Swipe Your Fingerprints! How Biometric Authentication Simplifies Payment, Access and Identity Fraud

Julian Fietkau
jfietkau@sect.tu-berlin.de

Starbug
starbug@berlin.ccc.de

Jean-Pierre Seifert
jpseifert@sect.tu-berlin.de

Security in Telecommunications
Technische Universität Berlin

Abstract

Biometric authentication is a trending topic in securing modern devices. Examples of this can be found in many widely deployed systems such as Apple's Touch ID or Microsoft's Windows Hello face recognition. Miniaturization and increased processing power are thereby leading to new applications not imaginable a couple of years ago. Such a solution is the new fingerprint smart card built by a Norwegian company that must not be named. Their biometric match-on-card platform is designed to provide a convenient solution for access, identity, and payment applications and aims to replace PIN authentication for the next generation of payment cards by VISA and Mastercard. In this paper, we are going to analyze how this company has implemented their already available demo kit for access control in hardware and software. We will point out critical weaknesses in its architecture and algorithm and show how these could be misused for payment, access and identity fraud by attackers able to steal or clone the device. Thereby, we combine software and hardware hacking techniques as well as extraction methods, to acquire fingerprints from photos and latent prints, to successfully spoof the system in various ways. This works in particular without the error-prone creation of physical dummies due to the exploitation of the insecure on-device communication. The attacks presented require little effort and low-cost equipment that can be already refinanced by abusing a single card at all. Finally, we will discuss countermeasures and ideas to improve the security of this and future implementations for match-on-card fingerprint authentication.

1 Introduction

Biometric authentication is evolving to one of the most used authentication schemes for mobile applications. That's why, several research groups and companies try to find new ways to make use of characteristics in users iris, fingerprints or heartbeat. Based on this, they want to provide a convenient authentication scheme to protect private and sensitive data, stored for instance on mobile devices. Despite the fact that no sufficient secure and reliable method has been developed for low-cost biometric authentication, large manufacturers yet integrate these solutions into devices and promote them as an improvement in security and comfort. A good example of this is a Norwegian company that must not be named. They try to integrate their fingerprint match-on-card platform into several devices to simplify payment, access and identity applications. So far, only the first version of access control cards is available, but they already announced to integrate their platform into the next generation of payment and ID cards. According to [16, 24], the first payment card, build in cooperation with VISA and Mastercard, is already under test.

In contrast to these efforts, many publications like [5, 22, 6, 7], clearly show that biometric authentication mechanisms can be levered, manipulated and circumvented in various ways. The reasons for this are many and varied. On the one hand, developers choose weak acceptance rates to offer users a convenient and reliable solution. On the other hand, mechanisms for liveness- or spoof detection are often dispensed with, because they are expensive, immature or inadequate. Another major problem is in fact that biometric features are unique and cannot be revoked or replaced like PINs and passwords. Moreover, biometric features do not even represent good secrets at all, because users tend to spread sensitive information on every object touched or copy them unintentionally by creating image and video data containing fingerprints and iris [22].

Taking all this into account, one can not simply trust this technology; instead these systems need to be scrutinized to identify the accompanying threats and risks. For this reason, we analyzed this new match-on-card device to explore its foundation and answer the question of whether this approach is a security improvement or not.

Our Contribution is the proposal of a low-cost attack against a biometric authentication device based on a new match-on-card platform. The attack uses a small hardware modification and some information gathering to bypass the authentication entirely and as often as needed. Related to that, we explore what kind of weaknesses in hardware and software can be abused to make this happen. To evaluate our findings, we provide all data and knowledge acquired during exploration of this fingerprint matching device.

2 BACKGROUND

In the following, we provide the required background of biometric authentication and fingerprint matching. We then describe what kind of attacks are known to circumvent fingerprint matching systems and how to create a physical dummy for traditional spoofing attacks.

2.1 Biometric Authentication

The term *biometric* is derived from the two Greek words *bios* and *metron* meaning measurement of life and describes the identification of people based on features of their bodies. Those features can be divided into static (like fingerprint or face) and behavioral (like a persons voice or signature). According to [14], all biometric features have to satisfy the following requirements:

- **Universality:** All people possess the feature.
- **Uniqueness:** The feature is different for people so the system can distinguish between them.
- **Permanence:** The feature only varies slightly over time.
- **Measurability:** The system can acquire and process the feature in an efficient way.
- **Safe against circumvention:** The system can distinguish between the real feature and a dummy.

Any biometric system requires thereby three main components: A sensor that captures the feature, a biometric application to compute and compare features, and a database to store a template [14]. A template is a mathematical description of the biometric feature. It is generated during the enrollment, the first stage of every biometric process in which a user is introduced to the system. During the later use, the live taken template will be compared against the stored template. Biometric comparisons, other than passwords, do not result in a clear right or wrong answer but in a probability of a matching score. Depending on the determined thresholds this matching score will lead to an accept or reject. Usually, biometric systems apply this in two modus operandi [14]:

- **Verification:** During verification or 1:1 match the user provides its template bound ID to the system. The system then compares the live taken feature to that template. If the threshold is matched, the user is authenticated successfully.
- **Identification:** Identification or the 1:n match compares the live taken feature against all the templates stored in the database. As a result, the template with the highest match score is selected, and if the threshold is exceeded, a successful match is indicated.

2.2 Fingerprint Authentication

Fingerprint authentication is the process of matching fingers based on the structure of the upper skin. During the prenatal development of a human being, the skin of hands and feet fold in a random process leading to ridges and valleys. Those end or bifurcate at certain points called minutiae. Although some genetic influence, the position, orientation and type of these minutiae points are unique for each human and even each finger. To recognize someone based on that, we first of all need to acquire the features. As summarized by [14], various technologies have been used to sample the fingerprint. Today, the most popular ones are the optical and capacitive sensors.

