TrustFA: TrustZone-Assisted Facial Authentication on Smartphone

Dongli Zhang
Stony Brook University

Abstract

Nowadays, many applications, such as Facebook, Dropbox and mobile banking, allow users to login and use the remote services via smartphones. Because of the limitations of credential-based authentication, biometric authentication is becoming popular. Facial authentication is one of the popular biometric authentication techniques and it consists of three phases: first, the photo is captured by hardware camera; second, the smartphone application retrieves the photo from hardware camera via OS; third, the smartphone application authenticates the user by sending the photo (or its extracted features) to remote services. To achieve the trusted facial authentication, all three phases should be secured. In this paper, we propose TrustFA, a TrustZone-assisted solution to secure all three phases in facial authentication on smartphone. We leverage the ARM TrustZone technique to capture the photo and collect the accelerometer data in TrustZone secure world. As all of the secure world memory, peripherals and interrupts are isolated from normal world legacy OS, attackers even with root privilege in legacy OS would not be able to break the authentication. Within our knowledge, this is the first effort that all of three phases in facial authentication are secured. Compared to prior works, the threat model regarding smartphone facial authentication assumed in this paper is the most strongest. Since the prototyping is still in progress, we envision the implementation of TrustFA on Freescale i.MX53 Quick Start Board (QSB).

1 Introduction

Recent years have experienced explosive growth of smartphone sales and smartphones become pervasive. By March 2013, Google has activated more than 750M Android-based devices [6]. Smartphones are no longer basic devices for making phone calls and receiving text messages, but powerful platforms with comparable functionalities to commodity PCs. Mobile applications provide users the functionality to access the data maintained by remote services such as Dropbox, Facebook, and online banking. The credential-based authentication requires the users to provide the password (PIN), which can be figured out by the attacker via social engineering attack. The longer the length of password, the more time consumed by the user to manually input the password. Recently, Shukla et al. [22] introduce a side-channel attack on the PIN entry process on a smartphone. The attack is entirely based on the spatio-temporal dynamics of the hands during typing to decode the typed text. It is demonstrated that the attack breaks an average of over 85% of the PINs in ten attempts on a dataset of 200 videos of the PIN entry process.

Compared to the credential-based authentication, biometric authentication is widely considered more secure. Unlike credential-based authentication which depends on ”the knowledge of user”, biometric authentication relies on ”the identity of user” which is very difficult for attackers to forge.

Facial authentication is one of the popular biometric authentication techniques. As shown in Figure 1, the data flow of facial authentication consists of three phases. First, the photo is captured by hardware camera on smartphone. Second, the smartphone application obtains the photo via the OS. Finally, the smartphone application (processes the photo and) sends the photo (or its extracted features) to the remote server. To achieve a trusted facial authentication, all of three phases, which are vulnerable to attacks, should be secured. Even if the device has a front camera, it is rarely used in practice because of its own limitations.

Figure 1: Facial Authentication Data Flow

Phase 1 The first phase is vulnerable to 2D media attack. Instead of the real 3D face, the attacker can easily fool the 2D face recognition by a flat photo of the user, which is not difficult to download from social networks [1]. Although more sophisticated 3D facial authentication techniques have been proposed to achieve higher security [20, 10], the image processing always consumes a lot of time and the ease of use is also compromised. For instance, to differentiate a real 3D face from a flat photo, the Toshiba Face Recognition Utility requires users to turn their heads towards four directions according to a sequence of arrows shown on the screen, and the whole authentication process takes about 30 seconds. In this paper, we leverage the solution proposed by Chen et al. in [11] to secure the first phase. Besides the user photo, accelerometer is employed to infer the position and orientation of the front camera. A small movement of the cellphone is applied to ensure a real 3D face. Although the proposal of [11] is more efficient than 3D facial authentication, it is still vulnerable in the second phase.

Phase 2 In the second phase, the photo (video) is retrieved by the smartphone application via the legacy OS. On the smartphone, the large TCB of legacy OS, e.g., Android, makes it vulnerable to malwares. The untrusted legacy OS would tamper the photo/video captured by the camera (e.g., virtual camera attack), or replace the captured photo/video with pre-captured ones. Although Chen et al. [11] propose Motion Vector Correlation, that is, to extract the non-intentional shakes of user from both the video and accelerometer, and to verify if they are correlated with each other, they assume the legacy OS is trusted, that is, the attacker is not able to tamper the collected data at OS level. Besides, the computation overhead of Motion Vector Correlation is also relatively high. To overcome the limitation in [11], we leverage the ARM TrustZone technology to ensure the trust of data from camera/accelerometer. Both photo and accelerations are collected in TrustZone secure world. Attackers even with root privilege in legacy OS would not be able to compromise the integrity and freshness of the collected data. The performance overhead of TrustZone world switch is trivial compared to Motion Vector Correlation in [11].

