ANALYSIS OF BUILT-IN KEY STORAGE MECHANISMS IN THE ANDROID OPERATING SYSTEM

Balagura D.S, Sydorenko Z.M.
Kharkiv National University of Radioelectronics, Kharkiv, Ukraine

The Android operating system is one of the most prevalent operating systems worldwide. Based on open sources, as of 2023, over 3 billion devices globally were operating under this operating system. It is evident that Android-powered devices are utilized across various domains, ranging from mobile communication and IoT systems to the management of technological processes within device groups. Naturally, a substantial portion of these devices may store and process information that requires protection against unauthorized access or modification through the use of cryptographic algorithms. A crucial aspect of employing cryptographic algorithms is the storage of private keys [1-3]. Android provides various mechanisms for secure key storage in the operating system itself or on the device. Developers may choose an approach based on their specific use case. **The work provides** some analysis of general methods for secure private key storage in Android OS which uses OS mechanisms and/or hardware platform capabilities.

Android Keystore System is a system-provided cryptographic API that allows developers to store and retrieve cryptographic keys securely. It provides developers with a secure container (keystore) for key storage. All keys stored in the keystore are protected against extraction. Android Keystore offers hardware-backed security, if available on the device. While no system can be considered completely invulnerable, the Android Keystore is designed to provide a high level of security for cryptographic key storage on devices developed for Android OS.

Secure Elements or Trusted Execution Environments (TEE): Some devices support secure elements or TEEs to store cryptographic keys securely. These environments are designed to be isolated from the main parts of operating system and resistant to attacks. If you need to store small pieces of sensitive information like keys, using of Secure Shared Preferences with encryption is one of the options as well Secure Shared Preferences is an approach in Android application development to enhance the security of storing sensitive information such as user preferences, authentication tokens, or encryption keys. The standard Shared Preferences in Android allows developers to store and retrieve key-value pairs, but it doesn't provide inherent encryption, making the stored data susceptible to unauthorized access. Secure Shared Preferences aims to address this limitation by introducing encryption mechanisms to protect the stored data.

References

1. "Android Cookbook: Problems and Solutions for Android Developers 2nd Edition, O'Reilly, Ian F. Darwin, 2017

2. "Android Security Internals ", No Starch Press, Inc, Nikolay Elenkov, 2015

3. Trusted Execution Environment a Complete Guide, The Art of Service, 2021