- **Optical:** Optical sensors use the effect of frustrated total reflection. Light is shined on a prism and the reflections are collected by a camera. Depending on whether a ridge or valley is touching the prism, the light is either reflected or scattered creating an image of the fingerprint.
- **Capacitive:** An array of single capacitor plates are exposed to the sensor surface. The capacitance of each plate depends on the material above. This way, one can distinguish between the skin of the ridges and the air in the valleys of a fingerprint.
- **Ultrasonic:** Ultrasonic waves are sent into the finger, get reflected on deeper layers of the skin and will be collected by the sensor. This even works well for dirty fingers and dry skin.
- **Thermal:** Thermal sensors measure the heat distribution and derive the fingerprint image from the different thermal properties of valleys and ridges.
- **Pressure:** Pressure based fingerprint acquisition works with an array of small sensors able to measure the pressure of ridges contacting it.

After sampling a fingerprint, the created image will usually be graphically preprocessed to reduce noise and errors, for example by thinning the ridges to a width of one pixel. Following that, a mathematical description will

be derived from this intermediate image to make it comparable and easy to store. The most popular matching techniques are based on image correlation, phase matching, skeleton matching or minutiae matching [13]. Due to its storage efficiency and accuracy, the minutiae-based matching is the most popular approach and has long since proven itself in criminal investigations and forensic applications.

To extract minutiae points, an algorithm travels along all ridges until these end or bifurcate. For those points, their relative position, orientation, and minutiae type will be stored. This can be done by creating a template using a standard format as defined in ISO/IEC 19794-2 [12]. When verifying the user, the algorithm applies the same method to the live captured fingerprint and compares this data with the stored template created during enrollment.

This matching process is generally a sophisticated pattern-recognition problem, because of the large intra-class variations due to pressure, rotation, translation and many physiologic conditions like skin dryness or cuts [13]. Additionally, there is a large interclass similarity between fingerprint images from different fingers, because there are only three types of major fingerprint patterns in particular: arch, loop, and whorl [13]. Within this error space, we have to carefully align both templates to initiate the verification. One basic approach is to align the fingerprints based on some random local minutiae structures and then consolidate the local matching on a global scale. This procedure usually involves four steps, as described in [13]:

1. Compute pairwise similarity between minutiae of two fingerprints by comparing the invariant minutiae descriptors.

2. Both fingerprints are being aligned according to the most similar pair of minutiae.

3. Search for minutiae pairs that are close enough in position and direction to be matching pairs.

4. Calculate the similarity score between both fingerprints based on the number of matching pairs, consistency of ridge count in between, et cetera.

2.3 Known Attacks and Related Work

As stated in [11], the main threat to any assets protected by a biometric system is that of an impostor impersonating another person who is enrolled and gaining access to the protected assets. When successful, the authentication mechanism is considered broken. A structured overview covering a threat model and risk evaluation for fingerprint matching systems is discussed in [11] and [26]. The threats discussed by [11], include four major scenarios to attack a matching system, namely: use of a dummy, use of latent prints, use of a biometric lookalike and use of the real finger of the victim.

Much more practical research results can be found in [22] and [6]. The author presents and analyzes real methods to spoof and trick specific implementations of fingerprint matching algorithms. A very prominent way to do this is to collect and physically clone the biometric features with fingerprint fuming or by extracting the features from photos of the victim. We will describe these techniques in more detail in Section 2.3.1.

Another attack scheme, based on a mathematic analysis of a large number of fingerprints, is presented in [1]. This work shows that so-called ”MasterPrints” can be synthesized based on similarities in different fingerprints, which can impersonate users with a given probability. This technique works best for systems using multiple partial fingerprints of a user to enroll, which is quite common today. While this attack was shown in theory and validated using a commercial fingerprint verification software (Verifinger 6.1), no real authentication system was bypassed. Moreover, the ”MasterPrints” are intended to impose a subset of users with a given probability and do not allow to target a specific person. The MasterPrints are not published and can not be tested.

More related work is presented in [4]. The authors point out the risk of known template attacks for minutiae-based matching algorithms. The overall problem is that sensitive data, stored on the device or in large databases, might be leaked in one way or another. Related to this, the paper presents a method to create sophisticated and natural-looking fingerprints only from the numerical template data. They successfully evaluate this approach against a number of undisclosed state-of-the-art algorithms and the NIST Fingerprint Image Software [25].

2.3.1 Circumvention with Physical Dummies

The circumvention of fingerprint systems with dummies consists of two parts, the collection of a high-quality image of the finger and the creation of the dummy itself. Since the skin is producing sweat and grease, we consistently leave latent prints on all surfaces we touch. Those prints are an exact copy of the ridges and valleys that form our fingerprint. When making these residues visible, they can be used to produce a dummy. There are different techniques known from police work to enhance latent prints on different surfaces. For glossy surfaces, like displays, latent prints are visible and can be digitalized easily using a scanner or camera [23]. To increase the contrast, the light of a specific wavelength and different angles can be projected onto the print or colored powder can be applied. On physical contact, there is always a chance of damaging the print. That’s why the contactless technique using cyanoacrylate vapor is often

used nowadays. Cyanoacrylate, the main ingredient of super glue, is put into a small chamber covering the area of the print. The vapor reacts with the grease leaving a solid white fingerprint. The grease from the print will not only stick onto glossy surfaces but also will be absorbed by the paper. To make use of this an amino acid indicator like Ninhydrin can be applied. This reacts with the amino acids in the latent print turning it purple. The result can be digitalized using a scanner or a camera and enhanced with an image editing application to reduce noise and scale it properly. Depending on the application the image must be inverted and/or mirrored. The final result is then transferred to a photodefinable PCB that acts as a mold. The dummy material, e.g. wood glue, is then poured into the mold and cured. To enhance the electrical properties graphite spray can be applied. After some time, the dummy finger can then be used to fool the sensor. As long as there are no fake detection mechanisms in place, the system will not be able to distinguish between the dummy and a living finger. [6]

3 Analysis and Weaknesses

In this section, we describe how the access control card is used and how it is implemented. We also present the findings gathered by exploring the device components.