ARM TrustZone TrustZone is a security extension introduced by ARM. The basic idea is to logically partition the computing platform into two execution domains: the normal world and the secure world. To facilitate context switch between the two worlds, monitor mode is introduced as the only entry point from normal world to secure world. Execution in the normal world jumps to the secure world by explicitly issuing the Secure Monitor Call (SMC) instruction. The secure world can access the full range of the physical memory and all hardware peripherals. On the other hand, some physical memory ranges and hardware peripherals can be restricted to be only accessed by the secure world. Therefore, these secure physical memory and hardware peripherals are under full hardware-based protections from attacks that can potentially compromise the normal world legacy OS. Besides, interrupts and DMA are also world-aware.

Phase 3 In the third phase, the photo (or features extracted from photos) is sent to the remote service to authenticate the user. As this phase can be secured by SSL/TLS, we will not discuss the detail in the paper.

In this paper, we propose TrustFA, a TrustZone-assisted facial authentication to secure all three phases mentioned above. The capture of photo, the collection of accelerations, and the encryption/decryption of related data are performed in TrustZone secure world. As all of the secure world memory, peripherals and interrupts are isolated from normal world legacy OS, attackers even with root privilege in legacy OS would not be able to break the authentication. In summary, we make the following contributions in the paper:

- We propose TrustFA, a TrustZone-assisted facial authentication. Within our knowledge, this is the first effort that all three phases in facial authentication are secured. Compared to prior works, especially [11], the threat model regarding smartphone facial authentication assumed in this paper is the most strongest.
- As we leverage ARM TrustZone to ensure the trust and freshness of camera/accelerometer data source, the performance overhead of securing Phase 2 is only the overhead of TrustZone switch. The future work only needs to focus on how to more efficiently prevent 2D media attack.
- Since the prototyping is still in progress, we envision the implementation of TrustFA on Freescale i.MX53 Quick Start Board (QSB). We demonstrate the preliminary performance evaluation of TrustZone world switch. Our vision is also applicable on other ARM development boards.

2 Design

2.1 Overview

The architecture of TrustFA is in Figure 2. We assume the smartphone is running Android OS as the legacy OS. Besides the legacy OS in TrustZone normal world, a secure kernel is placed in secure world. The secure kernel can be either a customized Linux kernel or a new tiny kernel developed from scratch (e.g., xv6 [7]). The device first boots into the secure kernel in secure world which then boots the legacy OS in normal world. Touch screen driver, display driver, crypto library and computer vision

library (e.g., OpenCV) are ported to secure world kernel. A kernel module (tz.ko) is loaded in normal world for the communication between two worlds.

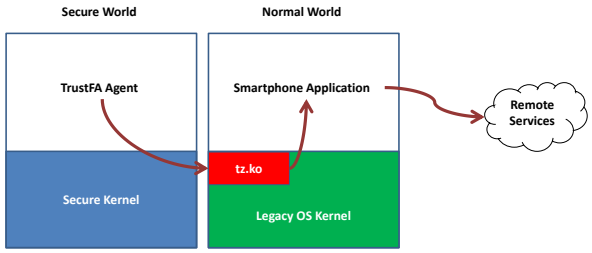

Figure 2: TrustFA Architecture

Trust Model Our trust model is rooted in the hardware isolation provided by the ARM TrustZone. The TCB includes secure kernel and TrustFA Agent in secure world. We assume the legacy OS is untrusted and can be potentially malicious. If the legacy OS becomes compromised, the TrustZone ensures the integrity and confidentiality of code and data residing in the secure world. Besides, we assume the attacker is able to obtain the flat photo and video of user from social networks to mount the 2D media attack..

