On mobile phone security

[Published at June, 2nd 2014 on https://pravokator.si*]

Author: Matej Kovačič

After the Snowden disclosures of the scope of NSA's surveillance, public interest in secure communications increased. The consequence of Snowden documents is development of several security applications and solutions for protection of e-mail, websites and other forms of internet communication. Mobile communications are not the exception, especially it is well known that GSM technology is vulnerable to several attacks, including eavesdropping, identity spoofing and manipulation of retention data.

One of the mobile security products is a special smartphone called Blackphone. It was released this year and is advertised as "*the world's first smartphone to put privacy and control ahead of everything else*". Some media even presented it as a "NSA-proof" mobile phone. However, creating a truly secure mobile phone is... well, impossible with the existing technology.

The problem is, that smart mobile phones are basically the computers, but they do not have all the (hardware) security mechanisms which are present in the modern computers. And even more important, mobile technology has some serious design flaws which is preventing the highest level of their security.

Smart phone security today is focused mainly in three areas whose aims are **to prevent unauthorized remote access to data**, **unauthorized local (physical) access to data** and **unauthorized access to communications** and **traffic data**.

Key solution for preventing unauthorized remote access is secure or "security hardened" operating system, which is usually some fork of Android (for instance *Replicant*, Blackphone's *PrivatOS*, *Guardian ROM*, *CyanogenMod*, etc.). Actually we are not talking just about OS, but more about the complete distribution, because this modified and privacy-enhanced OS has usually integrated several privacy protecting applications, which are preinstalled and preconfigured, for instance to prevent unauthorized access, to encrypt all SMS messages, to redirect all traffic to VPN, and so on.

Many of privacy enhancing technologies are already a part of an Android system. For instance, for **preventing unauthorized physical access to data** (for lost or stolen device) user can set up a screen lock (with PIN, password, pattern unlock or something else). Android itself has built-in countermeasures to prevent trying every possible password to unlock the mobile phone (with rate limiting and after some amount of unsuccessful tries it will usually lock the mobile phone or completely erase the data on it).

Another important technology to **prevent physical access to data** is encryption, which is protecting the internal memory of the mobile device. Unfortunately, this **encryption implementation in Android has a serious security flaw**, that is, it uses the same password for screen lock and internal memory encryption.

As mentioned, Android has some good countermeasures for preventing brute force attack (trying every possible password) on screen lock. Because of that, even a relatively short password offers

* https://pravokator.si/index.php/2014/06/02/on-mobile-phone-security/

adequate protection. But if an attacker gets physical access to the device and is able to make a memory image of it, he or she can try every possible password without worrying about rate-limiting or device lock. Screen lock password and hence internal memory encryption password is limited to maximum 16 characters, which is adequate protection of screen lock, but absolutely deficient protection for internal memory encryption. And even if there would be no limit on password length, the problem will persist – it is not reasonable to expect users would be using very long and complex password for screen lock, because they would need to enter this password each time they would need to unlock the screen, which is usually several times a day. Anyway, it is important to note, that it is possible to set up different password for encryption of internal memory and for a screen lock, but in that case rooted Android device is needed and user must take additional effort to implement this feature.

The last set of technologies whose aim is to prevent unauthorized access to the content of communications and in some cases even traffic data, consist of applications which try to provide **secure, end-to-end encrypted communication over the network**. For voice encryption is usually used data transfer (ZRTP encrypted VoIP communications), while encrypted messages could be sent as data or over an "plain old" GSM carrier as encrypted SMS. Some advanced solutions even try to apply some **anonymisation** (for instance *ChatSecure* can connect to a *Tor* network) or at least try to **hide traffic data from the mobile operator**. Typical example are applications which encrypt it's communication with some server. That (encrypted) data are hidden from the mobile network operator, but on the other side server operator can see them.