3.1 Implementation Overview

First of all, we will have a look at how the card works. As documented in [28], the system implements the principles from Section 2.2 in a straightforward way. First of all, the card owner needs to enroll to the card in a trusted environment. During this step, the fingerprint is captured 10 times and the extracted minutiae data is stored on the card according to the ISO/IEC 19794-2 standard [12]. Afterward, the activated card can be used like any other smart card. The owner inserts the card into the terminal or near the NFC field and has to identify himself to authorize the desired action. Without entering a PIN, the user places his finger on the built-in sensor to prove its identity. Then the system extracts the minutiae points from the finger and compares it with the stored template. When successful, the RFID functionality of the card will be enabled, authorizing the card to communicate with the reader and execute the following steps. In addition, the smart card will also support 3-factor authentication. When enabled, the user needs to provide the card, his fingerprint, and the PIN.

To perform all these functions the card integrates various hardware components. For RFID capabilities they offer multiple card versions with NFC Transponders like Mifare Classic, DESFire EV1, and others. These are compatible with most contactless ISO14443 RF readers

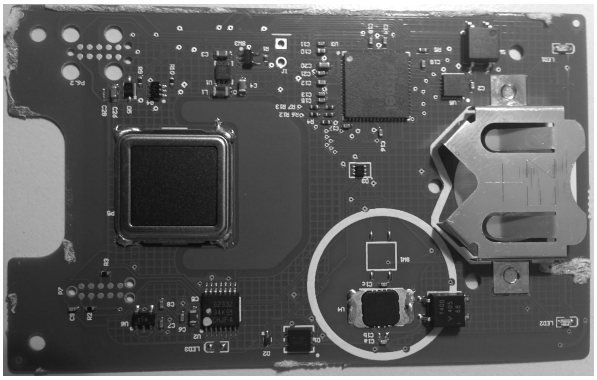

Figure 1: Frontside of the uncased access control card.

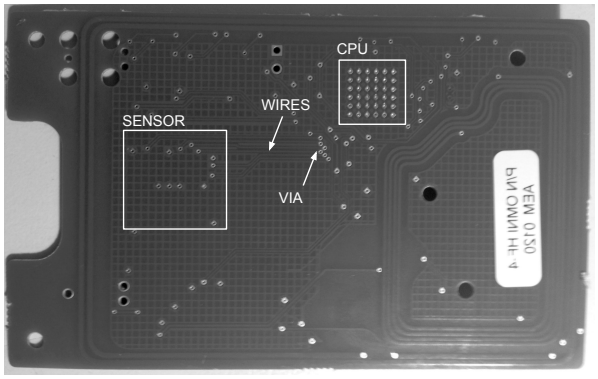

Figure 2: Backside of the uncased access control card.

and feature RF-field energy harvesting. The most innovative part is the integrated FPC1020 Touch Fingerprint Sensor manufactured by Fingerprints [8]. It is build using a capacitive array, has a size of 11×11mm and a spatial dot density of 508dpi. This way it can generate 8bit grayscale images with a dimension of 192x192 pixel. Moreover, there are two LEDs on the card to indicate the systems state and the result of enrollment and verification attempts. Sensor, LEDs and the RFID-subsystem are connected to an Atmel SAM4S microcontroller, which is based on a 32-bit ARM Cortex-M4 processor. This microcontroller executes a proprietary matching algorithm and supports procedures for enrollment and verification. The algorithm is optimized for embedded devices and can process a single evaluation in round about 500 to 550μs. The biometric template is stored on the internal memory of the microcontroller and will be encrypted. To protect confidential data and firmware the SAM4S prevents memory access by using security and lock bits to denial access via ICE and Flash programming interfaces. According to the SAM4S manual, the external bus interface is scrambled and memory integrity checks are implemented as well [8]. The announced payment and ID cards will be produced in ISO/IEC 7810 ID-1 card for-

mat. The access control card is slightly larger, packed in a plastic cover and requires an extra battery.

3.2 A Closer Look

To analyze the card we, first of all, removed the plastic cover revealing a two-layer PCB with all the components mounted on the upper side. After identifying the responsible pins, we initially confirmed the correct configuration of the SAM4S security bits using a JTAG debugger. We then started to analyze the application specific components of the device. As one can see in Figure 1 and 2, the fingerprint sensor and the microcontroller are connected with six wires, routed on both sides of the card. To sniff the communication between sensor and microcontroller we tapped the connection using a logic analyzer and soldering enameled wires to the VIAs. We have used the Saleae logic analyzer [20] and their software suite for a total of $109. After evaluating several recorded tracks, we were able to identify the implemented communication bus: A Serial Peripheral Interface (SPI) configured with a 12 MHz clock signal and transfer Mode 0. SPI is based on a master-slave architecture. In this case, the microcontroller represents the master and the fingerprint sensor the slave. This way the CPU can request data from the sensor that can be selected by sending special commands to the sensor.

3.3 Communication Protocol

Using the recorded tracks, we have started to get a better understanding of the intercepted protocol payload. In fact, we were able to identify several values and prominent communication flow patterns, which are clearly related to the matching algorithm. In Figure 3, we present a listing of the most interesting sections we have identified. To understand the data, we have visualized it as a grayscale image using one pixel per byte. After some manual alignments, we could identify the raw image of the evaluated fingerprint. This demonstrates that the SPI communication observed is not encrypted and all data transmitted can be intercepted, recorded and analyzed.

