Android full-disk encryption: a security assessment

Oliver Kunz

Abstract:

Mobile phones evolved from basic telecommunication devices to smartphones which are, in essence, pocket computers. With this technological evolution their usage also changed. Nowadays users do not just keep contact details and text messages but also e-mails, chat communications, documents, browsing history and other data stored on their mobiles.

Different actors are interested in this data: criminals, competitors, as well as law enforcers. In 2014, Google announced that it would enable Android’s full-disk encryption by default. Encryption is a dual-used good, since it protects data from invasive third parties but at the same time there are situations where law-supported third parties need access to that data. Law enforcers face increasing difficulties in collecting evidence for the prosecution of criminals.

This project aims to understand how Android’s full-disk encryption feature is implemented in the broader context of the framework and to assess the security of this feature. We analyse the source code which provides the encryption functionality, manages the encryption keys and triggers decryption on device boot. We researched known vulnerabilities to cryptographic primitives employed as well as to similar full-disk encryption implementations in Linux and Android.

According to Google, Android 5.0 was improved to prevent Offline Exhaustive Password Search Attacks to recover the screen lock method which is also used to protect the master key for disk encryption. In our research we confirm this statement while presenting an alternative attack approach we call Semi-Offline Exhaustive Password Search Attack. In contrast to the Offline approach, the smartphone is used for a particular step in the attack, hence the name Semi-Offline. Our attack takes five times longer than the Offline attack but is more than ten times faster than an Online Exhaustive Password Search Attack.

The threat model covered by Android, as we identified, only protects data at rest. This requires the smartphone to be shut down, a rather rare state for mobile phones. We therefore included in our assessment the attack scenario Device-ON. With transparent encryption, each disk read is decrypted and each write encrypted. An attacker who manages to bypass lock-screen authentication therefore has full access to the otherwise encrypted data.

We assessed the misuse potential of Smart Lock, an authorized lock-screen bypass feature introduced in Android 5.0, based on configurable trust agents. A trust agent can for example be a location, device, face or body movement. We were able to demonstrate in three of the four categories how an adversary can misuse the feature and described a potential scenario for the last.

For various attacks and vulnerabilities, we proposed new countermeasures or improvements to existing ones.