Digital Financial Services Mobile Application Security Best Practices

Table of Contents

1 Acronyms

AES	Advanced Encryption Standard
CA	Certificate Authority
DFS	Digital Financial Services
ENISA	European Union Agency for Cybersecurity
GCM	Galois/Counter Mode
GSMA	GSM Association
ITU	International Telecommunication Union
OTP	One Time Password
SE	Secure Element - A formally certified, tamper-resistant, stand-alone integrated circuit often referred to as a "chip" as defined by the European Payments Council or other recognized standards authority.
SHA	Secure Hash Algorithms
SIM	Subscriber Identity Module
SMS	Short Messaging Service
SQL	Structured Query Language
SS7	Signalling System No.7
SSL	Secure Sockets Layer
STK	SIM Application Toolkit
TLS	Transport Layer Security

2 Introduction

The template for application security proposes best practices that Digital Financial Services regulators which could be included in an app security policy document by DFS providers. The template strictly considers the mobile application on the device unless stated otherwise, and subsections describing recommendations deal with various aspects of the operation or underlying policy relating to the mobile application. The focus is primarily on Android applications given their large market share, though many recommendations are applicable across mobile operating systems. This template is extracted from Appendix 1 of **ITU-T Recommendation X.1150 : Security assurance framework for digital financial services**

The focus here is on general best practices and not specific individual technologies except where explicitly discussed. For this template, we draw on recent works on examining digital financial services applications from the standpoint of the mobile money application space, including the GSMA study on mobile money app security best practices,[1] the ENISA smartphone secure development guidelines,[2] and a mobile payment applications security framework developed by the State Bank of Pakistan.[3] This template can also be used also as input to an app security policy by DFS Providers.

In this section, we summarize the recommendations as a starting point for regulators or application security examiners to perform security assessments. The template strictly considers the mobile application on the device unless stated otherwise, and subsections describing recommendations deal with various aspects of the operation or underlying policy relating the mobile application. The focus is primarily on Android applications given their large market share, though many recommendations are applicable across mobile operating systems. Privacy is also an important factor to consider, but these recommendations focus on security.

3.1 Device and Application Integrity

i. The safest devices for performing financial transactions on are ones that have not been "jailbroken" or "rooted", as it can be difficult or impossible to assess the security of the underlying operating system when it has been replaced or exploited. Applications should thus use the mobile platform services to determine that they and the underlying platform have not been modified.
ii. Remove any extraneous code that might have been added to the application during development, such as features that are not designed for the device platforms that the app is to be deployed upon or developer/debug features to reduce the attack surface of the deployed production code.
iii. On the server-side, determine whether the app is running in a high integrity state through signature validation or hashing over the app or certain program function blocks.

[1] GSM Association, Official Document MM.01 – MM App Security Best Practices, Version 1.0, 28 June 2018.
[2] European Union Agency for Cybersecurity (ENISA), Smartphone Secure Development Guidelines, 10 February 2017.
[3] State Bank of Pakistan, Mobile Payment Applications (App) Security Framework (DRAFT version 1.0), April 2019.

3.2 Communication Security and Certificate Handling

i. Apps should be making use of standardized cryptographic libraries and for communication with back-end services, should use end-to-end encryption with standardized protocols, specifically transport layer security (TLS). The minimum recommended version of TLS that should be used is version
ii. TLS certificates should not be expired and should present strong cipher suites, specifically AES-128 encryption and SHA-256 for hashing. Authenticated encryption modes of operation such as Galois/Counter mode (GCM) are encouraged.
iii. Limit the lifetime of issued certificates to 825 days in accordance with the CA/Browser Forum best practices.
iv. Assure the trustworthiness of the certificate authority and consider a contingency plan for if the CA is no longer trusted.
v. Ensure the configuration of TLS is performed in a secure fashion and avoid misconfiguration issues that could result in failure to authenticate or poor algorithm selection.
vi. Certificate pinning should prevent replacement of certificates where it refers to the process of associating a host with their expected public key certificate or public key. Once a certificate or public key is known or seen for a host, the certificate or public key is associated or 'pinned' to the host.
vii. Client devices shall ensure that they correctly validate server certificates

3.3 User Authentication

i. PINs and passwords should not be easily guessable and weak credentials should be disallowed; however, users should not be forced to change passwords on a regular basis.
ii. Multi-factor authentication before performing financial or other sensitive functions should be strongly encouraged. A phishing-resistant form of multi factor authentication [b-ITU-T X.1277] [b-ITU-T X.1278] should be used.
iii. Smartphone authenticator apps should be used for sending one-time passwords rather than SMS due to the possibility of SS7 hijacking and other insecurities.
iv. If biometric information is used for authentication, it shall be stored with appropriate security measures such as encrypted in the Android Keystore or with the use of trusted hardware.

3.4 Secure Data Handling

i. Mobile devices should securely store confidential information.
ii. Trusted hardware should be used for the storage of sensitive information if it is available on client smartphones.
iii. Avoid storing information in external storage and if it is done, ensure that strong input validation is performed prior to using this data.
iv. Delete confidential data from caches and memory after it is used and avoid general exposure of information (e.g., placing the secret key on the stack). Assure the clean-up of memory prior to the application exiting.

v. Restrict data shared with other applications through fine-grained permissions. Minimize the number of permissions requested by the app and ensure that the permissions correlate to functionality required for the app to work.
vi. Do not hard-code sensitive information such as passwords or keys into the application source code.
vii. Validate any input coming from the client that is to be stored in databases to avoid SQL injection attacks.

3.5 Secure Application Development

i. Develop applications according to industry-accepted secure coding practices and standards.
ii. Assure a means of securely updating applications and assure that all dependent libraries and modules are secure; provide updates for these when required.
iii. Have code independently assessed and tested by internal or external code review teams