Secure Enrollment and Practical Migration for Mobile Trusted Execution Environments

Claudio Marforio, Nikolaos Karapanos, Claudio Soriente,
Kari Kostiainen and Srdjan Čapkun
Institute of Information Security
ETH Zurich
{firstname.lastname}@inf.ethz.ch

ABSTRACT

Smartphones can implement various security services from mobile banking to security tokens used for physical access control. System-wide trusted execution environments (TEEs), like ARM TrustZone, allow implementation of these services that withstand malware and operating system compromise. While researchers and developers have focused on secure storage and processing of credentials on such mobile TEEs, secure and practical bootstrapping of security services has been overlooked. The goal of this paper is to put forward the problem of secure user enrollment in the context of mobile system-wide TEEs. We explain why user identity binding to a mobile device is challenging on current smartphone platforms, and argue that current mobile device architectures do not facilitate secure enrollment and migration for such TEEs. We outline possible architecture changes that would enable the realization of secure and practical enrollment, and thus enable more widespread secure deployment of various mobile security services.

Categories and Subject Descriptors

D.4.6 [**Security and Protection**]: Authentication

Keywords

ARM TrustZone; Secure Enrollment; Second-factor Authentication; Trusted Execution Environment

1. INTRODUCTION

Modern smartphones can implement various security-sensitive services, from mobile banking to security tokens used for, e.g., public transport ticketing [20], physical access control [3], and second-factor authentication [12]. Compared to systems that use dedicated tokens, the ones using smartphones are less expensive to deploy and free the users from the burden of having to carry multiple tokens.

SPSM'13, November 8, 2013, Berlin, Germany.

http://dx.doi.org/10.1145/2516760.2516764.

As the complexity of smartphone platforms has increased, software vulnerabilities on mobile operating systems have become commonplace [8, 30]. To realize smartphone security services that are resistant against mobile OS compromise, the implementation of the service should be isolated from the untrusted mobile OS. Isolated code execution and secure storage are services offered by hardware-based *Trusted Execution Environments* (TEEs). The SIM card (Subscriber Identity Module), present in most mobile devices, constitutes a *dedicated* TEE where security-sensitive services, that withstand mobile OS compromise, could be implemented. However, services implemented on SIM cards are limited to isolated computations, and, furthermore, deployment of global services on SIM cards is challenging due to a large number of mobile network operators involved in the process.

In the last decade, mobile device manufacturers have equipped their devices with *system-wide* TEEs, such as ARM TrustZone (TZ) [2]. TrustZone augments the mobile device main CPU with a secure execution mode that allows isolated execution of small pieces of trusted code, called *trusted applications*, and secure storage of user credentials. Furthermore, system-wide TEEs enable secure access to system peripherals and memory resources. Therefore, they enable more complex security services, such as secure boot, enforcement of subsidy lock, and trustworthy sensing [21].

Smartphone applications that replace dedicated security tokens require *secure enrollment protocols,* where the service provider associates the identity of a user to his device. Dedicated security tokens are user-specific devices for which user-to-device *binding* happens within the service provider premises, before the token is shipped to the customer. When smartphones replace such tokens, a service provider must bind each user identity to his device after the user has signed up for the service. On SIM cards, such user-to-device binding is simple, due to the existing security association with the network operator. On system-wide TEEs like ARM TrustZone, similar pre-established user-to-device bindings do not exist. Secure establishment of such bindings at the time of service registration is hard to realize, assuming an adversary that controls the mobile device OS.

