data, we first focused the camera on the keypad and used the ExaminIR software to indicate the ten regions corresponding to each of the ten keys in which we were interested. For each run, we recorded the keypad for approximately 10 frames (or 3 seconds) before the user entered the assigned code, and then for 350 frames (or 100 seconds) after; a longer calibration period would likely result in better accuracy, as it would eliminate a fair amount of the noise we observed in our results. Sample stills as captured by the camera can be seen in Figures 3 and 4.

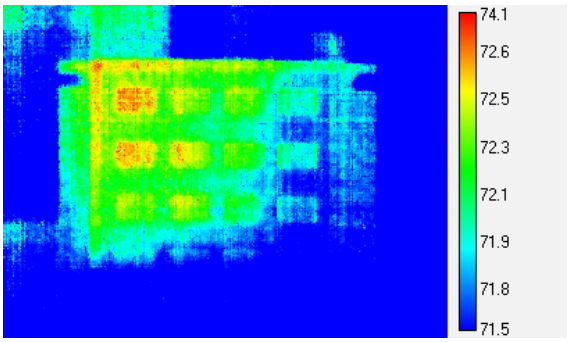

Figure 4: Another frame captured, from the same run as Figure 3, only this time 315 frames later (or 90 seconds) and without the boxes drawn on. We can see that the 1, 4, and 5 digits are still the hottest keys, but that the 8 has faded to the point where it is not clearly recognizable as one of the keys pressed.

3.2 Data Analysis

To analyze the data, we broke the footage up into three parts corresponding to before, during, and after the hand was in the frame (i.e., the code was being entered). This phase was done automatically: we wrote an algorithm to determine when a large warm object (i.e., a hand) was in the frame, and then broke the footage up accordingly. The part before the hand was considered a calibration period (which, as mentioned, lasted approximately 3 seconds), while the part after was our main focus. By splitting the footage up in this way, we treat each run as if the user properly shielded his code entry from view, thus embodying the advantages of thermal camera-based attacks (as we simply ignore any footage taken during the code entry itself).

3.2.1 Manual Review

To establish a performance baseline, we manually reviewed every tenth frame and attempted to recover the input code through inspection. By presenting the frames in a random order, we minimized the effects of pre-existing knowledge and allowed each frame to be considered on its own. Note that extracting the code from each frame individually is strictly harder than from a video where patterns can be seen; better results might be possible by observing temperature changes over time.

These results represent the efficacy of the easiest (but most labor-intensive) attack possible: direct human intervention. To alleviate the work this presents an attacker, we next consider automated code retrieval.

3.2.2 Automated Review

For each code entry, we focused our attention on 10 predetermined regions of a video frame, corresponding to the location of each button on the keypad (as demonstrated by the boxes in Figure 3). Each individual camera and keypad setup has its own particular regions, but they are fixed as long as the camera stays in place.

We then examined the regions of interest (ROIs) during the last frame of video before hand entry. This frame was treated as a calibration frame, recording the state of the keypad before user interaction. For each frame after code entry, we compared each region against this calibration frame in order to deduce button presses and order. Surprisingly, in our tests, examining multiple frames together provided similar accuracy to this per-frame approach, and we do not present results of full-video analysis here.

To do this frame-by-frame comparison, we applied one of three possible methods: *max*, *mean*, *binarize*. As we will see in the next section, the mean method far outperformed both the max and binarize methods.

- **Max.** With the max method, each ROI was represented by the maximum temperature recorded within the region. We found that, while this did pull out high temperatures caused by user warmth, it also fell prey to higher noise in blank regions.

- **Mean.** With the mean method, each ROI was represented as the arithmetic mean of its temperatures. This method did a far better job than the max method of smoothing out noise; still, we observed that old or light keypresses sometimes provided a very small movement in the mean. Overall, however, this method achieved the best performance out of the three.

- **Binarize.** Finally, for the binarize method, we compared each pixel against its counterpart in the calibration frame. If its temperature increased, we treated it as a 1; otherwise, as a 0. By then applying the mean method to the binarized image, we can gain a measure of how much of the ROI has increased in temperature.