While those three approaches look sufficient on the first sight, they actually **do not solve the problem of privacy**. The problem of privacy and security is still here – it is just **moved a few layers below the surface**. The main reason for that is, that mobile telephony **cannot be really secure**, because the **technology itself has some serious design flaws**. Let's take a look at some of them.

Network authentication problem and IMSI Catchers

Historically, GSM technology has been developed in a way, that all logic was on the network side and mobile devices were considered as "stupid clients" only. One typical example of this is a fact, that GSM standard contains several measures for a network to authenticate a mobile device, while **a mobile device can not authenticate network** (this changed only with 4G/LTE a few years ago).

This design feature has some serious consequences for privacy of mobile phones users. Because mobile device cannot authenticate mobile network, **it is possible to plant a fake base station which pretends to be a part of legitimate network**, and then hijack all the traffic of the victim's device. This device is called **IMSI Catcher**. It is special device, which is usually used by law enforcement and secret services for a control over target GSM devices. But there are also some open source projects, for instance OpenBTS, which could be used for building your own IMSI Catcher. A member of Osmocom-BB project (which is developing open source GSM baseband software implementation), *Sylvain Munaut* has shown how to turn an old mobile phone with Calypso chipset into a base station. From there to a real IMSI Cather is just a small step.

While it is not possible to avoid this problem with the existing technology, it could be possible to build an IMSI Catcher **detector**. Actually, there are two interesting projects going into this direction. One is SRLabs' Cather Cather (it is running only on a Osmocom-BB devices and is not suitable for "ordinary" users) and the second is Android IMSI-Catcher Detector (AIMSICD). However, none of the projects are finished and "production ready".

Associated problem is **detection of silent SMS messages**. Silent SMS messages are usually employed to locate a mobile phone (but can also be used to get some other information from the target mobile device, for instance the data needed for cryptanalysis of GSM encryption). They do not show up on a display of a target device, nor trigger any acoustical signal when received. But when they are delivered they generate a delivery receipt and – the most important – they are recorded in a data retention database together with the location of a mobile phone which received it. The scope of the use of silent messages could be demonstrated by the fact that in Germany in 2010 almost half a million silent SMS messages were sent by the law enforcement authorities.

IMSI Catcher detector software should have the feature to detect silent SMS'es and this feature is already implemented in SRLabs' Cather Cather. While this countermeasure does not prevent user from being tracked it is still usable because it warns user that he or she is under an active attack.

Identification of a mobile phone and location privacy

Another serious problem for a mobile privacy is **location tracking**. Documents revealed by The Washington Post has shown how NSA is using location tracking and how they can identify so called co-travelers – unknown associates who might be traveling with, or meeting up with a their known target.

It is not possible to mitigate that problem, unless an user decides to **switch off from the mobile network completely**. While this destroys the main aim of mobile communications, it is also **not always effective in practice**. Simply removing SIM card from the mobile phone is not a viable solution, because even in that case, a mobile phone **is sending out broadcasts** looking for a base station, and those broadcasts include unique ID's like IMEI number of a mobile device. This kind of tracking is actually used by the CIA and NSA for their target killings with the drones.

While it is technically possible to change IMEI number on **some** mobile phones, it is important to notice that this procedure is **quite complicated and illegal** in many countries.

Another solution would be using Wifi connections only. While Wifi device on a mobile phone **still broadcasts an unique ID** (the MAC address of the wireless card), it is possible to use a special application for changing this identifier. These applications are usually called **MAC changers**, because they are changing the MAC address of a mobile phone every time the device uses a different Wifi or at a specific time interval. That will make mobile tracking **much more difficult**, but it is **not usable** in environments where there is low density of Wi-fi hotspots or where Wi-fi requires some kind of **authentication or subscription**. And of course, we must not forget to **switch off a Bluetooth device**.

It is also important to note, that applications for changing MAC addresses (for instance Wireless Mac Address Changer, MAC Spoofer, etc.) **require rooted devices and are not working on all mobile devices**.