Besides initial user enrollment, a practical security service must also support easy and user-friendly *device migration.* In many application scenarios, it is realistic to assume a one-time registration operation carried out, for example, when the user visits the service provider in person. Requiring such a registration every time the user switches to a new device becomes both expensive for the service provider and incon-

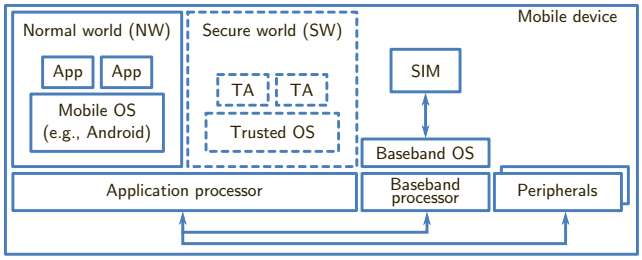

Figure 1: Mobile device architecture overview. The dashed boxes correspond to the enhancements that ARM TrustZone adds within the application processor.

venient for the user. On the one hand, dedicated TEEs like SIM cards provide easy migration, as they can be moved to the new device. On the other hand, secure and practical realization of device migration for system-wide TEEs like ARM TrustZone is harder to implement.

In this position paper, we discuss the problem of secure user enrollment and device migration on system-wide TEEs. We consider a powerful adversary that can compromise the target device mobile OS remotely as well as the system-wide TEE environment if he has physical access to the device. Such adversarial models are justified by software vulnerabilities found in mobile operating systems widely used today and by the fact that system-wide TEEs are not designed to provide tamper resistance against sophisticated physical attacks.

We explain why common user enrollment mechanisms cannot be implemented securely in the presence of such a powerful adversary, and conclude that current mobile device architectures must be changed, either in software or hardware, to achieve secure enrollment. We outline architectural changes including updates to baseband environments and SIM cards that enable the realization of secure enrollment for system-wide TEEs. Finally, we discuss applications where secure enrollment is needed and compare system-wide TEE service deployment to SIM card service deployment. We focus our discussion on ARM TrustZone, as it is currently the most widely deployed system-wide TEE in mobile devices, although it applies to other system-wide TEEs as well.

The purpose of this paper is to draw the attention of the security research community, mobile device manufacturers, and also network operators to secure enrollment—a problem that has been long overlooked. System-wide TEEs enable flexible implementation of various security services, but without secure enrollment and practical migration mechanisms, the potential of these TEEs cannot be fully utilized.

2. MOBILE DEVICE ARCHITECTURE

Solid boxes in Figure 1 show a typical mobile device architecture. The device has two processors. An application processor runs the mobile OS (e.g., Android) and the applications on top of it. A baseband processor handles cellular communication and mediates communication between the application processor and the SIM card. The software running on the baseband processor is called baseband OS and is typically smaller and less complex than the mobile OS.

A SIM card is a dedicated TEE, managed by the issuing mobile network operator. It stores the International Mobile Subscriber Identity (IMSI) and a corresponding secret key that is used for authentication to the network operator. Moreover, SIM cards can execute third-party applications, on prior agreement between the application provider and the issuing network operator [10].

Recent mobile devices also support system-wide security architectures, ARM TrustZone being the most popular one, which we briefly describe in the following paragraphs. The dashed boxes in Figure 1 illustrate the enhancements that ARM TrustZone brings within the application processor of the device. In a TZ-enabled device, execution is divided in two states: *secure world* and *normal world*. The application processor switches between these worlds using time-slicing, so that only one mode is active at a time. The normal world runs the mobile OS and regular applications on top of it, while the secure world runs *trusted applications* (TAs). The latter are small pieces of code that run on top of a small layer of software called the *trusted OS* that is responsible for managing the TAs.

Software running in the secure world can access hardware-protected memory areas and run in isolation from the mobile OS. Device peripherals and the baseband processor environment are available to both the trusted software in the secure world and the mobile OS in the normal world. Access control to hardware resources is implemented through specific control hardware and signals on the system communication bus. In contrast to SIM cards and similar dedicated TEEs, the secure world is able to access any peripheral on the device, just like the regular mobile OS does.