In further detail, the tracks we have recorded are composed of the following elements: Initially, the CPU sends a 0xFC command that is responded with the byte sequence 0x020A by the sensor. Afterward, the CPU will initialize the sensor using some values and commands we do not fully understand. In the following, the device goes into a short sleep cycle and will wake up repeatedly every 27ms to poll the sensors actual state. When activity is detected the sensor will signal this by sending a 0x81 byte. Following that, the CPU requests the overall image size and subsequently gather 12 small samples of the finger covering only partial sections of it (each with

CPU Payload (MOSI)	Sensor Payload (MISO)	Description
FC00 006C 3390 3768	0002 0A00 0000 0A00	» Hardware ID “FPC1020A”
3636 3636 3F3F 3F3F	0808 0909 1212 1313	
9C55 4000 3F24 8800	0055 4000 3F24 0000	» Unknown device configuration
0800 009C 5540 003F	0000 0000 5540 003F	
348C 32A8 0F1E 5C0B	2400 0200 0F1E 0003	
A00A 01A0 0A01	0000 0000 0A01	» Sleep 0.1448 seconds
1C00 241C 00A0 0A01	00FF 0000 0000 0A01	» Polling the sensor
1C 00A0 0A01..	00 0000 0A01..	Repeat until finger detected
....	....	
.... ..1C 00A0 0A01	.... ..00 8100 0A01	» Finger detected 0x81
5430 6060 081C 00C0	0000 C000 C000 8100	» Request image size 0xC0C0
1C00 C400 00..	0020 0000 AAA8 A59E	» Transmission of 12 subimages
.... 0000	 7870	8 x 8 x 12 = 768 Bytes
5400 C000 C01C 00C0	0030 6060 0800 0000	
1C00 C01C 00C4 0000	0000 0000 2000 00F5	» Transmission of full image
0000 0000	F4F5 .... 9DA9	192 x 192 x 8 = 36864 Bytes
A00A 0100 0000 0000	000A 0100 0000 0000	» End of communication

Figure 3: SPI communication between CPU and sensor.

a size of 8×8 pixel). Afterward, the transmission of the full image takes place, which is signaled with the byte sequence 0x200000. Following that, 36864 bytes will be transmitted, describing a 192×192 pixel grayscale image with one pixel per byte. In the end, the communication between sensor and CPU is terminated. For enrollment, we can observe a similar communication flow, but in this case, 10 images are sequentially requested by the CPU.

4 Proof Of Concept

In this section, we describe the technical details on how we interfaced the card and evaluated the overall performance of the device against various attacks.

4.1 Setup

Since we know that the communication between sensor and CPU is not confidential, we designed a man-in-the-middle attack that allows us to scrutinize more in-depth features of the device. Therefore, we prepared the backside of the card by carefully cutting the MISO wire disconnecting the sensor from the CPU. Afterward, we rerouted this signal through an FPGA by soldering two wires at both ends and connect them to the GPIO pins of the FPGA. Additionally, we tapped the SPICLK to be able to process the data transmitted. Based on this configuration, we created an FPGA design capable of reading and spoofing the communication stream. This application features two use-cases: A pass-through mode, reconnecting the origin wire signals and an injection mode overwriting the original data stream with a modified one stored in the FPGAs memory. To upload this data, we have added a UART communication interface connecting the FPGA with a host computer via USB UART/FIFO IC. For this implementation, we have used a DE0-Nano board containing an Altera Cyclone IV FPGA with 32 MB SDRAM (Comparable solutions available for $42). Certainly, the FPGA features only 3.3V inputs, while the card requires 1.8V, hence we added a level conversion

solution in between. Clearly, this solution is for experimental use and can not be used inconspicuously in front of other people. However, due to its NFC capabilities, this logic can be easily built on a small daughterboard attached on the backside of the device. Hereby the front side of the card will remain untouched and does not look suspicious at all.

Based on this setup, we created a number of experiments to scrutinize the internal functions of the biometric smart card. The experiments are based on an enrolled fingerprint of a real test person. We used the implemented enrollment procedure of the card and instructed the test person with the user guide published by the vendor [28]. The data and tools to repeat the experiments can be found in our repository [9].

4.2 Replay Attack

First of all, we tested how the system responds to a replay of a valid verification we have recorded. Using our implementation, we simply configured the FPGA that any data sent by the sensor will be discarded and replaced with the previously recorded one that is stored in the FPGA memory. On the first try, the replayed fingerprint was immediately accepted, and the attack could be repeated multiple times. From this simple experiment, we have already determined two major facts:

- No replay and no liveness detection was triggered
- No tamper protection was violated by our hardware modification

Furthermore, this shows that the recorded payload covers all the configuration and data required to perform a valid authentication. By modifying and replaying this payload, we are now able to scrutinize the in-depth functionality of the device that was difficult to access beforehand.

4.3 Fuzzing the Protocol

In the following, we started to manipulate single messages of the replay data to spot additional attack vectors. The recorded communication data has a size of round about 37 Kilobyte, but may vary depending on the number of wrong attempts and wait time during the recording. As described in Section 3.3, the protocol contains a preview function that sends 12 small samples (8x8 pixel) before the full image is transmitted. These samples have a fixed position and are distributed equally on the sensor at several points. According to the documentation, this feature is intended to decrease the response time and improve the overall performance [8]. However, the preview is not even evaluated by the card and can be replaced with arbitrary data. Furthermore, we modified several configuration fields received by the microcontroller from the sensor. We tried to manipulate the Hardware ID, vary the capture size and inject interesting corner cases for other parameters, to cause side-effects like out-of-bound read/write or downgrading of sensor capabilities. Based on our observation, the modification of these values was not effective. A succeeding read of these values returned their original configuration or did not yield any interesting change. Most probably, these values are hard-coded in the application, at least for this product version.