Secure boot ensures that only the untampered image can pass the integrity check of the chip and boot on the device (Section 3). The device owns a device key K_{DEV} which is flashed permanently on the one-time programmable (OTP) fuses. Only secure world has access to fuses to retrieve K_{DEV}. Besides, a key pair (K_{public} and $K_{private}$) is generated. K_{public} is placed on the remote server provisioning the service. $K_{private}$ is encrypted with K_{DEV} (as $E_{K_{DEV}}(K_{private})$) and stored on the persistent storage. The secure kernel retrieves the encrypted $K_{private}$ (as $E_{K_{DEV}}(K_{private})$) via the normal world legacy OS storage driver and decrypt it with K_{DEV} in secure world. As K_{DEV} is only accessible in secure world, the legacy OS is not able to obtain $K_{private}$ in normal world.

2.2 2D Media Attack

To differentiate the 3D face from 2D counterfeits, we leverage the solution in [11] to correlate the camera with the accelerometer. As soon as the authentication starts, the user moves the smartphone horizontally for a short distance in front of the face from left to right. Once the face area is greater or equal to 40% area of the video frame, the smartphone starts sampling video (from camera) and accelerations (from accelerometer). Once the face area is smaller than 30%, the sampling stops.

TrustFA Agent analyzes the sampled accelerations to calculate the time t_l and t_r when the smartphone is on the left and right of the face respectively. The video frames at t_l (as $P(t_l)$), t_r (as $P(t_r)$) and $t_m = (t_l + t_r)/2$ (as $P(t_m)$) are selected and processed. $P(t_l)$ and $P(t_r)$ are used as input for Nose Angle Detection algorithm. The algorithm processes the two frames and identify the nose's angle. The orientation of the angle is reversed when the camera is moved horizontally in front of the face. If the input of camera is a planar photo, the orientation change of nose angle will not happen. More details of the algorithm are in [11]. If the input is not a 2D counterfeit, TrustFA Agent will send $P(t_m)$ (or features extracted from $P(t_m)$) to the remote server for authentication.

2.3 Untrusted OS

When the legacy OS in normal world is compromised, the attacker is able to tamper the camera/accelerometer data or provide a pre-recorded set of video/accelerations for authentication. For the purpose of preventing this attack, we leverage TrustZone to configure the camera and accelerometer as secure. To reduce the code size of secure kernel, we will not implement the file system and networking driver for it.

To send data to the remote server over internet for authentication, the secure kernel first encrypts the data with $K_{private}$ and transfer the ciphertext to the normal world buffer. The normal world buffer is allocated by the TrustZone driver (tz.ko) in normal world. The buffer can be accessed by both secure world and normal world. Finally, the normal world legacy OS will send the ciphertext to the remote server using its networking driver.

2.4 TrustFA Authentication Workflow

Figure 3 shows the workflow of TrustFA. We assume a Android application wants to authenticate the user with TrustFA. To secure Phase 3, we assume an SSL session is established between the application and remote server to transmit the data securely over internet.

The application first retrieves a nonce from server and boots TrustFA Agent with the nonce in secure world. If $K_{private}$ is not in secure world, TrustFA Agent obtains it via the legacy OS. In step 4, the user moves the smartphone in front of the user's face to take video and accelerations as in section 2.2. In step 5, TrustFA Agent processes the accelerations and extracts three photos $P(t_l)$, $P(t_r)$ and $P(t_m)$. If the input is not a 3D counterfeit, TrustFA will encrypt $P(t_m)$ (or its features) and the nonce together as $E_{K_{private}}(D, nonce)$. The ciphertext is then copied to the buffer in normal world and forwarded to the remote server via SSL session in step 6 & 7. Finally, the server decrypts the ciphertext with K_{public}, check the nonce, and authenticate the user. Once the user is authenticated, the server will send an OAuth token to the application.