Memory resources that are allocated to the secure world are limited. Thus, a typical trusted application only handles security-critical processing, such as user credential processing or data encryption. The rest of the implementation, including network communication, is typically handled by an untrusted application in the normal world. The inclusion of complex libraries, like network stacks, in the trusted OS, increases the size of the Trusted Computing Base (TCB), and thus, the attack surface of the secure world. For these reasons, the trusted OS and TAs are kept small. We assume the mobile device TCB to consist of the device hardware (including the SIM card), the trusted OS, and the baseband OS.

A typical trusted OS only allows execution of code that has been signed by a trusted authority, such as the device manufacturer. Typically, the device manufacturer ships each device with a device-specific key-pair. The public key is certified by the manufacturer, and the issued device certificate contains an immutable device identifier like the IMEI number (International Mobile Equipment Identity). The corresponding private key is only accessible by software that runs in the secure world [17].

Many mobile device manufacturers have equipped their mobile devices with system-wide TEEs like ARM TrustZone for almost a decade, but as of today the usage of these environments has been primarily limited to a few manufacturer-specific use cases, like implementation of subsidy locks and secure boot. Deployment of third-party applications in system-wide TEEs has been limited, because the installation of new trusted applications is subject to the approval by the device manufacturer. Recent research has shown how system-wide TEEs can be safely opened up for third-party trusted application development [16] and on-going TEE API standardization activities [11] are likely to make trusted application deployment more accessible to third parties.

3. PROBLEM STATEMENT

We address the problem of secure enrollment and practical device migration for applications that leverage smartphone system-wide TEEs, such as ARM TrustZone. We assume a one-time service registration that requires an authenticated channel between the user and the service provider. For example, such a registration may require the user to visit the service provider in person.

After initial registration, the user should be able to enroll his device and eventually migrate conveniently to a new one. Both enrollment and migration should not rely on the authentic channel used for user registration as that is expensive for the service provider and inconvenient for the user.

The problem lies in designing a system architecture and protocols that allow a service provider to bind a user identity to the TEE of his smartphone, e.g., establishing a shared key with that TEE or learning its public key. The goal of the adversary is to launch an impersonation attack. The attack is successful if, at the end of the enrollment, the service provider binds the identity of the victim user to the TEE of the adversary's device, or to a public key for which the adversary holds the private key.

Adversarial model. We consider a powerful adversary that has full control on any mobile device to which he has physical access. On these devices the adversary controls the secure world trusted OS, the mobile OS, and the baseband OS. The adversary may also extract the TZ-protected keys from these devices. This is a realistic assumption, since the TrustZone architecture does not provide tamper resistance, and is thus susceptible to sophisticated physical attacks.

In our model, the adversary does not have physical access to the smartphone of the victim user, and therefore, he cannot access TZ-protected keys on the victim's device. Given software vulnerabilities commonly found in mobile operating systems, the adversary can remotely compromise the normal world mobile OS on the victim's smartphone; however, he cannot remotely compromise the secure world on that device.

Finally, we note that the adversary can control the Internet communication between the service provider and the victim's device. As the adversary can block the Internet communication and controls the victim's device OS, he can trivially disrupt any protocol. We exclude such denial-of-service attacks from our discussion, and instead focus on the problem of preventing impersonation attacks.

4. COMMON SOLUTIONS

In this section we explain why known and currently deployed enrollment mechanisms are not secure, assuming a powerful adversary as described in Section 3.

4.1 Device Identifier Enrollment

A simple way to bind a user identity to the certified public key of his device, is to leverage the mobile device IMEI that is typically included in the device certificate. The IMEI can be printed on the phone sales package or displayed on-screen. During registration, the user provides the IMEI of his phone to the service provider using a reliable channel, such as visiting the service provider in person. During enrollment, the service provider can verify the certificate of the user's device, with respect to the provided IMEI.

Communicating the IMEI to the service provider in a trustworthy way is more complicated than it seems. Device sales packages are not always available. Furthermore, as shown in Figure 2, a compromised mobile OS may control the identifier shown on the device screen.