Regardless of the method used to characterize ROIs, we then assigned each ROI a weight by subtracting the results for the calibration frame from the results in the live frame. For codes with four unique buttons, we're done: we can simply sort by weight and choose the top four ROIs in ascending order. Intuitively, the weight represents the temperature difference for the button, and the last button pressed theoretically has the highest delta.

To support repeated buttons, however, we added in one final tweak. Ideally, the weight of unpressed buttons should cluster around 0, regardless of which method is used. Keeping this in mind, we therefore averaged the weights of all ROIs, and considered only those whose weight exceeded the group average to have been pressed.

4 Performance Evaluation

In this section, we examine the success with which our human reviewer and automated system recovered complete or partial information about entered codes. Using our recorded videos, we applied the techniques described in the previous section to each frame to determine which buttons had been pressed. We then consider how effective these results might be for the various attacks described in Section 2.

4.1 Results

Using the experiments and data analysis methods outlined in the previous section, we were able to determine how accurate both a human reviewer and our automated system were in terms of recovering the keys pressed (Figures 5, 6, and 7), the code entered (Figure 8), and the code entered with a possible mistake made (Figure 9).

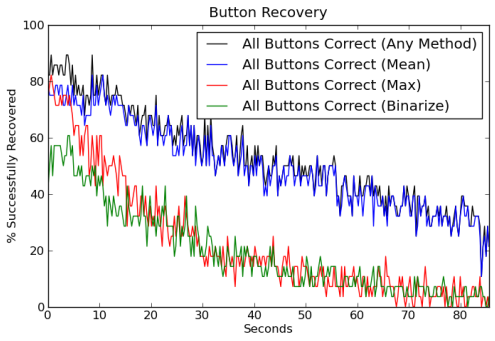

Figure 5: Percentage of perfect key combinations (i.e., without ordering but with all the keys correct) determined as a function of time when the camera was placed at 14 inches from the keypad; these results were obtained using our algorithm approaches of max, mean, and binarize (see the previous section for a reminder on what these are).

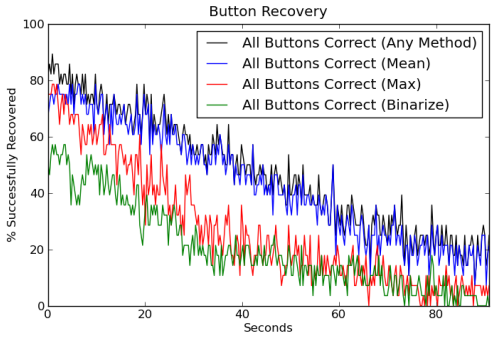

Figure 6: Percentage of perfect key combinations determined as a function of time when the camera was placed at 28 inches from the keypad; again, these results were obtained using our three algorithms.

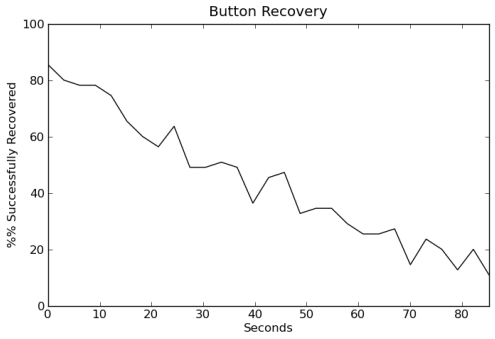

Figure 7: Percentage of perfect key combinations, this time for all 54 runs at both distances, determined as a function of time. The results here were obtained through human visual inspection.

As we can see in Figures 5 and 6, the distance from the keypad yielded almost no difference in key accuracy (and in fact the max method even did a tiny bit better when the camera was further), indicating that we probably could move the camera further away while still achieving the same results. In addition, the lens we were using was the cheapest lens available (albeit not particularly cheap!) and so it is likely that these results would have persisted for at least a few more feet with either this or a more performant lens. We can also see, comparing these figures with Figure 7, that the accuracy of our automated methods was slightly higher than that of our human reviewer, especially as more time passed; after a minute, the automated method could still recover approximately 50% of the codes, compared to the recovery rate of only 20-30% from visual inspection.

