Hypori Virtual Mobile Infrastructure Platform 4.2.0 Client (Android) Security Target

Version 1.0
January 12, 2021

Prepared by:
Hypori, LLC.
1801 Robert Fulton Drive, Suite 440
Reston, VA 20191

Copyright

LIST OF TABLES

1. Security Target Introduction

This section identifies the Target of Evaluation (TOE) along with identification of the Security Target (ST) itself. The section includes documentation organization, ST conformance claims, and ST conventions.

The TOE is the Hypori Client (Android) component of the Virtual Mobile Infrastructure Platform version 4.2 provided by Hypori, LLC.

The Security Target contains the following additional sections:

- Security Target Introduction (Section 1)
- TOE Description (Section 2)
- Security Problem Definition (Section 3)
- Security Objectives (Section 4)
- IT Security Requirements (Section 5)
- TOE Summary Specification (Section 6)
- Protection Profile Claims (Section 7)
- Rationale (Section 8).
- Appendix: Android APIs (Section 9).

1.1 Security Target, TOE and CC Identification

ST Title – Hypori Virtual Mobile Infrastructure Platform 4.2.0 Client (Android) Security Target

ST Version – Version 1.0

ST Date – January 12, 2021

TOE Identification – Hypori Client (Android) 4.2.0

TOE Developer – Hypori, LLC.

Evaluation Sponsor – Hypori, LLC.

CC Identification – *Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 5, April 2017*

1.2 Conformance Claims

This TOE is conformant to the following CC specifications:

This ST is conformant to the *Protection Profile for Application Software*, Version 1.3, 2019-03-01 [PP_APP_v1.3].

The following NIAP Technical Decisions apply to the security target or the evaluation assurance activities.

- TD0561: Signature verification update
- TD0554: iOS/iPadOS/Android AppSW Virus Scan
- TD0548: Integrity for installation tests in AppSW PP 1.3
- TD0544: Alternative testing methods for FPT_AEX_EXT.1.1
- TD0521: Updates to Certificate Revocation (FIA_X509_EXT.1)
- TD0515: Use Android APK manifest in test
- TD0498: Application Software PP Security Objectives and Requirements Rationale
- TD0495: FIA_X509_EXT.1.2 Test Clarification
- TD0486: Removal of PP-Module for VPN Clients from allowed with list
- TD0445: User Modifiable File Definition

- TD0444: IPsec selections
- TD0437: Supported Configuration Mechanism
- TD0427: Reliable Time Source
- TD0416: Correction to FCS_RBG_EXT.1 Test Activity

The following NIAP Technical Decisions are list on the NIAP website, but are not applicable to this evaluation:

- TD0543: FMT_MEC_EXT.1 evaluation activity update
- TD0540: Expanded AES Modes in FCS_COP
- TD0519: Linux symbolic links and FMT_CFG_EXT.1
- TD0510: Obtaining random bytes for iOS/macOS
- TD0473: Support for Client or Server TOEs in FCS_HTTPS_EXT
- TD0465: Configuration Storage for .NET Apps
- TD0435: Alternative to SELinux for FPT_AEX_EXT.1.3
- TD0434: Windows Desktop Applications Test

Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1, Revision 5, April 2017.

- Part 2 Extended

Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1 Revision 5, April 2017.

- Part 3 Extended

1.3 Conventions

The following conventions have been applied in this document:

- Security Functional Requirements – Part 2 of the CC defines the approved set of operations that may be applied to functional requirements: iteration, assignment, selection, and refinement.
  - Iteration: allows a component to be used more than once with varying operations. In the ST, iteration is indicated by a number in parentheses placed at the end of the component. For example, FDP_ACC.1(1) and FDP_ACC.1(2) indicate that the ST includes two iterations of the FDP_ACC.1 requirement, (1) and (2).
  - Assignment: allows the specification of an identified parameter. Assignments are indicated using bold and are surrounded by brackets (e.g., [**assignment**]). Note that an assignment within a selection would be identified in italics and with embedded bold brackets (e.g., [***[selected-assignment]***]).
  - Selection: allows the specification of one or more elements from a list. Selections are indicated using bold italics and are surrounded by brackets (e.g., [***selection***]).
  - Refinement: allows the addition of details. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… ~~some~~ **big** things …"). Note that 'cases' that are not applicable in a given SFR have simply been removed without any explicit identification.
- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.

1.3.1 Terminology

[PP_APP_v1.3] provides definitions for terms specific to the application software technology as well as general Common Criteria terms. The technology-specific terms are:

- Address Space Layout Randomization
- Application
- Application Programming Interface
- Credential
- Data Execution Prevention
- Developer
- Mobile Code
- Operating System
- Personally Identifiable Information
- Platform
- Sensitive Data
- Stack Cookie
- Vendor

Terms from the Common Criteria are:

- Common Criteria
- Common Evaluation Methodology
- Protection Profile
- Security Target
- Target of Evaluation
- TOE Security Functionality
- TOE Summary Specification
- Security Functional Requirement
- Security Assurance Requirement

This ST does not include additional technology-specific terminology.

1.3.2 Abbreviations

This section identifies abbreviations and acronyms used in this ST.

API	Application Programming Interface
App	Software application
ASLR	Address Space Layout Randomization
CC	Common Criteria
CEM	Common Evaluation Methodology
CTLs	Certificate Trust Lists
DEP	Data Execution Prevention
DoD	Department of Defense
OS	Operating System
PII	Personally Identifiable Information
PP	Protection Profile
PP_APP_v1.3	Protection Profile for Application Software
SAR	Security assurance requirement

SFR	Security functional requirement
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Functionality
TSS	TOE Summary Specification
VMI	Virtual Mobile Infrastructure

2. TOE Description

After a brief overview of the Hypori Virtual Mobile Infrastructure product, this section describes its Hypori Client (Android) component, which is the Target of Evaluation (TOE). The description covers TOE architecture, logical boundaries, and physical boundaries.

2.1 Product Overview

In the Hypori Virtual Mobile Infrastructure (VMI) platform, end users running a Hypori Client (Android) on their mobile device access a virtual Android device running on a server in the cloud. The virtual device on the server contains the operating system, the data, and the applications, using TLS 1.2 encryption to communicate securely with the Hypori Client (Android). The Hypori Android thin client application provides secure access to the remote Android virtual device and brokers access between the mobile device's sensors and the applications executing in the virtual device on a Hypori server. The client applications are agnostic to the version of Android executing in the virtual device.

The following diagram shows the Hypori system, including its components and networks. Unlike many software solutions, some of the Hypori servers are installed on virtual servers while others are installed on physical servers.