Finally, IMEI-based enrollment does not provide flexible device migration, since the user must reliably communicate the IMEI of his device to the service provider, each time he switches to a new smartphone.

4.2 Password Enrollment

Another commonly suggested way to implement user-to-device binding is to ask the user to type in his device a password, or some kind of initialization secret, shared with the service provider. Google 2-step verification [12] uses this approach. In a TZ-enabled device, the user can transfer the password to a trusted application through the device's touchscreen. The trusted application can then authenticate itself to the service provider using its certified public key and the user-provided password. Password enrollment provides also convenient device migration, as a similar enrollment procedure can be repeated on any new device.

As the mobile OS may be compromised, the user should type in the password only when it can be securely received by the trusted application. Otherwise, a compromised mobile OS can intercept the password and forward it to the adversary. The adversary can, therefore, use the password with his certified public key to impersonate the user.

A secure communication interface from the user to a trusted application is typically called *trusted path* [29]. In principle, the hardware and software resources needed for user interaction in a TZ-enabled device (e.g., access to the display buffer or the touchscreen input events) can be temporarily reserved for the secure world. A *security indicator*, such as a colored bar on the top of the screen [23] or a dedicated LED [1] can be used to inform the user about the type of the application he is communicating with (the correct trusted application or a normal world application). However, current smartphones typically do not implement such division of user-interface resources. Furthermore, several academic studies, and a few decades of practical experience, have shown that users tend to ignore security indicators [22]. If the mobile OS can intercept the communication between trusted applications and the user (cf. Figure 2), password-based enrollment mechanisms cannot provide secure enrollment.

Some research proposals suggest that password input operation can be trusted if the enrollment is done early in the device life-cycle, before the adversary has the opportunity to compromise the mobile OS [15]. Nevertheless, this argument is hard to justify since not every service enrollment happens at the beginning of the device's lifetime.

4.3 SMS Enrollment

Secure enrollment protocols can leverage the implicit binding between the user's phone number and his SIM card. An SMS sent to the user's phone number is securely delivered to the device where the SIM card is installed. This provides a natural way to establish a secure connection to the correct user smartphone. Assuming that the user has provided his phone number to the service provider during registration, the service provider can send an SMS with an initialization secret (e.g., a key) to that particular phone number. This approach is used by, e.g., WhatsApp [28]. When the user migrates to a new device, he typically moves his SIM card to the new phone. Thus, SIM cards also provide convenient device migration.

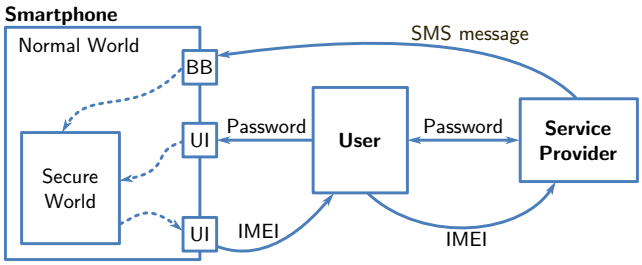

Figure 2: Common solutions for secure enrollment. Solid lines depict reliable channels, while dashed lines illustrate channels that the adversary controls. BB stands for "baseband processor", while UI stand for "user interface".

The problem with the approach described above is that SMS messages provide a trustworthy channel to the device baseband OS, but not to the TrustZone secure world on the same device. In current mobile device architectures, the baseband OS is accessible by both the normal world mobile OS and the secure world trusted OS. When the baseband OS receives an incoming SMS, it generates an event that can be processed by the untrusted mobile OS. Thus, the mobile OS can intercept initialization secrets sent over SMS messages and forward them to the adversary (cf. Figure 2).

4.4 Enrollment in General

The problem of secure enrollment is comparable to the establishment of an authenticated channel between two parties, without any pre-shared secrets. Solutions to this problem require an out-of-band channel that the adversary does not control.