Anyway, location tracking is basically a problem of several unique identifiers broadcasted by a mobile device, which is in fact a design feature. It is simply very hard or almost **impossible to prevent mobile phone from emanating one or another unique identifier**. This is a serious privacy problem, which is almost completely overlooked by developers of secure communications.

Attacks on the internal memory of mobile phones

Another class of the attacks on a smart mobile phones are attacks which could be used to **access phone storage or to inject malware**. Some of the attacks works **even if user is using encryption**, firewall, or other privacy enhancing technologies.

One possible attacks on encryption of the mobile phone's internal memory is ***Cold Boot Attack***. With this attack, the **attacker could obtain the encryption key and thus decrypt phone storage**. Attack is not very easy to implement, but security researchers *Tilo Müller*, *Michael Spreitzenbarth* and *Felix Freiling* from *Erlangen's Friedrich-Alexander University* have demonstrated that the *Cold Boot attack* is feasible with relatively simple equipment. They have cooled mobile phone to -15 degrees Celsius, and then used their own tool FROST (*Forensic Recovery Of Scrambled Telephones*) to extract the encryption key from mobile's internal memory.

Cold Boot Attack is known attack against ordinary PCs. But unlike for PCs, where there are at least some theoretical possibilities of protection, mobile phones do not have any. Reserchers *Tilo Müller*, *Felix C. Freiling* and *Andreas Dewald* have shown that in PC, it is possible to store the encryption key inside CPU registers rather than in RAM. All computations then take place only on x86 debug registers, and no sensitive encryption material is ever going to RAM. But as mentioned, this solution is currently available for a PCs only and even there mostly as a proof-of-concept. Slightly similar technology is PrivateCore vCage technology which uses *Trusted Platform Module* (TPM) and *Intel Trusted Execution Technology* (TXT) for validation of computer hardware and to encrypt the RAM content (vCage Host loads a secure hypervisor into CPU cache which is encrypting/decrypting memory paging in and out between the CPU cache and RAM).

Another possible attack to mobile phone internal memory is so called ***Evil Maid Attack***. In that case **the attacker needs physical access to the system**, where he or she installs hardware **or** software keylogger. The attacker is then able to reveal plain text password and therefore decrypt the encrypted data. Information security expert *Joanna Rutkowska* has demonstrated practical *Evil Maid attack* against TrueCrypt protected computer in 2009.

The same attack could also be performed **on mobile phones**. In July 2012, *Thomas Cannon* from the *Viaforensics* introduced the idea how to perform Evil Maid attack on mobile phones. He revealed that it is possible to **install a rootkit on the system partition of a mobile phone**. That rootkit can intercept passwords or allow an attacker to remotely access user's smartphone. It can be automatically loaded via special tailored (evil) mobile USB charger.

There are some technologies to mitigate that problem, though. Android 4.4 introduced a security feature called **verified boot**. The main purpose of verified boot is to assure that a **mobile device which is booting is in the same file system state as when it was last used**. Verified boot is therefore **solving the problem of malware with root privileges** which can hide from detection. An application implementing this feature is called *dm-verity*, but this new Android feature is still an experimental one.

Verified boot tries to solve *Evil Maid Attack* problem, however, it is not a complete solution for it. When a *dm-verity* device is configured, it is expected that the caller has been properly authenticated and the bootloader has not been tampered. *Dm-verity* also can not detect a trojan placed in the *ARM Trustzone* (the Security Extensions of ARM processors) or on a SIM card. While this solution is a good step toward better mobile security, it has some limitations we must be aware of.

There are of course some other attacks, including **planting malicious applications**, **planting fake digital signatures** on a malware, etc. (more on this topic could be read in *Jeff Forristal* presentation One Root to Own Them All).

