CRYPTOMATHIC

White Paper

Secure Mobile Transactions – Fact or Fiction?





1. Introduction

Many banks and various other businesses, such as online betting, have embraced the ´Go Mobile´ trend to take advantage of the ubiquity offered by smart phone devices and have been developing their own apps, some of which store users´ credentials and other sensitive data in order to allow their clients to carry out transactions while on the move.

Inspired by Apple iOS, Google Android and comparable operating systems, most app developers have directed their attention towards the user experience, but – as is often the case - few have placed emphasis on security, partially for priority reasons but also because of a lack of knowledge. In other words, despite the widespread adoption of smart phones, OSs (operating systems) still remain relatively immature when it comes to security. To tackle this, some initiatives such as the Trusted Execution Environment have emerged, but it could quite possibly take up to a decade before the average user will actually enjoy the security benefits when making a transaction with their preferred apps. The urgent question is then what do we do here and now? Shall we refrain from using mobile devices to transact online because security standards are not high enough on mobile devices?

The answer is not straightforward. To approach this matter pragmatically, Cryptomathic believes that the first step is to understand the attack scenarios and motivations in order to develop an adequate defence strategy. Delivering such a security strategy, while remaining independent from both the smart phone manufacturer and the mobile network operator is an essential element for enabling cost-effective large-scale deployments for a broad usage. The first part of this paper explores the current market threats and the second part explains Cryptomathic's evolutionary security strategies to overcome such threats while taking platform limitations into account. Finally, we will be introducing the reader to the Cryptomathic Mobile App Security Core in the third part.

2. The Threats

2.1 Status quo

Operating in a secure manner in the mobile space is perhaps still considered by many to be somewhat adventurous. However, there is more than simply blind optimism driving this surge: just as mobile is appealing to developers because of homogeneity of platforms and easy distribution channels for software with low barrier to entry, significant savings have also been afforded to the manufacturers. Nevertheless, the trend towards more similar, closed and regulated platforms helps manufacturers with security too, since this enables them to focus their efforts more effectively. As a result the current mobile malware scene is not merely a reflection of the adoption and changing use of the technology but also of the design and distinctive properties of this relatively new platform.

Currently malware has some presence on Android, but is much less prevalent on iOS. Typical malware functionality includes surreptitious sending of SMS messages and calling of premium rate numbers, click diversion (for stealing advertising revenue) and a little keylogging/SMS interception for harvesting credentials and SMS-based OTPs. However, nearly all this malware operates within the bounds of requested permissions, that is, the user clicks to agree to grant the application the permission it needs to perform these malicious acts, and very few currently escalate their privileges using root exploits. Thus the primary attack vector is to advertise the malware in the official appstore (or a third party store when available), and to have it installed by consent.

There is a very active mobile security research community, which arguably surges ahead of the criminal community (unlike Anti-Virus developers, which were playing catch up on PC), that have written about and demonstrated much more advanced and powerful attacks than what

is generally seen in the wild. There is some evidence of adoption of research ideas by hackers, but still generally present malware appears not to be that advanced.

The same view is not given by the companies looking to expand their anti-virus and protection software suites to mobile platforms, who deliberately raise fear, uncertainty and doubt about future trends and point towards the extremely fast rate of malware development. However, this malware development rate is comparable to the growth rate of the platform itself. In industry publications such as the Juniper Networks mobile threat report [1], one can see a bait-and-switch approach where the user is made fearful of the malicious capability of commercial spyware packages, and then switched to expect the proliferation levels of less severe malware. The commercial spyware packages such as Mobispy all require direct physical access to install and usually rooting/jailbreaking [2] of the phone first.

2.1.1 Summing up

It appears that currently, the operating system controls are effectively preventing applications from exceeding their authorised permissions, and that the major problem is the perennial challenge of educating users to make cautious decisions regarding which apps to install. While the latter issue is of course a threat for deploying mobile authentication in general, the reader should note that it means that users who do manage to install the legitimate app are not threatened by malware.

Furthermore, for so long as malware authors can continue to create revenue by exploiting non-escalating compromises (e.g. compromising the web browser but not the whole operating system) to perform actions such as click diversion and advertising, they are not motivated to develop more advanced attacks which threaten the security of existing installed apps. This partitions the malware community and means fewer hackers are working on the most powerful cracks.