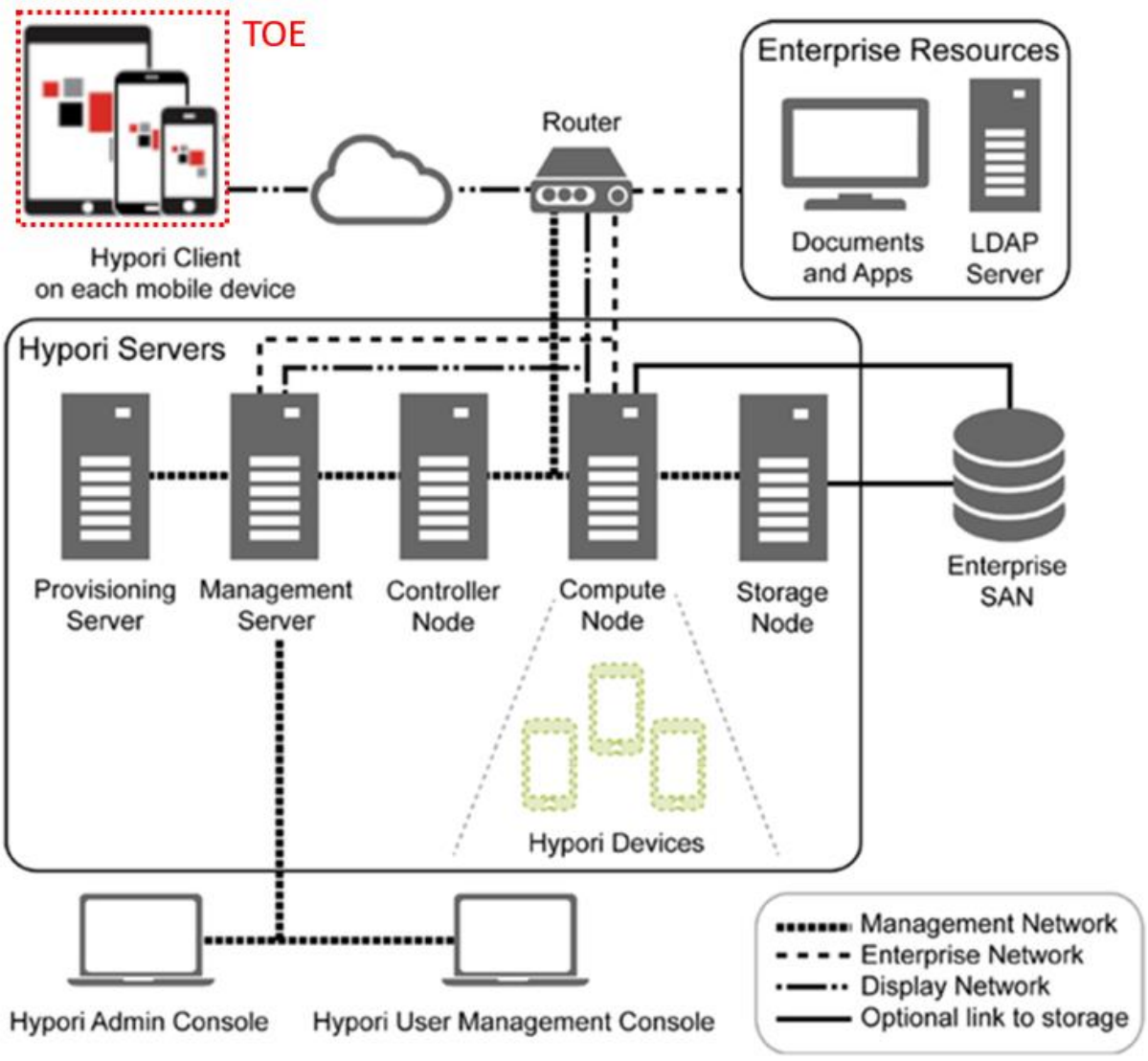

Figure 1 Hypori Virtual Mobile Infrastructure (VMI)

The Hypori VMI platform includes the following components:

- **Hypori Client:** This is an Android-based thin client that installs on the end user's mobile device and communicates with the Hypori Virtual Device on the server through secure encrypted protocols.
- **Hypori Virtual Device:** This is an Android-based virtualized mobile device executing on a server in the cloud.
- **Hypori Servers:** This is the cloud server cluster that hosts the Hypori Virtual Devices.
- **Hypori Admin Console:** This is a browser-based administration user interface that is used to manage the Hypori system.

2.2 TOE Overview

The TOE is the Android-based Hypori Client. The following diagram shows how the TOE interacts with a Hypori Device running applications on a Hypori Server. The Hypori Client is a thin client that communicates only with a Hypori Virtual Device on a Hypori Server and not with other servers or applications.

Figure 2 Hypori Client as Part of VMI Platform

2.3 TOE Architecture

The section describes the TOE architecture including physical and logical boundaries. Figure 3 shows the relationship of the TOE to its operational environment along with the TOE boundary. The security functional requirements identify the libraries included in the application package.

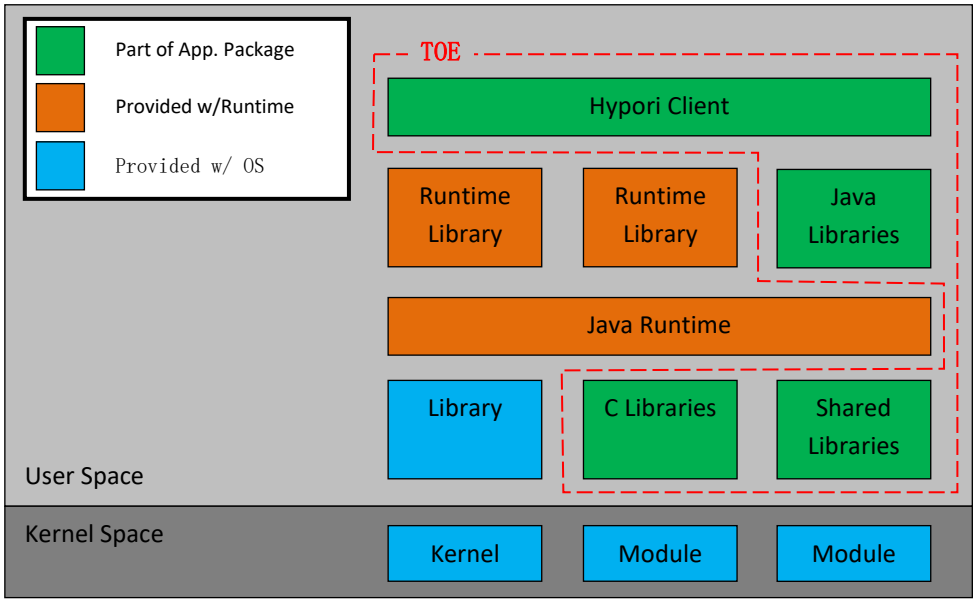

Figure 3 TOE Boundary for Android Devices

2.3.1 Physical Boundaries

The TOE consists of a Hypori Client application as defined in the Hypori Client installation package. The Hypori Client is an Android-based thin client that only communicates with the Hypori server. The Hypori server, applications running on the Hypori server, and any functions not specified in this security target are outside the scope of the TOE.

2.3.1.1 Software Requirements

The TOE was evaluated on Android releases 8.1, 9, and 10.

2.3.1.2 Hardware Requirements

The TOE imposes no hardware requirements beyond the Android operating system requirements.

2.3.2 Logical Boundaries

This section summarizes the security functions provided by the TOE:

- Cryptographic support
- User data protection
- Identification and Authentication
- Security management
- Privacy
- Protection of the TSF
- Trusted path/channels

2.3.2.1 Cryptographic support

The TOE establishes secure communication with the Hypori server using TLS. The client uses cryptographic services provided by the platform. TOE stores credentials and certificates for mutual authentication in the platform's key chain.

2.3.2.2 User data protection

The TOE informs a user of hardware and software resources the TOE accesses. It uses the platform's permission mechanism to get a user's approval for access as part of the installation process. The user initiates a secure network connection to the Hypori server using the TOE. In general, sensitive data resides on the Hypori server and not the Hypori Client, although the client does store credentials as per section 2.3.2.1.

2.3.2.3 Identification and Authentication

The TOE uses the platform's certification validation services to authenticate the X.509 certificate the Hypori server presents as part of establishing a TLS connection.

2.3.2.4 Security management

Security management consists of setting Hypori Client configuration options. The TOE uses the platform's mechanisms for storing the configuration settings.

2.3.2.5 Privacy

The TOE does not transmit PII over a network.

2.3.2.6 Protection of the TSF

The TOE uses security features and APIs that the platform provides. The TOE leverages package management for secure installation and updates. The TOE package includes only those third-party libraries necessary for its intended operation.

2.3.2.7 Trusted path/channels

The TOE invokes the platform-provided functionality to encrypt all transmitted data using TLS 1.2 for all communication with the Hypori server.

2.4 TOE Documentation

The TOE includes the following Hypori Client documentation.

- Hypori Virtual Mobility User Guide – Android, Client Release 4.2 – v1.1
- Hypori User Guide Common Criteria Configuration and Operation, Version 4.2.0

3. Security Problem Definition

This security target includes by reference the Security Problem Definition from the [PP_APP_v1.3]. The Security Problem Definition consists of threats that a conformant TOE is expected to address and assumptions about the operational environment of the TOE.

In general, the [PP_APP_v1.3] has presented a Security Problem Definition appropriate for application software that runs on mobile devices, as well as on desktop and server platforms. The Hypori Client is an Android application running on a mobile device. As such, the [PP_APP_v1.3] Security Problem Definition applies to the TOE.

4. Security Objectives

Like the Security Problem Definition, this security target includes by reference the Security Objectives from the [PP_APP_v1.3]. The [PP_APP_v1.3] security objectives for the operational environment are reproduced below, since these objectives characterize technical and procedural measures each consumer must implement in their operational environment.

In general, the [PP_APP_v1.3] has presented a Security Objectives statement appropriate for application software that runs on mobile devices, as well as on desktop and server platforms. Consequently, the [PP_APP_v1.3] security objectives are suitable for the Hypori Client TOE (Android).

4.1 Security Objectives for the Operational Environment

OE.PLATFORM	The TOE relies upon a trustworthy computing platform for its execution. This includes the underlying operating system and any discrete execution environment provided to the TOE.
OE.PROPER_USER	The user of the application software is not willfully negligent or hostile, and uses the software within compliance of the applied enterprise security policy.
OE.PROPER_ADMIN	The administrator of the application software is not careless, willfully negligent or hostile, and administers the software within compliance of the applied enterprise security policy.

5. IT Security Requirements

This section defines the Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) that serve to represent the security functional claims for the Target of Evaluation (TOE) and to scope the evaluation effort.

The security functional requirements have all been drawn from: *Protection Profile for Application Software*, Version 1.3, 1 March 2019 [PP_APP_v1.3]. As a result, refinements and operations already performed in that PP are not identified (e.g., highlighted) here, rather the requirements have been copied from that PP and any residual operations have been completed herein. Of particular note, [PP_APP_v1.3] made a number of refinements and completed some of the SFR operations defined in the CC. [PP_APP_v1.3] should be consulted to identify those changes if necessary.