As shown in Figure 2, if the adversary controls the mobile OS on the victim's device, an out-of-band channel cannot be established. Indeed, in current devices *any* communication channel to the TrustZone secure world can be intercepted by the untrusted mobile OS. Similar arguments to the ones discussed previously apply to any other communication interfaces that the device supports. In a practical mobile device configuration, all hardware elements that can perform external communication are configured to be accessible by both the normal world mobile OS and the secure world trusted OS. For example, access to the baseband processor or to other wireless interfaces cannot be reserved exclusively for the secure world, as many normal world applications require the functionality provided by these interfaces.

Our secure enrollment problem resembles the well-known problem of TLS handshake against a man-in-the-middle adversary who holds a key with a valid certificate. In a TLS handshake, two parties, where at least one of them has a certified public key, want to establish a secret. An adversary that controls any communication between the two parties and has a valid certificate for his public key, can launch a successful man-in-the-middle attack.

In our scenario, the two parties are the service provider and the TEE on the user's device, where the latter has a certified public key. The adversary controls all communication channels between the two parties (c.f., Figure 2), and he also holds a key with a valid certificate that he may have extracted from the TEE of his device.

Secure device pairing is another well-known scenario where two entities with no pre-shared secret want to establish a secure channel. Prominent solutions [18] to this problem leverage the user as an out-of-band channel (e.g., the user enters secrets or confirms the established key on both devices). In our scenario this is not possible because the mobile OS controls all the channels between the user and the TEE.

5. ARCHITECTURE CHANGES

In this section we discuss architectural changes to mobile devices that enable realization of secure enrollment. We assume a one-time registration in which the user provides his phone number to the service provider in a reliable manner. The enrollment mechanisms, therefore, leverage SMS messages sent to the user's phone number as a reliable channel between the service provider and the TCB components on the user's device.

Our proposals inherently support migration. When the user switches to a new device he typically moves his SIM card to the new phone. The SMS-based enrollment procedures can be conveniently repeated so that the service provider binds the user identity to the TEE running on the new device.

5.1 Software Changes

The baseband can be configured to recognize and process specially crafted enrollment SMS messages. The enrollment SMS message carries a secret key, that the baseband OS uses to authenticate the device IMEI. (The baseband reads the IMEI stored on read-only memory on the phone to handle cellular communication.) The enrollment SMS is then erased and the tag is passed to the normal world which forwards it to the service provider. At this time, the service provider only accepts the public key certificate corresponding to the authenticated IMEI. Full control over the victim's mobile OS does not give the adversary access to the secret key in the SMS message. With this solution the size of the TCB is not significantly increased, as only minor changes to the baseband software are needed. While it is deployable on current mobile device hardware, the drawback of this approach is that it requires the baseband processor to execute added functionality beyond handling cellular communication.

5.2 Hardware Changes

TrustZone-aware baseband. In current mobile device architectures, the baseband processor generates an "event", dispatched to the mobile OS, any time a significant cellular communication occurs. For example, upon receiving an SMS message, the baseband OS fires an interrupt that is processed by the mobile OS. To enable secure enrollment, the baseband processor event handling mechanism can be extended to differentiate between two types of events. The majority of cellular events are dispatched to the mobile OS as usual, while dedicated cellular events can be reserved for processing by the secure world. Upon receiving a particular SMS that triggers a secure enrollment protocol, the baseband OS generates a special event and the event processing mechanism (e.g., interrupt controller) triggers the secure world execution that handles the incoming message. With these modifications, the adversary does not learn the enrollment secret, because the untrusted mobile OS cannot re-configure the event dispatching mechanism. While this solution requires hardware configuration changes to the current mobile devices, it offers added flexibility. Service providers can deploy different

schemes to process enrollment messages, without requiring changes to the baseband environment.