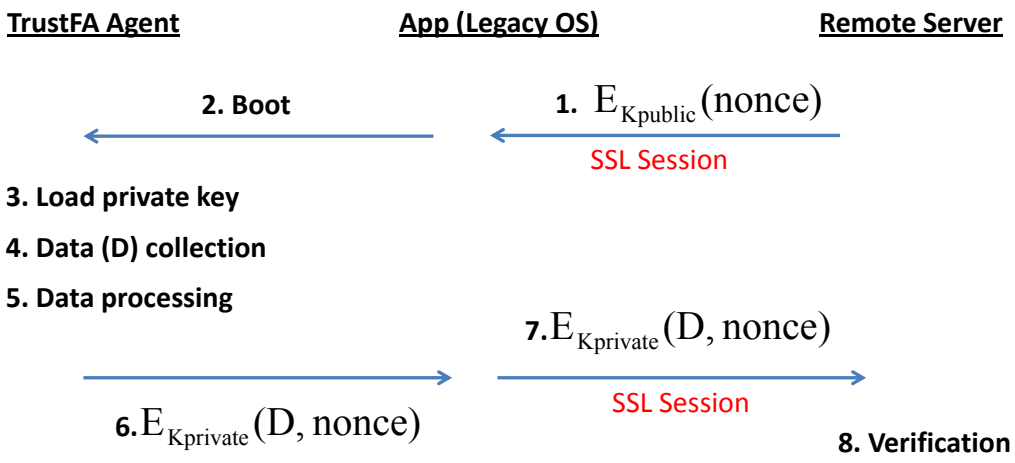

Figure 3: TrustFA Facial Authentication Workflow

3 Vision for Implementation

As the prototyping of TrustFA is still in progress, we envision its implementation on Freescale i.MX53 Quick Start Board (QSB) [4]. The board is equipped with a Cortex-A8 single core with processing speeds up to 1.2 GHz and 1GB DDR memory. Although our vision requires specific registers on i.MX53 QSB, registers achieving the same functionalities are available on other development boards, e.g., Freescale i.MX6 SABRE Lite [2].

Secure Boot The execution of TrustZone's secure world starts with the secure boot provided by the on-chip boot ROM. The i.MX53 processor provides this capability with the High Assurance Boot (HAB) component. The HAB uses digital signatures to authenticate the TrustZone secure world bootloader, which executes immediately after the on-chip boot ROM. The verified bootloader can then verify other secure kernel. HAB authentication is based on public key cryptography using the RSA algorithm in which image data is signed offline using a series of private keys. The resulting signed image data is then verified on i.MX53 processor using the corresponding public keys. Freescale Code Signing Tools HAB does this by computing a cryptographic hash of the Super Root Key (SRK) table and comparing the result with a pre-computed hash that is provisioned in One-time programmable (OTP) fuses. Attacker with unsigned image would not be able to boot the device.

Secure Memory We achieve memory isolation using the Multi-Master Multi-Memory Interface (M4IF), which supports two chunks of physically continuous secure memory. The secure memory can only be configured when the CPU is in TrustZone secure mode. Even after being compromised, the normal world legacy OS is not allowed to access TrustFA Agent and secure kernel. M4IF_WMSA and M4IF_WMEA registers define the start and end address of the secure memory chunks respectively. M4IF_WMIS registers enable the protection of each secure memory chunk. The DDR memory range of i.MX53 is 0x70000000-0xEFFFFFFF (1GB) and we assign 0xc0000000-0xEFFFFFFF (386MB) as secure world memory.

Secure Peripheral We use Central Security Unit (CSU) registers to configure the control policies between bus masters and bus slaves. This will allow us to separate the peripherals into distinct security worlds and prevent the normal world OS from gaining access to secure world peripherals, e.g., camera, accelerometer and touch screen. Besides, normal world peripherals are not able to access secure world memory via DMA. TZIC_INTSEC registers in TrustZone Aware Interrupt Controller (TZIC) specify whether each peripherals are in secure world or normal world.

Preliminary Evaluation The development of TrustFA is still in progress. We have successfully booted a simple bare-metal program in TrustZone secure world. The secure program then boots the Linux 2.6.38 in normal world. We have configured secure memory so that normal world Linux is not able to read/write data in secure world.

Figure 4 shows the performance overhead of TrustZone world switch. We implement a new system call which triggers a null SMC call in secure world (including the context switch between secure and normal world). We compare its performance overhead with null, getpid and fork system calls. Each system call is called 10 times and we measure the overhead of 10 calls together. The total overhead of 10 null system call is 13.2μs. As in Figure 4, SMC system call is 7.8 times of null system call.

Compared to fork, which is 242 times of null system call, the performance overhead of SMC is trivial. Actually, the data cache and instruction cache are disabled in secure world during the evaluation. The overhead of SMC would become less if we enable the cache in the final prototype of TrustFA.