2.2 Threat model

Let's now consider the threat model against which a mobile app might need to have protection. In a full model process these threats are derived from a variety of attacker goals, foremost being monetary gain but also include retribution, anarchy, curiosity and perceived public good. The attackers themselves are also classified/grouped by resource levels [3] and goals, as illustrated in Table 1 on the next page.

This table shows that a good mobile security strategy must defend both against specific mobile threats but also against more generic threats such as reputational attack and white-hat hacking, which have an increased prevalence and importance in the dynamic mobile market. However, it also shows that attacks involving direct physical contact (theft and borrowing) are of limited interest beyond achieving due diligence (giving the customer tools to protect their credentials from family/colleagues) mainly due to lack of scalability and ease of credential cancelling and reissue after theft.

We will return to this taxonomy of threats throughout the remainder of the document as we cover the mobile security architecture, and consider which features address which threats.

---

[1] Juniper Networks Malicious Mobile Threats Report 2010/2011

[2] *Rooting* and *Jailbreaking* refer to the owner gaining full administrative access to the phone in a way that circumvents the interests and security policy of the phone operating system manufacturer.

[3] It is worth noting that the landscape of threats in computer security in general has changed considerably since the arrival of the advanced persistent threat (APT), whose most unique and concerning characteristic is persistence of attack on a single target organisation, which may be the focus of attack for several years (the technology level of exploits/vulnerabilities in the attack rarely seems to be as high as initially thought).

Table 1

Threat	Goal/Resource	Notes
Malware attack	G: Monetary Gain R: Large black-market economy	Malware attacks remain the primary threat for mobile apps. Regardless of installation vector (phishing, app store poisoning, drive-by website) the result is similar and those deploying the attack are likely from the same criminal economy. Resistance comes from technical phone measures, user education and distribution channel policing.
Borrowed phone	G: Revenge, monetary gain R: Single layperson + commercial spyware market	The attacker might have brief direct access to the phone of a family member or colleague. Here the individual's resources are very limited but they may buy/licence quite advanced spyware. Best security is afforded through platform lockdown to prevent any type of spyware being installed, and user authentication before granting access (e.g. a PIN). Commercial spyware manufacturers can possibly be pressured to ensure their products cannot be used for stealing credentials (e.g. interception phone # black list, monitoring blacklists)
Stolen phone	G: Monetary Gain R: Small black-market economy	The attacker might steal the individual's phone either by mugging or pick pocketing. Here research shows that a majority of users will notice the theft within an hour, so the challenge is to ensure that credentials cannot be stolen, sold and abused all within the timeframe before reporting. Measures to damage efficiency of the criminal economy will help here. Some phones now have remote kill switches and tracking in addition.
Reputational attack	G: Perceived public good, anarchy R: Large organisation, top staff, limited budget	Researchers, pressure groups and lobbyists may take a dislike to a particular larger project (particularly those projects related to personal data centralisation and privacy) and attack the authentication mechanism as a way of highlighting risk or simply because it is there. Here what is important is that the architecture is seen to be secure and that security claims can be justified and defended. Likely the attack will come via the media. It may be necessary to prepare and brief spokespersons on the long-term mobile security strategy and to consider when briefing the difference between protecting the overall bottom line and the loss to the individual – assurance of fair dispute resolution mechanisms is important. The cruder (but also very effective) defence is stonewalling. Of most interest, resistance to this sort of attack comes from careful, clean and elegant engineering of the system, which has wide ranging design consequences.

3 Mobile Security Defence Strategy – an Iterative and Evolving Approach

3.1 The ever changing mobile ecosystem

It is important to appreciate the rate of change of software and hardware platforms in the phone industry. There are multiple incentive-aligned drivers for this rate of change:

- **Operating system vendors releasing new versions to:**
  - Close jailbreak and security loopholes that allow users to install unapproved software
  - Correct bugs or performance issues
  - Add new features to be innovative or match competition
- **Phone manufacturers**
  - Promoting their new handset models
  - Delivering more powerful CPU/GFX to the platform for gaming,
  - Necessitating API and OS updates

In average timescales there is a new iOS minor software version approximately every six weeks and a new Android version about every three months.