The security assurance requirements are the set of SARs specified in [PP_APP_v1.3].

5.1 Extended Requirements

All of the extended requirements in this ST have been drawn from the [PP_APP_v1.3]. The [PP_APP_v1.3] defines the following extended SFRs. Since these SFRs are not redefined in this ST, readers should consult [PP_APP_v1.3] for more information in regard to these CC extensions.

- FCS_CKM_EXT.1 Cryptographic Key Generation Services
- FCS_RBG_EXT.1 Random Bit Generation Services
- FCS_STO_EXT.1 Storage of Credentials
- FDP_DAR_EXT.1 Encryption Of Sensitive Application Data
- FDP_NET_EXT.1 Network Communications
- FDP_DEC_EXT.1 Access to Platform Resources
- FIA_X509_EXT.1 X.509 Certificate Validation
- FIA_X509_EXT.2 X.509 Certificate Authentication
- FMT_MEC_EXT.1 Supported Configuration Mechanism
- FMT_CFG_EXT.1 Secure by Default Configuration
- FPR_ANO_EXT.1 User Consent for Transmission of Personally Identifiable Information
- FPT_AEX_EXT.1 Anti-Exploitation Capabilities
- FPT_API_EXT.1 Use of Supported Services and APIs
- FPT_IDV_EXT.1 Software Identification and Versions
- FPT_LIB_EXT.1 Use of Third Party Libraries
- FPT_TUD_EXT.1 Integrity for Installation and Update
- FPT_TUD_EXT.2 Integrity for Installation and Update
- FPT_IDV_EXT.1 Software Identification and Versions
- FTP_DIT_EXT.1 Protection of Data in Transit

5.2 TOE Security Functional Requirements

The following table identifies the SFRs that are satisfied by the Hypori Client TOE.

Table 1 TOE Security Functional Components

Requirement Class	Requirement Component
	FCS_CKM_EXT.1 Cryptographic Key Generation Services

Requirement Class	Requirement Component
FCS: Cryptographic support	FCS_RBG_EXT.1 Random Bit Generation Services
	FCS_STO_EXT.1 Storage of Credentials
FDP: User data protection	FDP_DAR_EXT.1 Encryption of Sensitive Application Data
	FDP_DEC_EXT.1 Access to Platform Resources
	FDP_NET_EXT.1 Network Communications
FIA: Identification and authentication	FIA_X509_EXT.1 X.509 Certificate Validation
	FIA_X509_EXT.2 X.509 Certificate Authentication
FMT: Security management	FMT_CFG_EXT.1 Secure by Default Configuration
	FMT_MEC_EXT.1 Supported Configuration Mechanism
	FMT_SMF.1 Specification of Management Functions
FPR: Privacy	FPR_ANO_EXT.1 User Consent for Transmission of Personally Identifiable Information
FPT: Protection of the TSF	FPT_AEX_EXT.1 Anti-Exploitation Capabilities
	FPT_API_EXT.1 Use of Supported Services and APIs
	FPT_IDV_EXT.1 Software Identification and Versions
	FPT_LIB_EXT.1 Use of Third Party Libraries
	FPT_TUD_EXT.1 Integrity for Installation and Update
	FPT_TUD_EXT.2 Integrity for Installation and Update
FTP: Trusted path/channels	FTP_DIT_EXT.1 Protection of Data in Transit

5.2.1 Cryptographic Support (FCS)

5.2.1.1 Cryptographic Key Generation Services (FCS_CKM_EXT.1)

FCS_CKM_EXT.1.1 The application shall [***generate no asymmetric cryptographic keys].***

5.2.1.2 Random Bit Generation Services (FCS_RBG_EXT.1)

FCS_RBG_EXT.1.1 The application shall [***use no DRBG functionality***] for its cryptographic operations.

5.2.1.3 Storage of Credentials (FCS_STO_EXT.1)

FCS_STO_EXT.1.1 The application shall **[*invoke the functionality provided by the platform to securely store [user TLS client key and server account password]***] to non-volatile memory.

5.2.2 User Data Protection (FDP)

5.2.2.1 Encryption of Sensitive Application Data (FDP_DAR_EXT.1)

FDP_DAR_EXT.1.1 The application shall [***protect sensitive data in accordance with FCS_STO_EXT.1***,] in nonvolatile memory.

5.2.2.2 Access to Platform Resources (FDP_DEC_EXT.1)

FDP_DEC_EXT.1.1 The application shall restrict its access to [

- ***network connectivity,***
- ***camera,***
- ***microphone,***
- ***location services,***
- ***[Wi-Fi,***
- ***Phone]***

] .

FDP_DEC_EXT.1.2 The application shall restrict its access to [

- ***no sensitive information repositories***

] .

5.2.2.3 Network Communications (FDP_NET_EXT.1)

FDP_NET_EXT.1.1 The application shall restrict network communication to [

- ***user-initiated communication for [connecting to the Hypori server],***
- ***respond to [push notifications from Google's FCM platform by polling the Hypori server for notifications],***
- ***[polling the Hypori server for notifications],***

] .

5.2.3 Security Management (FMT)

5.2.3.1 Secure by Default Configuration (FMT_CFG_EXT.1)

FMT_CFG_EXT.1.1 The application shall provide only enough functionality to set new credentials when configured with default credentials or no credentials.

FMT_CFG_EXT.1.2 The application shall be configured by default with file permissions which protect the application's binaries and data files from modification by normal unprivileged users.

5.2.3.2 Supported Configuration Mechanism (FMT_MEC_EXT.1)

FMT_MEC_EXT.1.1[1] The application shall

- [***invoke the mechanisms recommended by the platform vendor for storing and setting configuration options***].

5.2.3.3 Specification of Management Functions (FMT_SMF.1)

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions [***[***

- ***setting configuration options***
- ***applying configuration policies from the Hypori server]***

].

5.2.4 Privacy

5.2.4.1 User Consent for Transmission of Personally Identifiable Information (FPR_ANO_EXT.1)

FPR_ANO_EXT.1.1 The application shall [***not transmit PII over a network***].

5.2.5 Protection of the TSF (FPT)

5.2.5.1 Use of Supported Services and APIs (FPT_API_EXT.1)

FPT_API_EXT.1.1 The application shall use only documented platform APIs.

[1] This SFR was modified per NIAP TD0437.

5.2.5.2 Anti-Exploitation Capabilities (FPT_AEX_EXT.1)

FPT_AEX_EXT.1.1 The application shall not request to map memory at an explicit address except for [**no exceptions**].

FPT_AEX_EXT.1.2 The application shall **[*not allocate any memory region with both write and execute permissions*].**

FPT_AEX_EXT.1.3 The application shall be compatible with security features provided by the platform vendor.

FPT_AEX_EXT.1.4 The application shall not write user-modifiable files to directories that contain executable files unless explicitly directed by the user to do so.

FPT_AEX_EXT.1.5 The application shall be built with stack-based buffer overflow protection enabled.

5.2.5.3 Integrity for Installation and Update (FPT_TUD_EXT.1)

FPT_TUD_EXT.1.1 The application shall [***leverage the platform***] to check for updates and patches to the application software.

FPT_TUD_EXT.1.2 The application shall [***provide the ability***] to query the current version of the application software.

FPT_TUD_EXT.1.3 The application shall not download, modify, replace or update its own binary code.

FPT_TUD_EXT.1.4[2] Application updates shall be digitally signed such that the application platform can cryptographically verify them prior to installation.

FPT_TUD_EXT.1.5 The application is distributed [***as an additional software package to the platform OS***].

5.2.5.4 Integrity for Installation and Update (FPT_TUD_EXT.2)

FPT_TUD_EXT.2.1 The application shall be distributed using the format of the platform-supported package manager.

FPT_TUD_EXT.2.2 The application shall be packaged such that its removal results in the deletion of all traces of the application, with the exception of configuration settings, output files, and audit/log events.

FPT_TUD_EXT.2.3[3] The application installation package shall be digitally signed such that its platform can cryptographically verify them prior to installation.

5.2.5.5 Use of Third Party Libraries (FPT_LIB_EXT.1)

FPT_LIB_EXT.1.1 The application shall be packaged with only [

- **Opus Audio Codec v1.1**
- **Protobuf v2.5.0**
- **Zxing core 3.2.1**
- **Yubico**
- **Spongycastle**

].

[2] Modified per TD0561

[3] Modified per TD0561

5.2.5.6 Software Identification and Versions (FPT_IDV_EXT.1)

FPT_IDV_EXT.1.1 The application shall be versioned with [***[Android application version identifier, internal build information]***].