Looking now at Figures 8 and 9, we can see that both our human reviewer and our automated system were much more successful at recovering the individual keys pressed than the exact in-order code, even when allowing for possible mistakes in the order. We therefore do not expect that further work on individual frame-based detection systems will yield much more promising numbers, but suggest that novel approaches to whole-video analysis might present an avenue for improvement.

4.2 Effectiveness of our results

After obtaining our results, we were led to wonder just how effective they would be in a real-world attack (such as those described in Section 2). As we saw in our figures, we consider three possible versions of "success": recovering the exact code, recovering the exact code but with a potential mistake (either a transposition or an incorrect digit), and recovering the digits of the code without any ordering information.

Recovering the perfect code is clearly the most desirable outcome: in this case, an attacker attempting to test

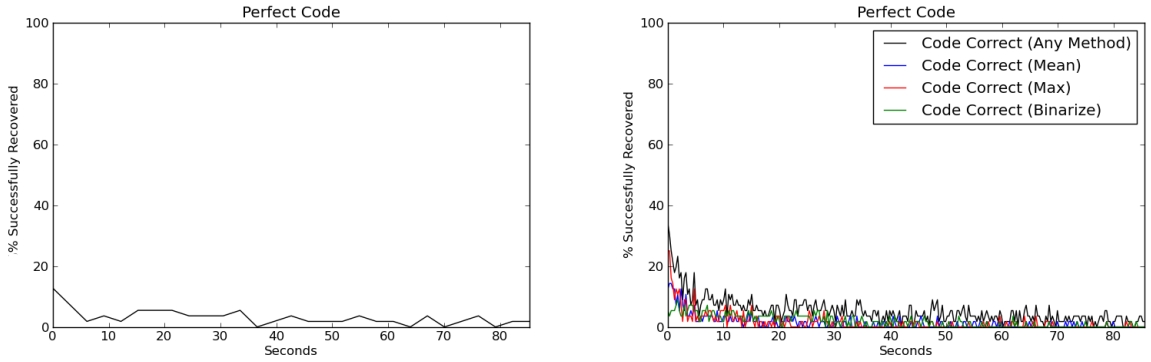

Figure 8: A comparison of the percentage of perfect codes we able to recover over all 54 runs, as a function of time, using both human inspection (the graph on the left) and our various algorithmic approaches (the graph on the right). We can see that neither is particularly successful, but that the automated approach does slightly outperform the human inspection.

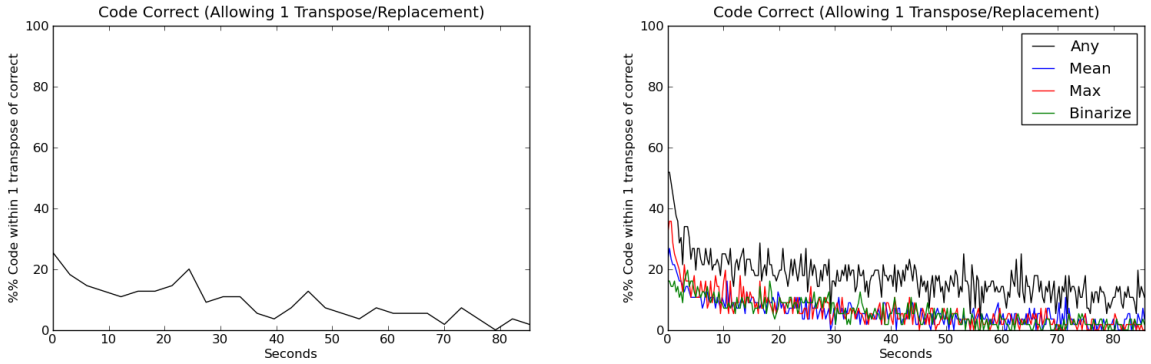

Figure 9: A comparison of the percentage of perfect codes we able to recover over all 54 runs, as a function of time, this time allowing for one transposition (e.g., 1845 instead of 1485) or incorrect key (e.g., 1985 instead of 1485). Again, we can see the results from human inspection on the left, and the results from our algorithmic approaches on the right.