TrustZone-aware SIM. Another option is to equip the application processor with a direct connection to the SIM card. (Access to the SIM card is currently mediated by the baseband.) In the ARM TrustZone architecture, the system bus communication carries the information about the current state of the application processor (normal world or secure world). The SIM card can be modified to read this information from the system communication bus and thus become "TrustZone-aware". With a TZ-aware SIM card, the service provider asks the network operator to send an SMS message containing an enrollment secret to the user's phone number. To prevent the message from being intercepted by a compromised mobile OS, the network operator encrypts the SMS under the SIM card secret key. Upon receiving the SMS message, the SIM card decrypts it and makes its contents available *only* if the application processor runs in secure world. This approach requires active collaboration with the network operator for every enrollment SMS message. This issue can be mitigated in scenarios where SIM cards are equipped with asymmetric keys [9]. In this scenario, the service provider only needs to fetch the public key of the user's SIM card from a public database provided by the network operator. We note that solutions based on TZ-aware SIM cards do not rely on the baseband OS being trusted (i.e., they remain secure even if the attacker manages to compromise the baseband OS).

6. DISCUSSION

In this section we broaden our discussion to other TEEs available on smartphones and we also present different application scenarios that require secure enrollment.

6.1 Dedicated TEEs

We have focused our discussion on the ARM TrustZone architecture, as it is currently the most popular system-wide TEE available on mobile devices. Our arguments similarly apply to any other system-wide TEE that augments the device application processor with a secure execution mode, like TI M-Shield [4] and, potentially, Intel SGX [19].

Besides system-wide TEEs, SIM cards constitute a dedicated TEE widely available on current mobile platforms. SIM cards offer tamper-resistance and, in the context of secure enrollment, easy migration. GlobalPlatform [10] has standardized protocols for third parties to deploy applications in isolated *security domains* within SIM cards. The deployment model requires network operators to set up the security domain for each service provider. Therefore, the service provider must interact with each network operator that issues SIM cards to its clients. For example, a country-wide service provider would need to negotiate with each network operator active in that country, so that all their SIM cards account for a security domain where the application can be installed. For globally targeted services even more contractual agreements would be needed. Such negotiations are clearly infeasible for many service providers. In contrast, current standardization efforts [16] for ARM TrustZone, are opening trusted application development and provisioning to third parties. We can expect current mobile application markets to include TrustZone trusted applications in the near future.

Some devices are equipped with a dedicated slot for smart cards (e.g., SD cards) that can serve as a TEE and are currently used as dedicated hardware tokens [24]. In principle, smart cards could run multiple trusted services. However, they would share the limitations highlighted in this paper for both ARM TrustZone (i.e., all communication is mediated by the mobile OS on the device) and SIM cards (i.e., constrained computational power and memory). Also, deployment of many services using such smart cards is challenging, as not all smartphones provide the necessary reader slots.

6.2 Application Scenarios

Secure enrollment is needed by any application that requires user-specific credentials. Besides public transport ticketing and physical access control, suggested use cases for the ARM TrustZone technology include mobile payments [13], online banking [5], e-health systems [14], and applications that leverage smartphones as second-factor authentication tokens [12]. For our analysis, these applications can be divided in two broad categories: *interactive* applications require security-critical user input or output with the TEE, while *non-interactive* applications provide security services that can be implemented without interaction between the user and the TEE.

Prominent examples of interactive applications are mobile payment and online banking solutions. In such scenarios, the user must endorse different operations, like payment transactions, using the smartphone touchscreen. As discussed in Section 4, a trusted channel from the user to the TEE does not exist in current devices, and safe implementation of such a channel is challenging. Therefore, applications that require security-critical user interaction cannot be implemented completely within the TEE. The TEE can provide secure storage for credentials and prevent simple *offline* attacks, like leaking credentials from the device persistent storage when the device is turned off. However, for *runtime* protection, the application must rely on mobile OS trustworthiness. If the mobile OS can be trusted during application execution, then it is reasonable to assume OS trustworthiness also during user enrollment. In this case, any of the enrollment mechanisms presented in Section 4 is applicable.