5.2.6 Trusted path/channels (FTP)

5.2.6.1 Protection of Data in Transit (FTP_DIT_EXT.1)

FTP_DIT_EXT.1.1[4] The application shall [

- i***nvoke platform-provided functionality to encrypt all transmitted data with [TLS]***] between itself and another trusted IT product.

5.2.7 Identification and authentication (FIA)

5.2.7.1 X.509 Certificate Validation (FIA_X509_EXT.1)

FIA_X509_EXT.1.1[5] The application shall [***invoked platform-provided functionality***] to validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certificate path validation.
- The certificate path must terminate with a trusted CA certificate.
- The application shall validate a certificate path by ensuring the presence of the basicConstraints extension, that the CA flag is set to TRUE for all CA certificates. and that any path constraints are met
- The application shall validate that any CA certificate includes caSigning purpose in the key usage field
- The application shall validate the revocation status of the certificate using [***OCSP as specified in RFC 6960, a Certificate Revocation List (CRL) as specified in RFC 5759***].
- The application shall validate the extendedKeyUsage (EKU) field according to the following rules:
  - Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.[6]
  - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the EKU field.
  - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the EKU field.
  - S/MIME certificates presented for email encryption and signature shall have the Email Protection purpose (id-kp 4 with OID 1.3.6.1.5.5.7.3.4) in the EKU field.[7]
  - OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the EKU field.

[4] The SFR was modified per TD0444.

[5] Modified per TD0521.

[6] The Hypori Client does not check extended key usage for Code Signing. The Hypori Client relies on the platform update mechanism. While Hypori signs each installation package with a Code Signing certificate, the platform verifies the certificate and package.

[7] The Hypori Client does not check extended key usage for Email Protection, since the Hypori Client does not perform email encryption or email signature verification.

- Server certificates presented for EST shall have the CMC Registration Authority (RA) purpose (id-kp-cmcRA with OID 1.3.6.1.5.5.7.3.28) in the EKU field.[8]

FIA_X509_EXT.1.2 The application shall only treat a certificate as a CA certificate if the basicConstraints extension is present and the CA flag is set to TRUE.

5.2.7.2 X.509 Certificate Authentication (FIA_X509_EXT.2)

FIA_X509_EXT.2.1[9] The application shall use X.509v3 certificates as defined by RFC 5280 to support authentication for [***TLS***].

FIA_X509_EXT.2.2 When the application cannot establish a connection to determine the validity of a certificate, the application shall [***not accept the certificate***].

5.3 TOE Security Assurance Requirements

The security assurance requirements in Table 2 are included in this ST by reference from the [PP_APP_v1.3].

Table 2 Assurance Components

Requirement Class	Requirement Component
ADV: Development	ADV_FSP.1 Basic functional specification
AGD: Guidance documents	AGD_OPE.1: Operational user guidance
	AGD_PRE.1: Preparative procedures
ALC: Life-cycle support	ALC_CMC.1 Labelling of the TOE
	ALC_CMS.1 TOE CM coverage
	ALC_TSU_EXT.1 Timely Security Updates
ATE: Tests	ATE_IND.1 Independent testing - conformance
AVA: Vulnerability assessment	AVA_VAN.1 Vulnerability survey

These assurance requirements imply the following requirements from CC class ASE: Security Target Evaluation.

- ASE_CCL.1 Conformance claims
- ASE_ECD.1 Extended components definition
- ASE_INT.1 ST introduction
- ASE_OBJ.1 Security objectives for the operational environment
- ASE_REQ.1 Stated security requirements
- ASE_TSS.1 TOE summary specification

Consequently, the assurance activities specified in [PP_APP_v1.3] apply to the TOE evaluation.

[8] The Hypori Client does not check extended key usage for CMC Registration Authority, since the Hypori Client does not perform Enrollment over Secure Transport.

[9] The SFR was modified per TD0444.

6. TOE Summary Specification

This chapter describes the security functions:

- Cryptographic support
- User data protection
- Certificate validation
- Security management
- Privacy
- Protection of the TSF
- Trusted path/channels

6.1 Cryptographic support

The Hypori Client makes use of the platform for cryptographic services. The Hypori Client uses platform TLS services for secure communication with the Hypori server, including mutual authentication. The client uses TLS client certificates and keys along with a CA certificate for the server. The user stores these certificates in the platform's key store during installation. The user need not install a CA certificate when the CA is a platform trusted CA.

The TOE relies on the platform to provide all of its cryptographic functionality. The following Android evaluations are conformant to the Common Criteria for IT Security Evaluation (ISO Standard 15408) and are listed at the National Information Assurance Partnership (NIAP) Product Compliant List.

Android 8.1 – VID11001 (https://www.niap-ccevs.org/MMO/Product/st_vid11001-st.pdf)

- Supported ciphersuites
    - TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
    - TLS_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246
    - TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288
    - TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
    - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
    - TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
    - TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

Android 9– VID10979 (https://www.niap-ccevs.org/MMO/ProductAM/st_vid10979-st-2.pdf)

- Supported ciphersuites
    - TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
    - TLS_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246
    - TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288
    - TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
    - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
    - TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
    - TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

Android 10 – VID11042 (https://www.niap-ccevs.org/MMO/ProductAM/st_vid11042-st-2.pdf)

- Supported ciphersuites

- TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

Each of these OS evaluations listed on the NIAP Product Compliant List is capable of presenting the supported Elliptic Curves Extension in the Client using the secp384r1 NIST curve.

6.1.1 FCS_CKM_EXT.1

The Hypori Client does not generate cryptographic keys. As part of installation, a user adds a Hypori server TLS client certificate and key to the platform's key store. The Hypori Client relies on the platform for TLS support. The platform generates all ephemeral TLS keys without direct Hypori Client action.

6.1.2 FCS_RBG_EXT.1

The Hypori Client relies on the platform for cryptographic services. Consequently, the Hypori Client itself uses no DRBG functions.

6.1.3 FCS_STO_EXT.1

Table 3 lists each Hypori Client persistent credential along with how the client uses and stores each credential.

Table 3: Persistent Credential Use and Storage

Credential	Purpose	Storage
User TLS client key	Authenticates Hypori Client when establishing TLS connection to Hypori server	Android Keystore System
Server account password	Authenticates user to Hypori server	Android Keystore System

6.2 User data protection

The Hypori Client uses the platform's permission mechanisms to inform the user of hardware and software resources the client accesses. The client presents the required permissions to the user for approval during installation. A user initiates network connections to the Hypori server. In general, sensitive data resides on the Hypori server and is not stored on the Hypori Client. Sensitive data on the Hypori Client is limited to credentials, which the client stores as described in section 6.1. The client does not maintain Personally Identifiable Information (PII).

6.2.1 FDP_DAR_EXT.1

Hypori Client sensitive data consist of user TLS client key and server account password credentials. FCS_STO_EXT.1 Storage of Secrets specifies the platform's Android Keystore System for protecting keys and credentials (see https://developer.android.com/training/articles/keystore for details on the Android Keystore System). In accordance with FCS_STO_EXT.1, the Hypori Client stores these credentials in the platform's Android Keystore System as described in section 6.1.3. Administrators can decide to provision credentials using the Android Keystore System (either the system-wide Android KeyChain or the application-only Android Keystore Provider).

The Hypori Client stores application account options using Android's SharedPreferences. The SharedPreferences files are accessed using the MODE_PRIVATE flag, even though the application account options do not contain sensitive data.

6.2.2 FDP_NET_EXT.1

The Hypori Client relies on user-initiated network communication to connect to the Hypori Virtual Device. The Hypori Client uses remote-initiated network communication to check for notifications and display them to the user when the system is configured for push notifications. The Hypori Client uses application-initiated network communication to periodically check for notifications and display them to the user when the system is configured for notification polling.

6.2.3 FDP_DEC_EXT.1

The installer presents to the user the permissions required by the Hypori Client. A user must accept the permissions to complete installation. Table shows the permissions required by the Hypori Client:

Permission	Description
INTERNET	Open network sockets.
USE_FINGERPRINT	Use fingerprint hardware.
WAKE_LOCK	Use PowerManager WakeLocks to maintain connection.
RECORD_AUDIO	Enable audio recording.
ACCESS_FINE_LOCATION	Access precise location.
ACCESS_LOCATION_EXTRA_COMMANDS	Access extra location provider commands.
READ_SYNC_SETTINGS	Read the sync settings.
WRITE_SYNC_SETTINGS	Write the sync settings.
ACCESS_NETWORK_STATE	Access information about networks.
CHANGE_NETWORK_STATE	Change network connectivity state.
ACCESS_WIFI_STATE	Access information about Wi-Fi networks.
MODIFY_AUDIO_SETTINGS	Modify global audio settings.
READ_PHONE_STATE	Read only access to phone state, including the phone number of the device, current cellular network information, the status of any ongoing calls.
CAMERA	Access the mobile device's camera.
INSTALL_SHORTCUT	Install a shortcut in the Launcher.
UNINSTALL_SHORTCUT	Uninstall a shortcut in the Launcher.
BLUETOOTH	Connect to paired Bluetooth devices.
BLUETOOTH_ADMIN	Discover and pair Bluetooth devices.
RECEIVE_BOOT_COMPLETED	Receive notification after the system finishes booting.
CALL_PHONE	Initiate a phone call bypassing the Dialer interface to confirm the call.
VIBRATE	Access to the mobile device's vibrator.
FLASHLIGHT	Access to the mobile device's flashlight.
GET_ACCOUNTS (Deprecated)	Access to the list of accounts in the Accounts Service.
MANAGE_ACCOUNTS (Deprecated)	Allow app to add and remove accounts.
AUTHENTICATE_ACCOUNTS (Deprecated)	Use the account authenticator capabilities of the AccountManager.