out different codes would need to test only a single code. In the case in which the ordering might be slightly off, we can consider the probability an attacker will have of guessing the exact code. For a 4-digit code, if the code given by the algorithm has two digits transposed, then there are 6 possible choices for the correct code (just counting the possible transpositions), and so an attacker will guess the correct code on his first try with probability $1/6$. If a digit is incorrect, however, then he might have to try each possible digit within each position (while keeping the other values fixed) before getting the correct code. If he knows which position is incorrect (for example, if the first digit pressed has cooled but the other ones still have a fair amount of heat residue left), then he has a $1/9$ chance of success; if he doesn't, then he might have to try each position in turn and therefore has a $1/36$ chance. We also mention that these success probabilities assume that the attacker actually knows which case he is dealing with; while this might seem unlikely, simple visual inspection might often be enough to give him a good idea (for example, if four keys are clearly pressed then he can assume a transposition error has occurred rather than an incorrect digit).

In addition to recovering the ordered code, we also consider the success probabilities in the case when only the keys pressed are recovered. Here, for a 4-digit code there are $4! = 24$ possible codes given no ordering information, and so an attacker will have probability $1/24$ in guessing the correct code on the first try. If only three keys were used (e.g., the code was 6268) and the attacker doesn't know which key was repeated then the success probability goes down to $1/36$; if only two keys were used, on the other hand, then his success probability goes up to $1/14$ (and of course if there is only one key pressed the success probability is 1). In the case that the code is longer, this result quickly becomes vastly less useful; if the code is, for example, 6 digits, then recovering no ordering information will give the attacker $6! = 720$ possible codes to try. As compared to the only 15 possible codes if the result contains a transposition error, we can

see that this approach degrades far more quickly than the perfect codes with a slight mistake.

Finally, recall that many access control systems, to prevent code guessing attacks, disallow access entirely after a small number of incorrect guesses. Naturally, recovering the precise code is therefore the optimal outcome of our attack. With larger data sets in which the attacker can move on to other, unlocked accounts, however, the simple recovery of keys pressed can still be considered a successful attack, as it will likely allow an attacker access to some non-negligible fraction of accounts.

5 Related work

In a broad sense, our work can be thought of as an attack taking advantage of persistent side-channel information (i.e., side channels that remain open even with the passing of time), in our case the heat residue left on a keypad after a code has been entered. The recent work of Aviv et al. [3] also falls into the category: in their paper, the authors identify smudges left by users entering their password on Android smartphones as a potential side channel for attacks; as these smudges persist even after standard usage of the phone (e.g., making a call), an attacker who steals the phone will be fairly successful in also recovering the code, thus granting him complete access to all the personal information stored on it.

More closely, our work is related to a study done by Michał Zalewski [12] on the use of thermal cameras to perform safecracking; to the best of our knowledge, he is the only other person to do any research in this area. As outlined in the introduction, our work aims to complement his by considering a wider spectrum of attack scenarios and demonstrating the relative effectiveness in each possible setting.

Finally, we mention that our attack falls into a family of side-channel attacks in which the theft of private information cannot be detected, at least not by the victim, until after the damage has been done (i.e., the information has already been used). In the ATM scenario, we exhibit a type of attack in which customer's bank card information is stolen; in particular, we identity a possible "PIN compromise" type of attack [9, 4]. Similar attacks include replicating physical door keys from high-resolution photographs [8], keylogging computer keyboard input with just the sound of the keyboard [2, 13], using electromagnetic emanations from a computer monitor to recover screen contents [7, 11], detecting CPU operation patterns from high-frequency acoustic noise [10], and many more.