4.4 Fingerprint Extraction Attacks

In the following, we started to manipulate the image data that is sent to the CPU, as shown in Figure 3. We created a tool to replace the data in the recorded sample with arbitrary image data. Furthermore, we have taken related research into account that shows how the biometric data of a victim can be collected [22]. As described there, fingerprints can be extracted from photos or copied from touched objects like coffee cups, keyboards, and other things. To test the device against this threat, we have developed two more experiments.

4.4.1 Latent Fingerprints on the Device

With regard to the manufacturer's promise that "fingerprint data cannot be extracted from the card", it was the most obvious idea to recover a latent fingerprint from the device itself. In everyday life, the card owners will touch the cards surface and its components and hence spread biometric data all over the device. To imitate this, the test person was requested to touch a similar smart card at several points intentionally. Afterward, the card was evaluated to spot latent prints and recover the biometric data from its surface. Figure 4 shows that multiple fingerprints are visible on the device, especially when it is illuminated and aligned in the right way (90° between the point of view and the light source). To copy the fingerprint residues we made several pictures using a standard iPhone 5 camera. We reviewed the pictures taken and selected the most promising one in terms of image quality. Afterward, we extracted a suitable fingerprint dummy using the following steps:

1. Crop the image area covering the whole fingerprint
2. Apply grayscale conversion, then color inversion
3. Crop and scale the relevant fingerprint area with respect to the physical and digital sensor size, e.g. 11×11mm, 192×192 pixel, 508dpi
4. Improve brightness, contrast, and gamma.

Using our tools, we embedded the extracted fingerprint dummy into the recorded communication payload and

Figure 4: Step-wise extraction process: 1. Initial photo of a smart cards surface with latent prints (w/o modification); 2. Remove irrelevant parts of the image; 3. Grayscale and color inversion; 4. Crop a final digital fingerprint dummy.

uploaded this to the FPGA. When activating the card, the FPGA injected the custom fingerprint created from the latent prints on the card. This way, we have successfully bypassed the authentication in a repeatable fashion.

4.4.2 Fingerprints on Digital Images

Another, more passive way to extract the biometric data can be done by using pictures of the actual user covering his fingerprints. These pictures can be created by the attacker from a distance or can be found on the web as discussed in [22]. To evaluate this scenario, we created multiple pictures covering the test person while showing his fingers. We set up an increasing target distance respectively to 3, 4, 5, 6 and 7 meters. The pictures were taken using a Canon EOS-D1 X with a 200mm lens in an outdoor daylight setting. After taking the pictures, we started the extraction process similar to the previous one. The main difference is in fact that we had to flip the image horizontally and further scale the area depending on the target distance. In Table 1 we are describing this relation and the final upscale factor we have used. Figure 5 shows the pictures taken and the extracted dummies. Again, we injected the obtained fingerprints into the recorded communication payload. We defined a maximum amount of 3 attempts per image, which has given us the freedom to slightly improve the image gamma, brightness, and contrast. Under these constraints, 3 out of 5 dummies caused a valid authentication and could successfully bypass the matching algorithm.

Distance	Crop Size	Upscale	Evaluation
3 m	136×136 px	141 %	Valid
4 m	101×101 px	190 %	Invalid
5 m	88×88 px	218 %	Valid
6 m	68×68 px	282 %	Valid
7 m	57×57 px	337 %	Invalid

Table 1: Results for dummies created from pictures.

4.5 Algorithmic Weaknesses

The previously shown attacks are not only resulting from design issues and the lack of security measures. One of the major reasons for this are various algorithmic weaknesses that undermine the attack resistance of this device.

First of all, we discovered that just 50% of the fingerprint image is sufficient to authenticate a user successfully. This means, with respect to the attacks shown, we don't even need to extract an ideal fingerprint snippet. Single parts of poor quality can easily be removed to improve the attack, as shown in Figure 6. Furthermore, we figured out that ridges without minutiae can be removed from a fingerprint, as long as they got replaced with some arbitrary ridges. In practice, we have simply created some circle-like elements which will be traced by the matching algorithm but do not yield any minutiae data. Combining these findings, we could gradually remove most parts of one fingerprint, until we had a minimal version to authenticate. Such a fingerprint dummy is shown in Figure 7.

The most concerning weakness we have identified is related to the required amount of matching minutiae. According to ISO/IEC 19794-2, this is the most important parameter for any minutiae-based matching system. The recommended minimum amount for enrollment is 16 and for verification is 12. Lowering these numbers will have a huge impact on the attack resistance of the system [12]. With this in mind, we tested the device for standard compliance. First of all, we tested how many minutiae points are actually required for enrollment. We created several fingerprint images, similar to the examples in Figure 8, which we have each enrolled and verified on the device. In conclusion, as long as enough ridges are contained, we could enroll even extremely simplistic fingerprints with less than 5 minutiae and verify them with success.