Protection against such attacks is an **appropriate "safety culture"** (physical security, caution when installing applications, regular system and security updates, etc.). *Evil Maid attack* could also be mitigated by using smart cards or hardware tokens and booting the system from a trusted medium (more on that in Joanna Rutkowska blog). Unfortunately, these solutions were developed for personal computers only. **Mobile security is a few steps behind that.**

Attacks over the radio processor

The last topic we will address here is the **security of baseband processor**. Mobile phones basically contain two types of processors. The first type represents the so called **application processor** (general-purpose applications processor), which run the operating system and user applications. But in addition to that processor, mobile phone also contain so called **radio processor** (baseband processor, also called modem or radio), which runs real time OS (in mobile networks the correct timing is very important for a communication between the mobile phone and the base station). This real time OS is basically closed-source operating system and cannot be changed. It is important to note that radio processor is primary processor and usually has full control over all the other hardware (including camera, microphone, screen, etc.) and is not sandboxed from the phone operating system. Some smartphones even use a shared memory architecture to transfer data from the baseband processor and application processor.

Let us also mention that there are only six manufacturers of radio processors, which are used in all modern smart phones. A majority market share has two of them. That means that a potential attacker, which is able to find backdoors or can even insert malicious code in a production of chips in a few selected production facilities, can gain access to virtually all mobile phones in the world.

Moreover, as *Ralf-Philipp Weinman* from the *University of Luxembourg* has shown, **radio processors contain several security vulnerabilities**. It is even possible to access a mobile phone from the network (through radio processor). His presentation titled The Baseband Apocalypse is worth listening. The same applies for his article Baseband Attacks: Remote Exploitation of Memory Corruptions in Cellular Protocol Stacks.

There are known cases, where law enforcement agencies used a covert listening device (a so called "*roving bug*") in their investigations. One of the first known cases is from 2006. In that case FBI remotely activated the cell phone microphone and listened to conversations in the vicinity of the phone, during its criminal investigation.

From the security point of view it is also important a discovery of the developers of the *Replicant* operating system. *Replicant* is an operating system for mobile phones based on Android. But unlike Android, it is **completely open (including drivers)** and does not contain any proprietary code. The group of developers who are working on *Replicant*, recently found that ***Android* running on a *Samsung Galaxy* mobile phones** contains backdoor. They have found a specific software code that allows **direct communication with the radio processor** in a way that radio processor **has direct access to the Android file system** (reading, writing and erasing files). That software is able to access the user data, even if they are encrypted (because it is accessing the data while the file system is unlocked).

Replicant OS of course does not contain that backdoor code, but due to the fact that radio processor is "primary", it is still able to take full control control of a mobile device. The problem is not only in software, **the problem is in hardware and it is also a design feature**.

And yes, there is another one processor in a mobile phone – it is located on the SIM card. And yes, as Karsten Nohl has shown, SIM cards are also vulnerable. Surprised?

Conclusion

As we have shown, mobile security is **far from being easy**. Using some security application for encryption of voice and messaging **is not a complete solution for a mobile privacy**, because the problems of mobile security are far below this layer. And vulnerable are not only mobile phones, but also the core networks of the mobile providers (there is well known wiretapping case which happened in Greece in 2004-2005, where unknown perpetrators compromised mobile carrier's network equipment) and even the lawful interception equipment (case from Italy and recent case from Israeli software company that sells a call recording solutions for law enforcement).

Of course, it depends on how far you want to go and what is your threat model. If you want to protect yourself from some neighbor hackers or some rogue system administrator from your mobile carrier, using of *TextSecure*, *ChatSecure*, *RedPhone*, *OpenVPN* and similar application is a good solution. But if your threat model is NSA or FSB... well, then maybe throwing away your mobile phone is the only considerable option.

Matej Kovačič is an independent security researcher from Slovenia, Europe. He is interested in mobile phone security and has co-authored a research on GSM security and the reliability of data retention in Slovenian GSM networks. Currently is working as a head of analytics and information security at the Commission for the Prevention of Corruption, but is always looking for new opportunities. You can check his complete CV here.