Given this natural rate of flux (which is also a source of unpredictability) it is perfectly reasonable to expect to update a security app to a new version several times a year, certainly once a quarter, if not more frequently. Mobile phone app stores can ensure that users are sufficiently reminded to go ahead and install updates, and the positive feedback of updates yielding new features or levels for games, and fixing issues related to new OS versions means the user is likely to make a positive choice to upgrade rather than to ignore it.

3.2 Malware and countermeasures

Meanwhile malware evolves at increased rates too, and especially when considering drive-by infections from websites (where the user is infected automatically upon visiting a site due to a browser vulnerability), the trend on the PC platform is toward polymorphic generation of a unique malware binary for each infectee, which has almost totally undermined virus infection statistics collection. On mobile the trend is toward automatic reverse engineering of existing applications, which adds malware to the app and then resubmits it to the store under a different name. The same attacker might submit 50-100 new apps to the marketplace, which rip off other apps in this way. Although the fake marketplace publishing account may be identified, the cost of manufacturing a new identity remains low (to encourage legitimate developers to participate), and the attacker may likely be in an uncooperative jurisdiction. Alternatively, the attacker might steal credentials from a legitimate but lapsing developer to launch the attack.

The future for mobile might also bring with it new hardware-backed security features, be it new TPMs (trusted platform modules), proprietary cryptoprocessors (such as found in the iPhone), SIMs or Secure Elements (SEs) which are part of the GlobalPlatform TEE trust architecture for mobile payments. Our measures take advantage of these emerging security technologies but with two important caveats.

3.3 Shared risk and access

The first caveat is the issue of shared risk, where if one relies on a security technology, which is used by another party for a different purpose, then you both share risk of compromise. An example is the shared infrastructure used for digital rights management (DRM), where the security mechanisms come under daily attack from organisations and groups that wish to freely pirate movies/games or simply to evade region lock or accessory control. This adversary may be far more powerful and widespread than the native criminal adversary, such as hacking banking authentication, and is able to operate with total impunity in multiple jurisdictions. Therefore, adopting a new security technology on the mobile platform may create shared risk depending on the source of the technology and this must be weighed up before making the decision on how to proceed.

The second caveat is the issue of negotiating access. A smart phone may have some secure capability, such as a secure element, but this might require cooperation of both handset manufacturer(s) and mobile network operator(s) in order to have access to loading a trusted application. Any iterative plan that considers relying on different and new secure elements must be mindful of all the probable challenges facing a company adopting the technology, from a business and economic perspective. This is particularly relevant to those creating payment or transaction authentication applications as other access-granting companies may demand payment in the form of a transaction fee. Thus for a new technology to be part of the plan there needs to be a credible route for it to become widely available, such that one does not become a hostage to the access-granting service provider.

3.4 Coming under attack

Once an app service is launched, how does one deal with the fact that some users start to come under attack, and how does one make sure the attacks are detectable? If an attack is a malware threat, it could be detected by monitoring app stores, transaction logs, AV reports or intelligence gathering as illustrated below.

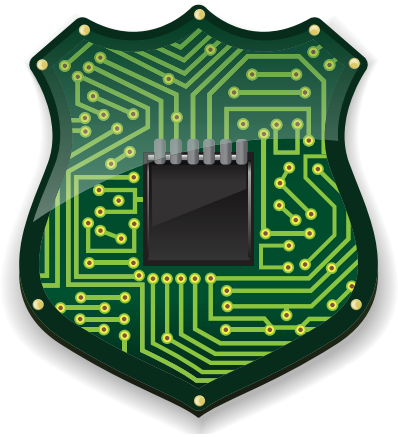
Due to the nature of the mobile platform, it is likely that root exploits of phones are readily available on the black market to attackers. Root exploit requires countermeasures to be deployed to limit the effectiveness of the malware from stealing credentials until the operating system vendor can patch the vulnerability and affected users can recover their phones.

Note that if the exploit was delivered by downloading a malicious app that used privilege escalation from an app store, the app store provider is in a position to collaborate and provide a list of all users who have downloaded both the authentication application and the malicious application. This provides the option for a very targeted security warning to be sent out, which should yield very few false positives. Malware infecting from website drive-by will not be enumerable in the same easy manner, but should be less frequent as it requires two exploits together – one to seize control through the web browser, and a second to escalate privileges to root.