Secondly, we started to remove single minutiae from a previously enrolled fingerprint to determine the min-

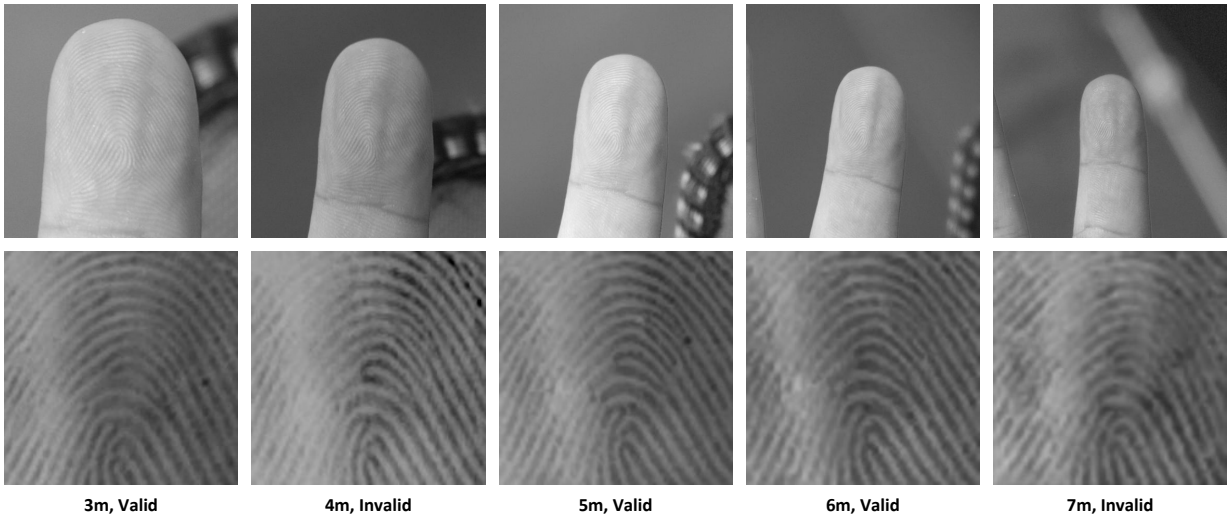

Figure 5: Original finger images (cropped by 450×450 pixel) and obtained fingerprint dummies for 3-7m distance.

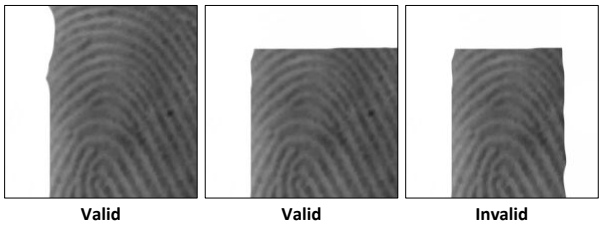

Figure 6: Examples of incomplete fingerprint samples. From left to right the images are cut by 23%, 40%, 53%.

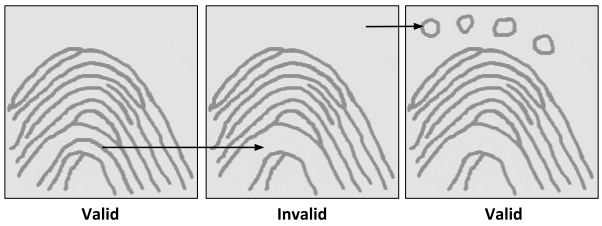

Figure 7: Replace irrelevant ridges with circle structures to evaluate the corner cases of the matching algorithm.

imum threshold for verification. For our baseline template, containing 18 minutiae points overall, we have reduced the number to less than 10 points and were still able to authenticate. Indeed, the concrete threshold depends on the amount of perceived and stored minutiae, image quality, and the applied scoring system. Since these factors cannot be reviewed for the proprietary algorithm, further investigation is not meaningful. Nevertheless, from these examples, we can already conclude that this device is not compliant with ISO/IEC 19794-2 and several corner cases can be exploited to make attacks more reliable. More examples can be found in [9].

4.6 Known Template Attack

Due to the poor performance of the algorithm, it becomes clear on how frighteningly little data is needed to trick this device. That's why we evaluated a more general threat for biometric devices related to the work from [4]. Biometric data is stored and used on multiple devices like smartphones, smart cards, and ID cards. Furthermore, it is known that the integrated memory in these devices is not fully secure. A large number of attacks have been published in the past to read sensitive data from different kinds of memory without authorization, like [21]. Additionally, software bugs and side-channels will further augment the attack surface of systems storing and processing biometric data. A very close example is given by [7]. They demonstrate how non-invasive side-channel attacks can be used to extract the template data from matching algorithms during processing. In the end, the leaked data can be used to create artificial fingerprints with similar features like the original [4].

To take up this work, we have tested how resistant this particular system is against the reuse of template data. Therefore, we extracted the corresponding template from the fingerprint of the enrolled test person, including minutiae coordinates, types and orientations. Based on this information only, we started to create several artificial fingerprints. We placed the correct type of minutiae at their relative locations and successively added some arbitrary ridges by applying the findings of our previous analysis. The resulting fingerprint images can be seen in Figure 8. Obviously, any human can recognize that these fingerprints are clearly counterfeit, but the implemented algorithm accept them without complaining.

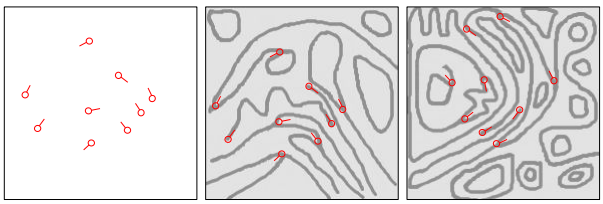

Figure 8: Hand-crafted fingerprint dummies only based on template data, with minutiae locations and directions.

Zoom Deviation	Zoom In Crop Size	Zoom In Evaluation	Zoom Out Crop Size	Zoom Out Evaluation
± 0 px	192 px	valid	192 px	valid
±12 px	180 px	valid	204 px	valid
±22 px	170 px	valid	214 px	valid
±28 px	164 px	valid	220 px	valid
±30 px	162 px	invalid	222 px	valid
±32 px	160 px	invalid	224 px	valid
±42 px	150 px	invalid	234 px	invalid

Table 2: Evaluation of zoom thresholds.

4.7 Limitations

In the following, we discuss the limitations, obstacles, and assumptions we have made for the attacks.