The non-interactive application category includes, for example, public transportation ticketing and physical access control applications. In these scenarios, the user places the smartphone close to an NFC-enabled reader device to trigger a protocol between the mobile TEE and the reader. Such services do not require explicit interaction between the user and the mobile TEE (although user confirmation can provide relay attack protection). All security-critical processing can be implemented within the TEE, and thus such applications can withstand a compromised mobile OS at runtime. These applications would benefit the most from the enrollment mechanisms enabled by the architectural changes we have outlined in this paper.

7. RELATED WORK

Usage of smartphones as replacements for dedicated hardware tokens is currently deployed in various systems ranging from second-factor web authentication [12] to banking systems [5]. All of these deployed solutions assume mobile OS trustworthiness. ARM TrustZone has received recent interest from the research community. In [26] the authors evaluate how TZ-enabled devices can substitute hardware tokens for second-factor authentication. They assume a trusted path to the user for which secure implementation is problematic. In [27] the authors systematize hardware-based solutions that

increase the security guarantees available on mobile platforms. In their work the authors do not address the problem of secure enrollment. Recent work proposes the use of mobile TEEs for security-critical applications [6, 7, 25], however they do not address the problem presented in this paper.

8. CONCLUSIONS

With the popularity that mobile devices are enjoying today and their increased use as replacement for dedicated security tokens, a careful understanding of the requirements to employ them in a secure way is important. Mobile security architectures, such as ARM TrustZone, widely available in current devices, offer the opportunity for service implementations with high security assurances. In this paper we highlighted the problem of secure enrollment and practical migration for services that leverage such security architectures, in the presence of a realistic adversary. We conclude that current mobile device architectures need to be changed in order to achieve this. We have outlined architectural modifications through either software or hardware changes, that offer both stronger security guarantees and improved user experience, currently not offered by today's mobile device architectures.

Acknowledgements

This work was partially supported by the Zurich Information Security Center (ZISC). It represents the views of the authors.

9. REFERENCES

[1] ARM. Securing the system with trustzone ready program. (last access 2013). `www.arm.com/files/pdf/Tech_seminar_TrustZone_v7_PUBLIC.pdf`.

[2] ARM. Building a Secure System using TrustZone Technology. `www.arm.com`, 2009.

[3] Assa Abloy. Evaluation of the world's first pilot using NFC phones for check-in and hotel room keys. (last access 2013). `www.assaabloy.com`.

[4] Azema, J., and Fayad, G. M-Shield mobile security technology: Making wireless secure. `focus.ti.com/pdfs/wtbu/ti_mshield_whitepaper.pdf`, 2008.

[5] Barclays. Mobile PINsentry. (last access 2013). `goo.gl/pcuGo`.

[6] Busold, C., Taha, A., Wachsmann, C., Dmitrienko, A., Seudié, H., Sobhani, M., and Sadeghi, A.-R. Smart keys for cyber-cars: secure smartphone-based NFC-enabled car immobilizer. In *ACM conference on Data and application security and privacy* (2013), CODASPY'13.

[7] Dmitrienko, A., Sadeghi, A.-R., Tamrakar, S., and Wachsmann, C. SmartTokens: Delegable Access Control with NFC-enabled Smartphones. In *International Conference on Trust and Trustworthy Computing* (2011), TRUST'11.

[8] Felt, A. P., Finifter, M., Chin, E., Hanna, S., and Wagner, D. A survey of mobile malware in the wild. In *ACM workshop on Security and privacy in smartphones and mobile devices* (2011), SPSM'11.

[9] Gemalto. Mobile ID. (last access 2013). `www.gemalto.com/govt/coesys/mobile_id.html`.

[10] GlobalPlatform. Card Specification Version 2.2.1. `www.globalplatform.org/specificationscard.asp`.