DEFENCE STRATEGY WHEN COMING UNDER ATTACK

3.5 Limitations of secure coprocessors

Secure coprocessors used for holding crypto keys such as SIM cards, Secure Elements and TPMs may become available at some future phase of deployment. But assuming they are available, an important aspect to consider before utilising the coprocessor is whether the algorithm used for authentication has predictable inputs or not.



For example, time-based OTPs, counter-based OTPs and transaction signing all have predictable inputs. Therefore attackers can simply use temporary access to the security coprocessor, from having infected the main processor OS with malware, to harvest a codebook or dictionary of inputs and outputs, which represents everything they will need in the future to pretend to have possession of this key.

It is only where a truly random challenge number is provided by the authentication service (such as in a challenge response protocol) that the attacker has no idea in advance of an authentication attempt what data he will need to process using the stored key. Unfortunately moving to challenge/response authentication incurs significant usability penalties as the user must transfer the challenge onto the mobile device to be signed.

Thus for proper benefit a secure coprocessor needs to have specific security features available, and *coprocessors which simply store a key securely, but grant access to use it for any purpose, have little value*. Thus in the long term we expect to see limits on the effectiveness of secure coprocessors for defence, and their likely value may mainly be in obscurity and creating a new barrier for the attacker to reverse engineer against.

Additionally, in the long-term the secure coprocessor will need a trusted path to the user – to display the data which is about to be authenticated and to seek approval or rejection in a way which cannot be interfered with by the malware. However such a trusted UI is unlikely to be graphically very pretty and will grate considerably with the value proposition of modern smartphones, where visuals & aesthetics and usability are rated above all. For that reason it is unlikely that trusted UI proposals will garner a lot of support from the handset manufacturers though it is crucial to have them on board.

3.6 Common challenges for app developers today

Even though many app developers take different routes when developing apps with user authentication or transaction purposes, most will face the same security issues and challenges. They all need to ensure that the application offers a sufficient level of protection against malware, borrowed phones and reputational attacks on all supported platforms including, but not limited to, iOS and Android, which are very different in design.

This entails that they pay particular attention to:

- Building a secure yet convenient registration workflow
- Implementing reverse engineering resistance and introducing techniques such as anti-debugging, anti-tampering (modifying the app to patch out protections), anti-jailbreaking and emulation detection
- Preserving multi-channel security and ensure that apps and browsers run correctly on different devices to mitigate risks
- Storing in a secure manner customer credentials and sensitive key material
- Being able to uniquely identify devices and implement some device fingerprinting technique that cannot be reverse engineered easily
- Establishing a trustworthy connection to the back-end to be able to exchange data and ultimately sign transactions

3.7 The Cryptomathic mobile security strategy

The Cryptomathic mobile security strategy is an iterative evolutionary strategy. It is iterative in that it uses distinct phases of security improvements, and relies on frequent updates to software and protocols. It is evolutionary in that it adapts and responds to developing threats which due to the complexity of the market do not necessarily develop in a predictable way or according to a predictable timescale.

To address the above mentioned challenges in light of the mobile security strategy exposed above, Cryptomathic has implemented the world's most comprehensive, effective and evolutionary security API for mobile phone apps, namely the Cryptomathic Mobile App Security Core.

The next section introduces the reader to this ground-breaking concept.

4 Cryptomathic Mobile App Security Core

4.1 Introduction

Cryptomathic's Mobile App Security Core (MASC) provides technology for reverse-engineering resistance, malware detection, secure configuration and operation of generic mobile apps. It is aimed mainly at apps for transaction processing and other apps that need to perform cryptography. It features multiple layers of security, libraries for security protocols, TLS authentication with pinned certificates, and third party libraries integrated for malware detection, jailbreak detection and device fingerprinting.

Cryptomathic develops the majority of the security protections itself, but also partners with other security providers to combine and offer a full range of protection mechanisms for mobile apps in a single and cost-effective framework. In addition, Cryptomathic is constantly evaluating the effectiveness of the Mobile App Security Core and its security mechanisms using both in-house and third party reverse engineering capabilities.

4.2 Integration with existing environments

Cryptomathic MASC is a platform independent security layer that can be integrated with relatively little effort into existing apps and their corresponding back-end components.

It features different modules, which can be used either independently or jointly. In practice, each platform consists of a C core with thin wrappers implemented in Objective C and Java. The developers therefore still retain control over the UI design and may implement their own workflow with the security of the Cryptomathic Mobile App Security Core.