First of all, all attacks shown suffer from the fact that the attacker needs to guess which finger is enrolled. Starting from 10 fingers, we can halve this number by figuring out if a person is right or left-handed. In addition to that, most users will deliberately use the thumb or index finger, due to the way how a smart card is held with a single hand. Additionally, the cropped fingerprint section needs to be part of the enrolled template, although it is unknown to the attacker. However, the device manual, like most fingerprint matching systems, advises the user to increase coverage as much as possible during enrollment [28]. This increases the usability as much as the chances of a successful attack. At least, the central part of the fingerprint might always be covered. Additionally, some attacks require that the selected fingerprint section need to physically and digitally fit the size of the sensor (e.g. 11×11 mm, 192×192 pixel). Having a physical copy of the fingerprint the relevant area can easily be measured. Other cases, such as picture extraction as shown in Figure 5, require a special scaling ratio related to the target distance. For similar setups, these values could be derived from Table 1. To determine the limits of this technique, we have evaluated how the system responds when different zoom levels will be applied. Starting from an ideal section with a size of 192×192, we incrementally selected a smaller/larger area of the fingerprint and cropped it again to the size of 192×192 pixel. This transition relates to the error made during the extraction of the fingerprint by estimating the distance and dimension of the finger. As shown in Table 2, an error of ± 30 pixel can still result in a valid authentication.

When a promising image region has been selected and scaled, any irregularities need to be compensated. The easiest way is to use image editing tools to change the contrast, brightness or gamma values of the image. In this context, analysis tools for fingerprints such as vFinger [18] can be used to measure if changes have improved or degraded the overall quality. For us, a subjective review was always sufficient. In addition, enrollment algorithms like [18] will also help to compose decent fingerprints out of multiple photos from different parts of the fingerprint.

5 DISCUSSION

In the final section, we discuss open questions and conclude our findings. We also describe countermeasures and ideas to make this and similar devices more secure.

5.1 Countermeasures

Our research has shown that, besides ARM memory protection, no further active or passive countermeasures in hard- or software is used on the evaluated system. In the following section, we want to summarize what techniques can be applied to mitigate fraud. The shown attacks generally benefit from the missing replay and liveness detection. On embedded devices, replay detection can be done by using rolling or fuzzy hashes of already-seen fingerprints. These hash methods are able to identify bit-identical samples, as well as slightly modified ones. An introduction to this and additional information can be found in [15] and [10]. For live detection capabilities, the device might process multiple samples instead of single ones or integrate more sophisticated sensors able to evaluate the physical characteristics of the test object. To protect the on-device communication, data bus encryption can be used to mitigate wiretapping attacks. In addition, session identifiers, nonces or timestamps can be used to detect replay attacks of encrypted content. Hardware countermeasures like logic duplication and mesh detectors are able to detect physical modification of the device [17]. To prevent side-channels, as shown in [7], dummy instructions or side channel free algorithms must be used. Overall, it will be important to improve and modify the internal thresholds and decrease the false match rate. In particular, the guideline and parameters for matching and decision from ISO/IEC 19794-2 should be taken into account [12]. For users, we highly recommend making use of the optional 3-factor authentication which requires a valid card, PIN, and the fingerprint. In the end, most countermeasures will affect

the product's performance, cost and user acceptance, but will strengthen trust and confidence. Even when other attacks remain possible, it will increase the effort, costs, and skills to attack such a device in a meaningful way.

5.2 Conclusion

In this paper, we presented the fundamentals of biometric fingerprint authentication based on the example of a new match-on-card device created by a company who must not be named. We have analyzed the already available demo card for access control of their match-on-card platform, which is going to be integrated into upcoming payment and identity cards [27, 16]. The final product version could not be evaluated because it is not released and due to legal and ethical issues. However, this work is intended to help developers and responsible parties to improve this and similar systems before deployment and point out the risks to the customers beforehand.

In summary, we have shown how software and hardware hacking techniques can be used to bypass the match-on-card device. The underlying threat model requires to steal or copy the card and extract some biometric data of the user. Using these ingredients, an attacker can exploit several design flaws to spoof the communication between the CPU and sensor. Due to this issue, it is possible to pass arbitrary data to the biometric matching algorithm, and this way inject digital fingerprint dummies to bypass the authentication mechanism. The required biometric data will be unintentionally spread by a user touching the card or any other object. This data can be copied and collected using various techniques. To demonstrate this, we have evaluated how latent fingerprints can be photographed from surfaces like the card itself. In addition to that, we have tested how sufficient fingerprint dummies can be created from pictures showing the palms of a victim. During this step, the attacker normally needs to create a physical dummy, which is an elaborate and error-prone task, as shown in [6]. In contrast, our attack can make use of digital dummies that vastly increase the quality, ease of use and reusability of the dummy. Furthermore, we have demonstrated the practical impact of leaked fingerprint templates on a real device. In comparison to [4], this is possible even without the effort of creating natural-looking fingerprints from the template. In fact, we created obviously counterfeit fingerprints that nevertheless could be used to bypass the matching algorithm successfully.

In conclusion and with respect to the threats discussed in [11] and [26], this device is suffering from several kinds of attacks: the use of dummies (no liveness or replay detection), the use of latent prints (reuse of fingerprint residues) and the use of biometric lookalikes (known template attack). The main reasons for this are the weak matching algorithm, badly chosen thresholds and the lack of software and hardware countermeasures. As a consequence, an impostor can easily impersonate another person who is enrolled and gain access to the protected assets. For this reason and with respect to [11], the evaluated architecture cannot be considered secure. Compared to a PIN-protected card, nobody can prevent that the secret key will be unconsciously leaked by touching the device, taking pictures or in various other ways. A PIN-protected card provides way more protection because the attacker needs to guess, extort or find the written PIN, which is at least under the control of the user. To improve the idea of match-on-card fingerprint authentication, we provide several countermeasures for the responsible people. To evaluate our test cases on this or similar platforms, we publish our tools and examples in a public repository [9].