Permission	Description
GET_TASKS (Deprecated)	Allow the app to retrieve information about currently and recently running tasks

Updates to the Hypori Client may automatically add additional capabilities within each group. A user must accept new permissions to complete any update that includes permissions not in the list above.

A user initiates a network connection to the Hypori server by starting the Hypori Client and entering account information. After the Hypori Client connects to the Hypori server, the applications the user accesses run on the Hypori Device in the Hypori server, not on the mobile device. The Hypori Client does not listen on any ports for inbound connection requests. The Hypori Client interacts only with the Hypori server. When a Hypori Device application needs information from a server (such as a map server), the Hypori Device – not the Hypori Client – communicates with the server (which may be an internal, enterprise server).

The Hypori Client does not maintain PII. Hence, it does not transmit PII over any network.[10] As per the claimed PP, the TOE is not considered to maintain PII unless it provides an interface intended specifically to collect such data; general-purpose communications interfaces may contain PII supplied by the user that the TSF is not expected to treat in a special manner.

The TOE does not contain sensitive information repositories as defined in the [PP_APP_v1.3].

6.3 Identification and authentication

The Android platform follows RFC 5280 *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile* for certification path validation. The Hypori Client uses the Android certification validation services to authenticate the X.509 certificate the Hypori server presents as part of the establishing a TLS connection.

6.3.1 FIA_X509_EXT.1

The Android platform performs certificate path validation in accordance with RFC 5280 as part of the TLS service. It recursively builds certificate chains until a valid chain is found or all possible paths are exhausted. The chain begins at the leaf certificate and ends in the final trusted root certificate[11].

The Hypori Client relies on the platform for TLS services and package updates. Hence, the platform checks extended key usage for Server Authentication, Client Authentication, and Code Signing purposes. The Hypori Client performs validation of the revocation status using the Certificate Revocation List (CRL) as specified in RFC 5759 provided by the CRL distribution point specified in the certificates.

The Android platform validates the revocation status of the certificate using the Online Certificate Status Protocol (OCSP) as specified in RFC 6960. When the platform cannot establish a connection to an OCSP server providing the status to determine certificate validity, the platform will reject the connection.

The Hypori Client does not perform email encryption, email signature verification, and Enrollment over Secure Transport. Consequently, no check is made for extended key usage Email Protection and CMC Registration Authority purposed.

[10] The Hypori Client accesses user credentials. In particular, the Hypori Client transmits a user's account name and TLS client certificate when connecting to the Hypori Server. However, **Error! Reference source not found.** distinguishes credentials from PII.

[11] The platform certificate path algorithm is described by its Android platform source code, available at: https://cs.android.com/android/platform/superproject/+/master:external/conscrypt/common/src/main/java/org/conscrypt/TrustManagerImpl.java. See the checkTrusted() method at line 267 and line 275 for older Android versions for the algorithm.

6.3.2 FIA_X509_EXT.2

The Hypori Client presents the TLS client certificate and key to the Hypori server to authenticate a TLS connection. During account setup, the user identifies which certificate to present for each account. The user selects a certificate from the certificate store. The user can change the selection from Client Certificate under Connection on the Settings page. The TLS client certificate is an X.509 certificate. The user stores a CA certificate for the server certificates in the platform's key store during installation. (The user need not install a CA certificate when the CA is a platform trusted CA.)

The user stores a CA certificate for the server certificates in the platform's key store during installation. (The user need not install a CA certificate when the CA is a platform trusted CA.) On Android devices, the Hypori Client uses Android platform certificate path validation services with the CA certificate to validate the certificate presented by the Hypori server. The Hypori Client extracts the CRL distribution point from the certificate, contacts the server to download the CRL, and validates that the certificate is not revoked. The TOE also supports revocation checking of the certificate presented using OCSP (as specified in RFC 6960). If the CRL/OCSP server fails to respond or there is an error, the Hypori Client will not accept the certificate (invalid) and not establish the connection.

6.4 Security management

Security management consists of setting Hypori Client configuration options. The client uses Android mechanisms for storing the configuration settings.

6.4.1 FMT_CFG_EXT.1

Hypori Client credentials consist of user TLS client key and server account password. The Hypori Client installer does not include a default client key or server account password. A user installs a TLS client certificate and private key from a certificate file using the platform's certificate services. A user's IT group provides the user with a server account password. The user is not able to access any TOE functionality prior to installing the TLS client certificate and private key, and entering the server account password.

The default file permissions protect the application's binaries and data files from modification by normal unprivileged users.

The "default permissions" are those provided by Android. In particular, the base apk permission is 644, which breaks down to the following:

- it is able to be read + written to by owner
- it is able to be read by group,
- it is able to be read by others.

The shared library files permissions are 755 as required by Android, which break down to:

- the owner which is root will read, write and execute in the directory,
- the group will only read and execute in the directory,
- others will only read and execute in the directory.

All of these files are owned by system/system.

Preferences/options files are stored in the shared_prefs directory with permissions 660, which breaks down to the following:

- the owner can read and write but not execute,
- the group is able to read and write but not execute,
- others cannot read, write or execute,

The preferences/options files are owned by the uid/gid associated with the application (varies per installation). These are all defined by Android.

6.4.2 FMT_MEC_EXT.1

The Hypori Client invokes the recommended Android mechanisms for storing account settings files. On Android devices, the client uses SharedPreferences and extends PreferenceActivity.

6.4.3 FMT_SMF.1

For each account, the Hypori Client provides the capability to set the Hypori server IP address, Hypori server port, account name, and TLS client certificate (key). The Hypori Client can enable the Remember Password setting for each account. The operational guidance recommends that the user disable this functionality. The Hypori Client Remember Password setting can also be disabled by policies received from the Hypori server.

The Hypori Client does not require any configuration to use ports and protocol. The Hypori Client does not listen on any ports for inbound connection requests. The Hypori Client interacts only with the Hypori server. When a Hypori Device application needs information from a server (such as a map server), the Hypori Device – not the Hypori Client – communicates with the server (which may be an internal, enterprise server).

6.5 Privacy

The Hypori Client does not transmit PII over a network.

6.5.1 FPR_ANO_EXT.1

The Hypori Client does not transmit PII over a network.

6.6 Protection of the TSF

The Hypori Client uses security features and APIs that the platform provides. This includes address space layout randomization, data execution protection, Security Enhancements for Android, and stack-based buffer overflow protection. The client leverages Android package management for secure installation and updates. The Hypori Client package includes only those third-party libraries necessary for its intended operation.

6.6.1 FPT_AEX_EXT.1

Hypori enables address space layout randomization (ASLR) in the Android Hypori Client using `-fpic` when building the application with Android Native Development Kit (NDK r15c) using gcc. The Hypori Client is a Java application that includes Java Native Interface (JNI) libraries. Hypori enables stack-based buffer overflow protection using `-fstack-protector-strong`. The Hypori Client does not invoke `mmap` or `mprotect` from the Android NDK.

6.6.2 FPT_API_EXT.1

The Hypori Client uses the Android APIs listed in Section 9 Appendix: Android APIs and Section 10. Appendix: Java Library APIs.

6.6.3 FPT_IDV_EXT.1

The TOE is the Hypori Client (Android) v 4.2.0. The TOE is identified and versioned by the Android application version identifiers as well as the internal Hypori build information.

The values listed below are provided for illustration purposes:

- Version name: 4.2.0 (on Play Store and in App)
- Version code: 401140004 (hidden inside app, not exposed to user)
- Internal build code (created by build system, shown in App)

Below is an example of version string as shown in Hypori UI:

- 4.1.14 (407000019-a53904e)

Versionname = (internal build code)

The versionname string is also provided in Google Play Store and the Android App Info for installed apps.

The Hypori (Android) Client 4.2.0 version is interpreted as a major.minor.maintenance-release format.

6.6.4 FPT_LIB_EXT.1

The Hypori Client package includes only the third-party libraries listed in the security functional requirements.