[11] GlobalPlatform. Device specifications. `www.globalplatform.org/specificationsdevice.asp`.

[12] Google Inc. Google 2-Step Verification. (last access 2013). `www.google.com/landing/2step/`.

[13] Google Inc. Google wallet. (last access 2013). `www.google.com/wallet/`.

[14] Harvard Health Publications. Using smartphone apps for heart health. (last access 2013). `www.health.harvard.edu/family-health-guide/updates/using-smartphone-apps-for-heart-health`.

[15] Kostiainen, K., Asokan, N., and Afanasyeva, A. Towards user-friendly credential transfer on open credential platforms. In *International conference on Applied cryptography and network security* (2011), ACNS'11, pp. 395–412.

[16] Kostiainen, K., Ekberg, J.-E., Asokan, N., and Rantala, A. On-board credentials with open provisioning. In *International Symposium on Information, Computer, and Communications Security* (2009), ASIACCS'09.

[17] Kostiainen, K., Reshetova, E., Ekberg, J.-E., and Asokan, N. Old, new, borrowed, blue – a perspective on the evolution of mobile platform security architectures. In *ACM conference on Data and application security and privacy* (2011), CODASPY'11.

[18] Kumar, A., Saxena, N., Tsudik, G., and Uzun, E. Caveat emptor: A comparative study of secure device pairing methods. In *IEEE International Conference on Pervasive Computing and Communications* (2009), PerCom'09, pp. 1–10.

[19] McKeen, F., Alexandrovich, I., Berenzon, A., Rozas, C., Shafi, H., Shanbhogue, V., and Savagaonkar, U. Innovative instructions and software model for isolated execution. In *Workshop on Hardware and Architectural Support for Security and Privacy* (2013).

[20] NFC Times. New York Transit Authority to Test Tag-based Ticketing with Nokia NFC Phones. (last access 2013). `nfctimes.com/news/new-york-transit-authority-test-tag-based-ticketing-nokia-nfc-phones`.

[21] Saroiu, S., and Wolman, A. I am a sensor, and I approve this message. In *Proceedings of ACM International Workshop on Mobile Computing Systems and Applications (HotMobile)* (2010).

[22] Schechter, S. E., Dhamija, R., Ozment, A., and Fischer, I. The emperor's new security indicators. In *IEEE Symposium on Security and Privacy* (2007), SP'07.

[23] Selhorst, M., Stüble, C., Feldmann, F., and Gnaida, U. Towards a trusted mobile desktop. In *International conference on Trust and trustworthy computing* (2010), TRUST'10.

[24] Sorber, J., Shin, M., Peterson, R. A., and Kotz, D. Plug-n-trust: practical trusted sensing for mhealth. In *International Conference on Mobile Systems, Applications, and Services* (2012), MobiSys'12.

[25] Tamrakar, S., and Ekberg, J.-E. Tapping and Tripping with NFC. In *Proceedings of the International Conference on Trust and Trustworthy Computing (TRUST'13)* (2013).

[26] van Rijswijk-Deij, R., and Poll, E. Using trusted execution environment in two-factor authentication: comparing approaches. In *Open Identity Summit* (2013), OID'13.

[27] Vasudevan, A., Owusu, E., Zhou, Z., Newsome, J., and McCune, J. M. Trustworthy execution on mobile devices: What security properties can my mobile platform give me? In *International Conference on Trust and Trustworthy Computing* (2012), TRUST'12, pp. 159–178.

[28] WhatsApp Inc. WhatsApp mobile messaging application. `www.whatsapp.com`, last access 2013.

[29] Ye, Z., Smith, S. W., and Anthony, D. Trusted paths for browsers. *ACM Trans. Inf. Syst. Secur. 8*, 2 (2005), 153–186.

[30] Zhou, Y., and Jiang, X. Dissecting android malware: Characterization and evolution. In *IEEE Symposium on Security and Privacy* (2012), SP'12.