Disclosure Note. The results have been disclosed to the company before publication on July 3rd, 2018. The company requested to remove all of their brands from the report. The company also wants to state that the analyzed product, a version of the company's access card, was discontinued and represents technology that is severely outdated from a hardware, firmware and security feature perspective. Additionally, there are fundamental architecture and technology differences between the company's access control and the payment and ID offerings. The technology and findings of the report are not relevant for the company's payment and ID products, and the findings relate in no way to the current state of the company's technology offering. The authors want to state that there is no legal agreement with the company. The analyzed cards have been ordered from the company's online shop, and the device is still available there. We don't know whether and how the architectural and technological differences impair the attacks on other devices.

Acknowledgements. The authors thank Tobias Fiebig and the Review Committee for the valuable feedback and comments. This work was supported by the German Federal Office for Information Security (BSI) and the Federal Ministry of Education and Research of Germany in the framework of Software Campus 2.0 project no. FKZ 01IS17052. Opinions, views, and conclusions are those of the authors and do not reflect the views of anyone else.

References

[1] Aditi Roy et al., "MasterPrint: Exploring the Vulnerability of Partial Fingerprint-Based Authentication Systems.", IEEE Transactions on Information Forensics and Security 12.9 (2017): 2013-2025.

[2] Atmel Corporation, "SAM4S ARM Cortex-M4 Microcontroller Manual", 2015. [Online]. Available: http://www.atmel.com/products/microcontrollers/arm/sam4s.aspx

[3] Biggio, Battista, et al., "Security evaluation of biometric authentication systems under real spoofing attacks.", IET biometrics 1.1 (2012): 11-24. goo

[4] Cappelli, Raffaele, et al., "Fingerprint image reconstruction from standard templates.", IEEE transactions on pattern analysis and machine intelligence 29.9 (2007).

[5] Chaos Computer Club, "Chaos Computer Club breaks iris recognition system of the Samsung Galaxy S8", 2017. [Online]. Available: https://www.ccc.de/en/updates/2017/iriden.

[6] Chaos Computer Club, "Chaos Computer Club breaks Apple TouchID", 2013. [Online]. Available: https://www.ccc.de/en/updates/2013/ccc-breaks-apple-touchid.

[7] Dürmuth, Oswald, and Pastewka, "Side-Channel Attacks on Fingerprint Matching Algorithms", Proceedings of the 6th International Workshop on Trustworthy Embedded Devices. ACM, 2016.

[8] Fingerprints, "Product Specification FPC1020", 2014.

[9] GitHub Repository, https://github.com/julieeen/swipe

[10] Hartloff, Jesse, et al., "Security analysis for fingerprint fuzzy vaults." Proc. SPIE. Vol. 8712. 2013.

[11] Henniger, Scheuermann and Kniess. "On security evaluation of fingerprint recognition systems." Internation Biometric Performance Testing Conference (IBPC). 2010.

[12] International Organization for Standardization, "ISO/IEC 19794-2:2011: Information technology – Biometric data interchange formats – Part 2: Finger minutiae data", Dec 2011. [Online]. Available: https://www.iso.org/standard/50864.html

[13] Jain, Anil K., Jianjiang Feng, and Karthik Nandakumar, "Fingerprint matching", 2010. Computer 43.2 (2010).

[14] Jain, Anil K., Patrick Flynn, and Arun A. Ross, eds. "Handbook of biometrics". Springer Science & Business Media, 2007.

[15] Juels, Ari, and Madhu Sudan, "A fuzzy vault scheme.", Designs, Codes and Cryptography 38.2 (2006): 237-257.

[16] Mastercard Inc., Press Releases, "Thumbs Up: Mastercard Unveils Next Generation Biometric Card", April 20, 2017. [Online]. Available: https://newsroom.mastercard.com/press-releases/thumbs-up-mastercard-unveils-next-generation-biometric-card/.

[17] Mukhopadhyay, Debdeep, and Rajat Subhra Chakraborty, "Hardware security: Design, threats, and safeguards", CRC Press, 2014.

[18] Neurotechnology, "Biometric Technology Applications". [Online]. http://www.neurotechnology.com/download.html

[19] Precise Biometrics AB, "White paper: Understanding Biometric Performance Evaluation", Nov 2014.

[20] Saleae, Inc., "Saleae Logic. The logic analyzer you'll love to use.", 2017. [Online]. Available: https://www.saleae.com/

[21] Samyde, Skorobogatov, Anderson, Quisquater, "On a new way to read data from memory", In Proceedings - 1st International IEEE Security in Storage Workshop, SISW 2002, pages 65-69, 2003.

[22] Starbug, "Ich sehe, also bin ich ... du", Talk at 31C3, 2014. [Online] https://events.ccc.de/congress/2014/Fahrplan/events/6450.html

[23] Tobias Fiebig, Jan Krissler, and Ronny Hänsch, "Security Impact of High Resolution Smartphone Cameras.", WOOT 2014.

[24] Visa Inc., "Press Here! Visa Begins Pilots of New Biometric Payment Card", Press Releases, January 14, 2018. [Online]. Available: https://usa.visa.com/about-visa/newsroom/press-releases.releaseId.15401.html

[25] Watson and Garris, "NIST Fingerprint Image Software 2 (NFIS2)", National Inst. of Standards and Technology, http://fingerprint.nist.gov/NFIS, 2006.

[26] Zafar, Rehman and Shah. "Fingerprint authentication and security risks in smart devices." Automation and Computing (ICAC), 2016 22nd International Conference on. IEEE, 2016.

[27] A Norwegian biometric smart card company. Product Overview and Product Sheets. 2017. [Online].

[28] A Norwegian biometric smart card company. Access Card Product Manual, 2017.