6.6.5 FPT_TUD_EXT.1, FPT_TUD_EXT.2

Hypori distributes the Hypori Client as a .APK file for Android devices. A user may obtain the installation package through Google Play or the enterprise IT group of the user. A user obtains Hypori Client updates using the platform's update mechanism or from the user's IT group. Hypori digitally signs the installation package as well as updates and includes the corresponding public key certificate in the package. Android will install an update only when the certificate in the update matches the certificate in the installed client. The client is signed with a unique certificate. It can be delivered via the Google Play store, MDM, or other enterprise app stores.

A user can see the current version of the Hypori Client by checking the footer information on all screens.

6.7 Trusted path/channels

The Hypori Client uses TLS 1.2 for all communication with Hypori server.

6.7.1 FTP_DIT_EXT.1

The Hypori server is the only trusted IT product the Hypori Client communicates with. For all communication with the Hypori server, the Hypori Client connects to the server using TLS 1.2 provided by the platform.

The TOE uses the platform android.net.SSLCertificateSocketFactory and javax.net.ssl.SSLSocket calls to invoke the functionality.

6.8 Timely Security Updates

6.8.1 ALC_TSU_EXT.1

Hypori provides customers with timely updates. A customer chooses their preferred communication. The Hypori Support Department will notify customers of updates using each customer's preferred communication mechanism. Application changes may be pushed to end users via the Google Play Store like any other application or via an enterprise application store internal to a customer. Typical delivery times for security updates are 5 to 10 business days.

Hypori maintains a Security Portal online. Every customer is registered with the Support Portal. Hypori notifies each customer of a new security report on the Support portal using the customers preferred communication mechanism. Hypori secures the Support Portal via SSL and user authentication. Each customer contact must log in with their specific credentials in order to see the security reports.

7. Protection Profile Claims

This ST conforms to the *Protection Profile for Application Software,* Version 1.3, 2019-03-01 [PP_APP_v1.3].

As explained in Section 3, Security Problem Definition, the Security Problem Definition of the [PP_APP_v1.3] has been included by reference into this ST.

As explained in Section 4, Security Objectives, the Security Objectives of the [PP_APP_v1.3] have been included by reference into this ST.

The following table identifies all the security functional requirements in this ST. Each SFR is reproduced from the [PP_APP_v1.3] and operations completed as appropriate.

Table 4 SFR Protection Profile Sources

Requirement Class	Requirement Component	Source
FCS: Cryptographic support	FCS_CKM_EXT.1 Cryptographic Key Generation Services	[PP_APP_v1.3]
	FCS_RBG_EXT.1 Random Bit Generation Services	[PP_APP_v1.3]
	FCS_STO_EXT.1 Storage of Credentials	[PP_APP_v1.3]
FDP: User data protection	FDP_DAR_EXT.1 Encryption of Sensitive Application Data	[PP_APP_v1.3]
	FDP_DEC_EXT.1 Access to Platform Resources	[PP_APP_v1.3]
	FDP_NET_EXT.1 Network Communications	[PP_APP_v1.3]
FIA: Identification and authentication	FIA_X509_EXT.1 X.509 Certificate Validation	[PP_APP_v1.3]
	FIA_X509_EXT.2 X.509 Certificate Authentication	[PP_APP_v1.3]
FMT: Security management	FMT_CFG_EXT.1 Secure by Default Configuration	[PP_APP_v1.3]
	FMT_MEC_EXT.1 Supported Configuration Mechanism	[PP_APP_v1.3]
	FMT_SMF.1 Specification of Management Functions	[PP_APP_v1.3]
FPR: Privacy	FPR_ANO_EXT.1 User Consent for Transmission of Personally Identifiable Information	[PP_APP_v1.3]
FPT: Protection of the TSF	FPT_AEX_EXT.1 AntiExploitation Capabilities	[PP_APP_v1.3]
	FPT_API_EXT.1.1 Use of Supported Services and APIs	[PP_APP_v1.3]
	FPT_IDV_EXT.1 Software Identification and Versions	[PP_APP_v1.3]
	FPT_LIB_EXT.1 Use of Third Party Libraries	[PP_APP_v1.3]
	FPT_TUD_EXT.1 Integrity for Installation and Update	[PP_APP_v1.3]
	FPT_TUD_EXT.2 Integrity for Installation and Update	[PP_APP_v1.3]
FTP: Trusted path/channels	FTP_DIT_EXT.1 Protection of Data in Transit	[PP_APP_v1.3]

8. Rationale

This security target includes by reference the [PP_APP_v1.3] Security Problem Definition, Security Objectives, and Security Assurance Requirements. The security target makes no additions to the [PP_APP_v1.3] assumptions. [PP_APP_v1.3] security functional requirements have been reproduced with the [PP_APP_v1.3] operations completed. Operations on the security requirements follow [PP_APP_v1.3] application notes and assurance activities. Consequently, [PP_APP_v1.3] rationale applies but is incomplete. The TOE Summary Specification rationale below serves to complete the rationale required for the security target.

8.1 Dependency Rationale

The Protection Profile for Application Software [PP_APP_v1.3] contains all the requirements claimed in this Security Target. As such, the dependencies are not applicable since the PP has been approved.

8.2 TOE Summary Specification Rationale

Each subsection in Section 6, the TOE Summary Specification, describes a security function of the TOE. Each description is followed with rationale that indicates which requirements are satisfied by aspects of the corresponding security function. The security functions work together to satisfy all of the security functional requirements. Furthermore, all of the security functions are necessary in order for the TSF to provide the required security functionality.

This section in conjunction with Section 6 TOE Summary Specification provides evidence that the security functions are suitable to meet the TOE security requirements. The collection of security functions works together to provide all of the security requirements. The security functions described in the TOE summary specification are all necessary for the required security functionality in the TSF. Table 5 demonstrates the relationship between security requirements and security functions.

Table 5 Security Functions vs. Requirements Mapping

	Cryptographic support	User data protection	Identification and authentication	Security management	Privacy	Protection of the TSF	Trusted path/channels
FCS_CKM_EXT.1	X						
FCS_RBG_EXT.1	X						
FCS_STO_EXT.1	X						
FDP_DAR_EXT.1		X					
FDP_NET_EXT.1		X					
FDP_DEC_EXT.1		X					
FIA_X509_EXT.1			X				
FIA_X509_EXT.2			X				
FMT_CFG_EXT.1				X			
FMT_MEC_EXT.1				X			
FMT_SMF.1				X			
FPR_ANO_EXT.1					X		
FPT_AEX_EXT.1						X	
FPT_API_EXT.1						X	
FPT_IDV_EXT.1						X	
FPT_LIB_EXT.1						X	
FPT_TUD_EXT.1						X	

	Cryptographic support	User data protection	Identification and authentication	Security management	Privacy	Protection of the TSF	Trusted path/channels
FPT_TUD_EXT.1						X	
FTP_DIT_EXT.1							X

9. Appendix: Android APIs

The Hypori Client uses the following Android APIs:

1. android.accounts.AbstractAccountAuthenticator
2. android.accounts.Account
3. android.accounts.AccountAuthenticatorResponse
4. android.accounts.AccountManager
5. android.accounts.AccountManagerCallback
6. android.accounts.AccountManagerFuture
7. android.accounts.AccountsException
8. android.accounts.NetworkErrorException
9. android.accounts.OperationCanceledException
10. android.animation.Animator
11. android.animation.AnimatorListenerAdapter
12. android.animation.ValueAnimator
13. android.annotation.SuppressLint
14. android.annotation.TargetApi
15. android.app.ActionBar
16. android.app.Activity
17. android.app.ActivityManager
18. android.app.ActivityManager.RunningTaskInfo
19. android.app.AlertDialog
20. android.app.Application
21. android.app.Application.ActivityLifecycleCallbacks
22. android.app.Dialog
23. android.app.DialogFragment
24. android.app.IntentService
25. android.app.KeyguardManager
26. android.app.ListActivity
27. android.app.Notification
28. android.app.NotificationChannel
29. android.app.NotificationChannelGroup
30. android.app.NotificationManager
31. android.app.PendingIntent
32. android.app.ProgressDialog
33. android.app.SearchManager
34. android.app.SearchableInfo
35. android.app.Service

36. android.app.UiModeManager
37. android.app.admin.DeviceAdminReceiver
38. android.app.admin.DevicePolicyManager
39. android.bluetooth.BluetoothAdapter
40. android.bluetooth.BluetoothClass
41. android.bluetooth.BluetoothDevice
42. android.bluetooth.BluetoothGatt
43. android.bluetooth.BluetoothGattCallback
44. android.bluetooth.BluetoothGattCharacteristic
45. android.bluetooth.BluetoothGattDescriptor
46. android.bluetooth.BluetoothGattService
47. android.bluetooth.BluetoothProfile
48. android.bluetooth.BluetoothServerSocket
49. android.bluetooth.BluetoothSocket
50. android.content.AbstractThreadedSyncAdapter
51. android.content.ActivityNotFoundException
52. android.content.BroadcastReceiver
53. android.content.ComponentName
54. android.content.ContentProvider
55. android.content.ContentProviderClient
56. android.content.ContentResolver
57. android.content.ContentUris
58. android.content.ContentValues
59. android.content.Context
60. android.content.DialogInterface
61. android.content.DialogInterface.OnClickListener
62. android.content.Intent
63. android.content.Intent.ShortcutIconResource
64. android.content.IntentFilter
65. android.content.ServiceConnection
66. android.content.SharedPreferences
67. android.content.SharedPreferences.Editor
68. android.content.SharedPreferences.OnSharedPreferenceChangeListener
69. android.content.SyncResult
70. android.content.UriMatcher
71. android.content.pm.ActivityInfo
72. android.content.pm.ApplicationInfo