Looking more in detail, measures such as obfuscation, anti-debugging and anti-tampering can all be applied to a generic application without affecting functional interfaces and simply cause the application to disable (deletes its keys, shuts down or crashes) if it detects that it is being modified or run in a debug environment. These protections could be ported into nearly any app due to the lack of functional interaction – they either act simply as a wrapper layer or as a set of modules which can be independently embedded into the existing code. Anti-jailbreak and malware can also be integrated with relative ease so long as a defined local action can be taken on detection (e.g. deleting the sensitive key).

Moving beyond conventional smartphone environments, Cryptomathic also offers advice on how to mitigate risks via direct application interaction with the user interface to deliver protection methods such as anti-keylogging, screen scraping protection or UI-level anti-tampering. This topic is beyond the scope of the paper so please contact Cryptomathic for further information.

4.3 Security architecture and functionality overview

The Cryptomathic MASC is designed to perform the following security specific roles as shown below:

CRYPTOMATHIC MOBILE APP SECURITY CORE - FUNCTIONAL OVERVIEW

More under the hood, the Cryptomathic Mobile App Security Core contains a number of modules:

AuthManager
Authentication Manager

A subsystem which implements communications protocols for enrolment and logon. It contains state machines which govern the behaviour of these protocols, and draws together and uploads responses from other subsystems for back-end processing.

Anti-malware
The Anti-Malware Subsystem

Responsible for detecting device rooting/jailbreaking and the presence of malware. It uses software security mechanisms to detect and react to device compromise. It reacts to the presence of known malware which may interfere with the app and harvest data.

Broxy
Browser Proxy

A subsystem that acts as a gateway for application communication. It passes on requests, adds TLS encapsulation etc. The Broxy also integrates messages from the AuthManager to the back-end system.

Fingerprinting
Device Fingerprinting Module

This subsystem queries OS APIs to generate fingerprints from the device, and these are passed to the AuthManager for upload during enrolment and logon workflows, similarly to the malware detection messages.

SecSto
The Secure Storage Subsystem

Used by the AuthManager for holding the crypto keys used in authentication protocols and for holding application state. The subsystem uses obfuscation, secret-sharing, encryption and steganography to drive up the cost of recovering the data.

Obfuscation
Code and Data Obfuscation Module

The design includes obfuscation of stored data in file on disk which is designed to deter users from trivially extracting keys/data via analysing the stored files when running the app within an emulator or on a jailbroken/rooted phone.

Crypto
The Crypto Module

Based on Cryptomathic's off-the-shelf crypto toolkit called PrimeInk, and implements AES, RSA and SHA crypto functions required by the AuthManager.

Anti-debugging
The Anti-Debugging Module

Consists of a number of tests and includes specific countermeasures for debugging, library hooking/patching and sentinels which detect step-debugging and platform emulation.

TLS
Transport Layer Security Module

TLS for communications security is provided by an integrated SSL library. It implements pinned server certificate verification and has client certificate storage.

5 Conclusion

In short, the Mobile App Security Core delivers a foundation to enhance app security and support future technologies, without the need to expend extended time and costs redeveloping applications to support changing requirements. This ensures that mobile apps and their security framework remains future-proof and requires fewer resources to manage long-term.

We invite the reader interested in learning more to contact us so that we can expand on our secure techniques, or to test the methodology adopted to defeat attackers, as well as our patent pending security features.

Contact us on:

technical_enquiry@cryptomathic.com
enquiry@cryptomathic.com

or find our local offices:
http://www.cryptomathic.com/contact/cryptomathic-offices

Disclaimer

Jægergårdsgade 118, DK-8000 Aarhus C, Denmark

www.cryptomathic.com

ABOUT CRYPTOMATHIC

Cryptomathic is one of the world's leading providers of security solutions to businesses across a wide range of industry sectors, including finance, technology, government, mobile and cloud. With more than 25 years' experience, Cryptomathic provides customers with systems for e-banking, PKI initiatives, , ePassport, card issuing, mobile payments, advanced key management and managed cryptography utilizing best-of-breed security software and services. Cryptomathic prides itself on its strong technical expertise and unique market knowledge. Together with an established network of partners, Cryptomathic assists companies around the world with building security from requirement specification to implementation and delivery.