6 Conclusions and Future Work

In this paper, we have demonstrated a thermal camera-based attack against keypad code entry that is easily scalable and, in many scenarios, quite effective: even a minute after the keypad was pressed, we were still able to recover over half of the entered codes. We further demonstrated, by comparison to a human reviewer, that our automated approach rivals visual inspection in terms of accuracy (and in some cases even outperforms it), in addition to offering a clear advantage in large-scale attacks such as those on ATMs.

In addition to these positive results, we also found that in some settings our attack was significantly less effective than in others. For users with a light touch or a low body temperature, for example, the heat residue degraded significantly more quickly and we were able to obtain very little information, even after a matter of seconds. The material of the keypad also made a huge difference: against metal keypads, the few runs that we did perform were almost completely abortive. Much of this can be attributed to the high conductivity of the metal, which meant that the heat residue remained localized to the key that had been pressed for only a few seconds; we also observed, however, that either the keypad itself or a paint put on the keypad caused it to act as a thermal mirror, meaning it was hard to even get a clear reading on the keypad at all. Therefore, at least based on our current results, the obvious approach to prevent our (and essentially any thermal-camera-based) attack would be to use metal keypads exclusively.

For future work, it would of course be possible to explore more broadly the spectrum of possible attacks; for example, determining how much environmental temperature plays a role (all our tests were done in one air-conditioned room) or how far away a camera could be before results began to seriously degrade. In terms of increasing the effectiveness of our attacks, one clear goal would be to obtain any results whatsoever from metal keypads; more generally, we could hope to have a higher success rate in recovering the exact code and to have our success in recovering the keys pressed last for even longer than it already does. We leave these all as interesting open problems.

References

[1] ANDERSON, R. Why cryptosystems fail. In *Proceedings of CCS 1993* (1993), pp. 215–227.

[2] ASONOV, D., AND AGRAWAL, R. Keyboard acoustic emanations. In *Proceedings of the 2004 IEEE Symposium on Security and Privacy* (2004), pp. 3–11.

[3] AVIV, A., GIBSON, K., MOSSOP, E., BLAZE, M., AND SMITH, J. Smudge attacks on smartphone touch screens. In *Proceedings of WOOT 2010* (2010).

[4] ENISA. Atm crime: overview of the European situation and golden rules on how to avoid it, 2009. `http://www.enisa.europe.eu/media/press-releases/enisa-warn`.

[5] FLIR. A320 camera specifications. `http://www.flira320.com/PDF/datasheet_thermal_imaging_camera_flirA320.pdf`.

[6] KREBS, B. ATM skimmers: hacking the cash machine, 2011. http://krebsonsecurity.com/tag/atm-skimmer.

[7] KUHN, M. Electromagnetic eavesdropping risks of flat-panel displays. In *Proceedings of the 4th Workshop on Privacy Enhancing Technologies (PET)* (2004), pp. 88–106.

[8] LAXTON, B., WANG, K., AND SAVAGE, S. Reconsidering physical key secrecy: teleduplication via optical decoding. In *Proceedings of CCS 2008* (2008), pp. 469–478.

[9] RUSSELL, D. Atm crime, ATM fraud overview. http://ezinearticles.com/?ATM-Crime,-ATM-Fraud-Overview.

[10] SHAMIR, A., AND TROMER, E. Acoustic crypanalysis: On nosy people and noisy machines. http://tau.ac.il/~tromer/acoustic/.

[11] VAN ECK, W. Electromagnetic radiation from video display units: An eavesdropping risk? *Computers & Security 4*, 4 (1985), 269–286.

[12] ZALEWSKI, M. Cracking safes with thermal imaging, 2005. http://lcamtuf.coredump.cx/tsafe.

[13] ZHUANG, L., ZHOU, F., AND TYGAR, J. Keyboard acoustic emanations revisited. In *ACM Transactions on Information and System Security* (2009), vol. 13.