73. android.content.pm.PackageInfo
74. android.content.pm.PackageManager
75. android.content.pm.PackageManager.NameNotFoundException
76. android.content.pm.ResolveInfo
77. android.content.pm.Signature
78. android.content.res.AssetFileDescriptor
79. android.content.res.AssetManager
80. android.content.res.Configuration
81. android.content.res.Resources
82. android.content.res.TypedArray
83. android.database.ContentObserver
84. android.database.Cursor
85. android.database.DataSetObserver
86. android.database.MatrixCursor
87. android.database.sqlite.SQLiteDatabase
88. android.database.sqlite.SQLiteException
89. android.database.sqlite.SQLiteOpenHelper
90. android.graphics.Bitmap
91. android.graphics.Bitmap.CompressFormat
92. android.graphics.BitmapFactory
93. android.graphics.Canvas
94. android.graphics.Color
95. android.graphics.ImageFormat
96. android.graphics.Matrix
97. android.graphics.Paint
98. android.graphics.Path
99. android.graphics.PixelFormat
100. android.graphics.Point
101. android.graphics.PointF
102. android.graphics.PorterDuff
103. android.graphics.Rect
104. android.graphics.RectF
105. android.graphics.SurfaceTexture
106. android.graphics.SurfaceTexture.OnFrameAvailableListener
107. android.graphics.Typeface
108. android.graphics.YuvImage
109. android.graphics.drawable.BitmapDrawable

110.android.graphics.drawable.ColorDrawable
111.android.graphics.drawable.Drawable
112.android.graphics.drawable.GradientDrawable
113.android.graphics.drawable.LayerDrawable
114.android.graphics.drawable.ShapeDrawable
115.android.graphics.drawable.TransitionDrawable
116.android.graphics.drawable.shapes.RoundRectShape
117.android.hardware.Camera
118.android.hardware.Camera.Area
119.android.hardware.Camera.CameraInfo
120.android.hardware.Camera.Face
121.android.hardware.Camera.FaceDetectionListener
122.android.hardware.Camera.Parameters
123.android.hardware.Camera.PictureCallback
124.android.hardware.Camera.Size
125.android.hardware.Sensor
126.android.hardware.SensorEvent
127.android.hardware.SensorEventListener
128.android.hardware.SensorManager
129.android.location.GpsSatellite
130.android.location.GpsStatus
131.android.location.GpsStatus.Listener
132.android.location.GpsStatus.NmeaListener
133.android.location.Location
134.android.location.LocationListener
135.android.location.LocationManager
136.android.location.LocationProvider
137.android.media.AudioFormat
138.android.media.AudioManager
139.android.media.AudioRecord
140.android.media.AudioTrack
141.android.media.CamcorderProfile
142.android.media.CameraProfile
143.android.media.MediaActionSound
144.android.media.MediaCodec
145.android.media.MediaCodec.BufferInfo
146.android.media.MediaCodecInfo

147.android.media.MediaCodecList
148.android.media.MediaFormat
149.android.media.MediaMetadataRetriever
150.android.media.MediaPlayer
151.android.media.MediaRecorder
152.android.media.ThumbnailUtils
153.android.media.audiofx.AcousticEchoCanceler
154.android.media.audiofx.AutomaticGainControl
155.android.media.audiofx.NoiseSuppressor
156.android.net.ConnectivityManager
157.android.net.LinkAddress
158.android.net.LinkProperties
159.android.net.LocalSocket
160.android.net.LocalSocketAddress
161.android.net.Network
162.android.net.NetworkInfo
163.android.net.RouteInfo
164.android.net.SSLCertificateSocketFactory
165.android.net.Uri
166.android.net.Uri.Builder
167.android.net.wifi.WifiConfiguration
168.android.net.wifi.WifiInfo
169.android.net.wifi.WifiManager
170.android.net.wifi.WifiManager.WifiLock
171.android.opengl.EGL14
172.android.opengl.EGLConfig
173.android.opengl.EGLContext
174.android.opengl.EGLDisplay
175.android.opengl.EGLExt
176.android.opengl.EGLSurface
177.android.opengl.GLES11Ext
178.android.opengl.GLES20
179.android.opengl.GLSurfaceView
180.android.opengl.GLSurfaceView.Renderer
181.android.opengl.GLUtils
182.android.opengl.Matrix
183.android.os.AsyncTask

184.android.os.BatteryManager
185.android.os.Binder
186.android.os.Build
187.android.os.Bundle
188.android.os.Environment
189.android.os.Handler
190.android.os.HandlerThread
191.android.os.IBinder
192.android.os.Looper
193.android.os.Message
194.android.os.Parcel
195.android.os.ParcelFileDescriptor
196.android.os.ParcelFileDescriptor.AutoCloseOutputStream
197.android.os.Parcelable
198.android.os.PowerManager
199.android.os.Process
200.android.os.StatFs
201.android.os.SystemClock
202.android.os.UserHandle
203.android.os.UserManager
204.android.os.Vibrator
205.android.preference.CheckBoxPreference
206.android.preference.EditTextPreference
207.android.preference.Preference
208.android.preference.Preference.OnPreferenceChangeListener
209.android.preference.Preference.OnPreferenceClickListener
210.android.preference.PreferenceFragment
211.android.preference.PreferenceGroup
212.android.preference.PreferenceManager
213.android.provider.ContactsContract
214.android.provider.MediaStore
215.android.provider.MediaStore.Images
216.android.provider.MediaStore.Images.ImageColumns
217.android.provider.MediaStore.MediaColumns
218.android.provider.MediaStore.Video
219.android.provider.MediaStore.Video.VideoColumns
220.android.provider.Settings

221.android.provider.Settings.System
222.android.security.KeyChain
223.android.security.KeyChainAliasCallback
224.android.security.KeyChainException
225.android.service.notification.StatusBarNotification
226.android.system.ErrnoException
227.android.telephony.PhoneNumberUtils
228.android.telephony.PhoneStateListener
229.android.telephony.ServiceState
230.android.telephony.SignalStrength
231.android.telephony.TelephonyManager
232.android.text.Html
233.android.text.InputFilter
234.android.text.Spannable
235.android.text.SpannableString
236.android.text.Spanned
237.android.text.TextUtils
238.android.text.format.DateFormat
239.android.text.format.Time
240.android.text.method.MovementMethod
241.android.text.method.ScrollingMovementMethod
242.android.text.style.StyleSpan
243.android.util.AttributeSet
244.android.util.Base64
245.android.util.Base64OutputStream
246.android.util.DisplayMetrics
247.android.util.Log
248.android.util.Range
249.android.util.SparseArray
250.android.util.TypedValue
251.android.util.Xml
252.android.view.Display
253.android.view.GestureDetector
254.android.view.Gravity
255.android.view.InflateException
256.android.view.InputDevice
257.android.view.KeyEvent

258.android.view.LayoutInflater
259.android.view.Menu
260.android.view.MenuInflater
261.android.view.MenuItem
262.android.view.MenuItem.OnMenuItemClickListener
263.android.view.MotionEvent
264.android.view.OrientationEventListener
265.android.view.SoundEffectConstants
266.android.view.Surface
267.android.view.SurfaceHolder
268.android.view.SurfaceView
269.android.view.VelocityTracker
270.android.view.View
271.android.view.View.OnClickListener
272.android.view.View.OnTouchListener
273.android.view.ViewConfiguration
274.android.view.ViewGroup
275.android.view.ViewGroup.LayoutParams
276.android.view.ViewParent
277.android.view.ViewTreeObserver
278.android.view.Window
279.android.view.WindowManager
280.android.view.accessibility.AccessibilityEvent
281.android.view.animation.AccelerateInterpolator
282.android.view.animation.AlphaAnimation
283.android.view.animation.Animation
284.android.view.animation.Animation.AnimationListener
285.android.view.animation.AnimationUtils
286.android.view.animation.DecelerateInterpolator
287.android.view.inputmethod.EditorInfo
288.android.view.inputmethod.InputConnection
289.android.view.inputmethod.InputMethodManager
290.android.webkit.WebSettings
291.android.webkit.WebView
292.android.webkit.WebViewClient
293.android.widget.AbsListView
294.android.widget.Adapter