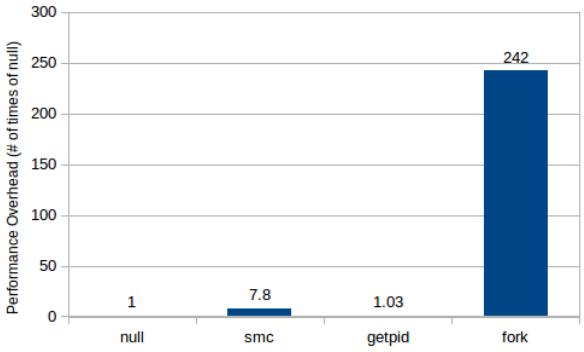

Figure 4: Performance Overhead Comparison of System Calls

4 Discussion

As the world switch is primarily triggered by SMC instruction, the super-privileged attacker in legacy OS can just block the call of SMC instruction to mount the deny-of-service attack to prevent the user from switching to TrustFA Agent in secure world. A solution would be to add a special button on the smartphone and set the corresponding interrupt as secure. Once the button is pressed, secure kernel will be triggered immediately.

The attacker can also mount the man-in-the-middle (MIM) attack by executing a counterfeit of TrustFA Agent in normal world. Instead of the real TrustFA Agent, the user will interact with the counterfeit. As the counterfeit is not able to retrieve $K_{private}$ in secure world, the photos not encrypted with the private key cannot pass the verification of the remote server.

As mentioned in section 3, many peripheral drivers are going to be ported to the secure kernel, which will increase the TCB size. TrustFA is orthogonal to TrustUI [15] minimizing the TCB in secure world by placing only the driver's wrapper in secure world.

5 Related Work

Biometric Authentication Many biometric authentication techniques are involved in our life. Fingerprint-based authentication is introduced with Apple's iPhone [5]. Karthikeyan et al. [14] compare the usability of Apples iPhone 5S Touch ID fingerprint-based authentication with PIN-based authentication and concludes that the former is better than the latter from usability standpoint. In addition, face unlock is also implemented on Android [3]. A fundamental limitation of biometric authentication is that it is not very difficult for attacker to covertly obtain a person's photo and video from social networks, or fingerprint pattern from an object or surface touched by a person. Researchers have developed numerous facial liveness detection techniques [13], e.g., to capture spontaneous eye blinks or lip movements [19]. While it is useful for photo attacks, it cannot deal with recorded videos. 3D face recognition has been widely studied in the recent years [24, 8, 20]. The 3D capturing process is much more time consuming than 2D methods or entering a password. Chen et al. [11] make the first effort to employ motion sensors of smartphones to improve the performance and security of facial authentication. Our work also utilize smartphone accelerometer to correlate 2D photo with 3D facial model and leverage the Nose Angle Detection algorithm in [11]. Regarding Phase 2, our work assume a stronger threat model than [11]. While [11] assumes the legacy OS is trusted, we assume it is potentially malicious. In addition, the computation overhead of virtual camera attack detection of [11] in Phase 2 is relatively high. By leveraging TrustZone, the overhead of machine learning related computing can be eliminated and user only needs to pay for the overhead of TrustZone world switch, which is in the magnitude of microsecond.

ARM TrustZone TrustZone is the security extension of ARM. One of its application is to protect the integrity of the OS kernel. TrustDump [23] is a TrustZone-based memory acquisition mechanism to reliably obtain the RAM memory and CPU registers of the mobile OS kernel even if the OS has crashed or has been compromised. TZ-RKP [9] and SPROBES [12] propose real-time OS protection mechanisms where the kernel is instrumented and all critical kernel modifications will be trapped to TrustZone secure world. Besides, TrustZone has been used to protect sensitive data. DroidVault [16] establishes a secure channel between data owners and data users while allowing data owners to enforce strong control over the sensitive data with a minimal TCB in TrustZone secure world. TrustUI [15] proposes a new trusted path design for mobile devices that enables secure interaction between end users and services based on ARM's TrustZone technology, which is orthogonal to our work. TLR [21] enables the separation of application security-sensitive logic from the rest of the application, and isolates it from the OS and other apps. VeriUI [17] introduces a TrustZone-assisted credential-based authentication. Unlike [15], our TrustFA is a TrustZone-assisted facial authentication and we leverage the accelerometer to different 2D counterfeit from 3D face. Considering the urge requirement of trusted reading from sensors such

as GPS, camera, or microphones, Liu et al. [18] implement a software abstraction for trusted sensors. In our prototype, we will just implement our own software interfaces.

6 Conclusion

The facial authentication data flow on the smartphone can be divided into three phases. TrustFA is the first effort that all three phases are secured. Compared to prior works, the threat model regarding smartphone facial authentication assumed in this paper is the most strongest. We leverage TrustZone to guarantee the trust and freshness of data from camera and accelerometer. In the future, people only need to focus on the first phase, that is, how to more efficiently differentiate 2D flat photo from 3D user.

References

[1] Android 4.0 Face Unlock feature defeated using a photo [Video]. `http://goo.gl/5cgFGz`.

[2] BD-SL-i.MX6. `http://boundarydevices.com/product/sabre-lite-imx6-sbc/`.

[3] How to set up Face Unlock on your Android phone. `http://goo.gl/NFg609`.

[4] IMX53QSB: i.MX53 Quick Start Board. `http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=IMX53QSB`.

[5] Security. Right at your ngertip. `https://www.apple.com/iphone-6/touch-id/`.

[6] Update from the CEO. `http://googleblog.blogspot.co.uk/2013/03/update-from-ceo.html`.

[7] Xv6, a simple Unix-like teaching operating system. `http://pdos.csail.mit.edu/6.828/2014/xv6.html`.

[8] AMBERG, B., KNOTHE, R., AND VETTER, T. Expression invariant 3d face recognition with a morphable model. In *Automatic Face Gesture Recognition, 2008. FG '08. 8th IEEE International Conference on*.

[9] AZAB, A. M., NING, P., SHAH, J., CHEN, Q., BHUTKAR, R., GANESH, G., MA, J., AND SHEN, W. Hypervision across worlds: Real-time kernel protection from the arm trustzone secure world. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (CCS)* (2014).

[10] BLANZ, V., AND VETTER, T. Face recognition based on fitting a 3d morphable model. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* (2003).

[11] CHEN, S., PANDE, A., AND MOHAPATRA, P. Sensor-assisted facial recognition: An enhanced biometric authentication system for smartphones. In *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services* (2014), MobiSys.

[12] GE, X., VIJAYAKUMAR, H., AND JAEGER, T. Sprobes: Enforcing kernel code integrity on the trustzone architecture. In *Mobile Security Technologies (MoST)* (2014).

[13] JAIN, A., AND NANDAKUMAR, K. Biometric authentication: System security and user privacy. *Computer* (2012).

[14] KARTHIKEYAN, S., FENG, S., RAO, A., AND SADEH, N. Smartphone fingerprint authentication versus pins: A usability study. In *Technical Reports: CMU-CyLab-14-012* (2014).

[15] LI, W., MA, M., HAN, J., XIA, Y., ZANG, B., CHU, C.-K., AND LI, T. Building trusted path on untrusted device drivers for mobile devices. In *Proceedings of 5th Asia-Pacific Workshop on Systems (APSys)* (2014).

[16] LI, X., HU, H., BAI, G., JIA, Y., LIANG, Z., AND SAXENA, P. Droidvault: A trusted data vault for android devices. In *Engineering of Complex Computer Systems (ICECCS), 2014 19th International Conference on*.

[17] LIU, D., AND COX, L. P. Veriui: Attested login for mobile devices. In *Proceedings of the 15th Workshop on Mobile Computing Systems and Applications (HotMobile)* (2014).

[18] LIU, H., SAROIU, S., WOLMAN, A., AND RAJ, H. Software abstractions for trusted sensors. In *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services* (2012), MobiSys.

[19] PAN, G., SUN, L., WU, Z., AND LAO, S. Eyeblink-based anti-spoofing in face recognition from a generic webcamera. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*.

[20] QUEIROLO, C. C., SILVA, L., BELLON, O. R., AND SEGUNDO, M. P. 3d face recognition using simulated annealing and the surface interpenetration measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2010).

[21] SANTOS, N., RAJ, H., SAROIU, S., AND WOLMAN, A. Using arm trustzone to build a trusted language runtime for mobile applications. In *Proceedings of the 19th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)* (2014).

[22] SHUKLA, D., KUMAR, R., SERWADDA, A., AND PHOHA, V. V. Beware, your hands reveal your secrets! In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14.

[23] SUN, H., SUN, K., JING, J., AND JAJODIA, S. Trustdump: Reliable memory acquisition on smartphones. In *European Symposium on Research in Computer Security (ESORICS)* (2014).

[24] WANG, Y., LIU, J., AND TANG, X. Robust 3d face recognition by local shape difference boosting. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* (2010).