295.android.widget.AdapterView
296.android.widget.AdapterView.OnItemClickListener
297.android.widget.AdapterView.OnItemLongClickListener
298.android.widget.ArrayAdapter
299.android.widget.BaseAdapter
300.android.widget.Button
301.android.widget.CheckBox
302.android.widget.CompoundButton
303.android.widget.CompoundButton.OnCheckedChangeListener
304.android.widget.CursorAdapter
305.android.widget.EditText
306.android.widget.Filter
307.android.widget.Filterable
308.android.widget.FrameLayout
309.android.widget.GridView
310.android.widget.HorizontalScrollView
311.android.widget.ImageButton
312.android.widget.ImageView
313.android.widget.ImageView.ScaleType
314.android.widget.LinearLayout
315.android.widget.ListView
316.android.widget.PopupMenu
317.android.widget.PopupWindow
318.android.widget.ProgressBar
319.android.widget.RelativeLayout
320.android.widget.ScrollView
321.android.widget.SearchView
322.android.widget.SimpleAdapter
323.android.widget.Switch
324.android.widget.TabWidget
325.android.widget.TextView
326.android.widget.Toast
327.androidx.annotation.NonNull
328.androidx.annotation.Nullable
329.androidx.appcompat.app.AppCompatActivity
330.androidx.biometric.BiometricManager
331.androidx.biometric.BiometricPrompt

332.androidx.core.app.ActivityCompat
333.androidx.core.app.NotificationCompat
334.androidx.core.content.ContextCompat
335.androidx.core.view.MotionEventCompat
336.androidx.core.view.ViewConfigurationCompat
337.androidx.drawerlayout.widget.DrawerLayout
338.androidx.fragment.app.DialogFragment
339.androidx.fragment.app.Fragment
340.androidx.fragment.app.FragmentActivity
341.androidx.fragment.app.FragmentManager
342.androidx.fragment.app.FragmentStatePagerAdapter
343.androidx.fragment.app.FragmentTransaction
344.androidx.localbroadcastmanager.content.LocalBroadcastManager
345.androidx.multidex.MultiDex
346.androidx.multidex.MultiDexApplication
347.androidx.recyclerview.widget.LinearLayoutManager
348.androidx.recyclerview.widget.RecyclerView
349.androidx.viewpager.widget.ViewPager
350.com.google.android.gms.common.ConnectionResult
351.com.google.android.gms.common.GoogleApiAvailability
352.com.google.android.material.snackbar.Snackbar
353.com.google.firebase.FirebaseApp
354.com.google.firebase.FirebaseOptions
355.com.google.firebase.iid.FirebaseInstanceId
356.com.google.firebase.iid.FirebaseInstanceIdService
357.com.google.firebase.messaging.FirebaseMessaging
358.com.google.firebase.messaging.FirebaseMessagingService
359.com.google.firebase.messaging.RemoteMessage
360.org.xmlpull.v1.XmlPullParser
361.org.xmlpull.v1.XmlPullParserException
362.java.beans.PropertyChangeEvent
363.java.beans.PropertyChangeListener
364.java.io.BufferedInputStream
365.java.io.BufferedOutputStream
366.java.io.BufferedReader
367.java.io.BufferedWriter
368.java.io.ByteArrayInputStream

369.java.io.ByteArrayOutputStream
370.java.io.Closeable
371.java.io.DataInputStream
372.java.io.DataOutputStream
373.java.io.File
374.java.io.FileDescriptor
375.java.io.FileInputStream
376.java.io.FileNotFoundException
377.java.io.FileOutputStream
378.java.io.FileReader
379.java.io.FileWriter
380.java.io.FilenameFilter
381.java.io.IOException
382.java.io.InputStream
383.java.io.InputStreamReader
384.java.io.ObjectInputStream
385.java.io.ObjectOutputStream
386.java.io.OutputStream
387.java.io.OutputStreamWriter
388.java.io.PrintStream
389.java.io.PrintWriter
390.java.io.RandomAccessFile
391.java.io.Serializable
392.java.io.StringWriter
393.java.io.UnsupportedEncodingException
394.java.io.Writer
395.java.lang.Thread.UncaughtExceptionHandler
396.java.lang.annotation.ElementType
397.java.lang.annotation.Retention
398.java.lang.annotation.RetentionPolicy
399.java.lang.annotation.Target
400.java.lang.ref.WeakReference
401.java.lang.reflect.Array
402.java.lang.reflect.Constructor
403.java.lang.reflect.Field
404.java.lang.reflect.InvocationTargetException
405.java.lang.reflect.Method

406.java.math.BigInteger
407.java.net.ConnectException
408.java.net.HttpURLConnection
409.java.net.InetAddress
410.java.net.MalformedURLException
411.java.net.Socket
412.java.net.SocketException
413.java.net.URL
414.java.net.URLEncoder
415.java.net.UnknownHostException
416.java.net.UnknownServiceException
417.java.nio.BufferOverflowException
418.java.nio.BufferUnderflowException
419.java.nio.ByteBuffer
420.java.nio.ByteOrder
421.java.nio.CharBuffer
422.java.nio.DoubleBuffer
423.java.nio.FloatBuffer
424.java.nio.IntBuffer
425.java.nio.LongBuffer
426.java.nio.ShortBuffer
427.java.nio.charset.StandardCharsets
428.java.security.GeneralSecurityException
429.java.security.InvalidKeyException
430.java.security.InvalidParameterException
431.java.security.Key
432.java.security.KeyFactory
433.java.security.KeyManagementException
434.java.security.KeyPair
435.java.security.KeyPairGenerator
436.java.security.KeyStore
437.java.security.KeyStoreException
438.java.security.NoSuchAlgorithmException
439.java.security.NoSuchProviderException
440.java.security.Principal
441.java.security.PrivateKey
442.java.security.Provider

443.java.security.PublicKey
444.java.security.SecureRandom
445.java.security.SecureRandomSpi
446.java.security.Security
447.java.security.Signature
448.java.security.SignatureException
449.java.security.UnrecoverableKeyException
450.java.security.cert.CRLException
451.java.security.cert.CertPath
452.java.security.cert.CertPathBuilder
453.java.security.cert.CertPathBuilderException
454.java.security.cert.CertPathValidatorException
455.java.security.cert.CertStore
456.java.security.cert.Certificate
457.java.security.cert.CertificateEncodingException
458.java.security.cert.CertificateException
459.java.security.cert.CertificateExpiredException
460.java.security.cert.CertificateFactory
461.java.security.cert.CertificateNotYetValidException
462.java.security.cert.CertificateParsingException
463.java.security.cert.CertificateRevokedException
464.java.security.cert.CollectionCertStoreParameters
465.java.security.cert.PKIXBuilderParameters
466.java.security.cert.PKIXCertPathBuilderResult
467.java.security.cert.TrustAnchor
468.java.security.cert.X509CRL
469.java.security.cert.X509CertSelector
470.java.security.cert.X509Certificate
471.java.security.interfaces.ECPublicKey
472.java.security.interfaces.RSAPublicKey
473.java.security.spec.AlgorithmParameterSpec
474.java.security.spec.ECParameterSpec
475.java.security.spec.X509EncodedKeySpec
476.java.text.DateFormat
477.java.text.ParseException
478.java.text.SimpleDateFormat
479.java.util.ArrayDeque

480.java.util.ArrayList
481.java.util.Arrays
482.java.util.Calendar
483.java.util.Collection
484.java.util.Collections
485.java.util.Comparator
486.java.util.Date
487.java.util.EmptyStackException
488.java.util.EnumMap
489.java.util.EnumSet
490.java.util.Enumeration
491.java.util.Formatter
492.java.util.HashMap
493.java.util.HashSet
494.java.util.Hashtable
495.java.util.Iterator
496.java.util.LinkedList
497.java.util.List
498.java.util.Locale
499.java.util.Map
500.java.util.Map.Entry
501.java.util.Random
502.java.util.Set
503.java.util.Stack
504.java.util.StringTokenizer
505.java.util.TimeZone
506.java.util.Timer
507.java.util.TimerTask
508.java.util.TreeMap
509.java.util.TreeSet
510.java.util.UUID
511.java.util.WeakHashMap
512.java.util.concurrent.ArrayBlockingQueue
513.java.util.concurrent.Callable
514.java.util.concurrent.CopyOnWriteArrayList
515.java.util.concurrent.CountDownLatch
516.java.util.concurrent.Executor

517.java.util.concurrent.ExecutorService
518.java.util.concurrent.Executors
519.java.util.concurrent.Future
520.java.util.concurrent.LinkedBlockingQueue
521.java.util.concurrent.RejectedExecutionException
522.java.util.concurrent.Semaphore
523.java.util.concurrent.TimeUnit
524.java.util.concurrent.TimeoutException
525.java.util.concurrent.atomic.AtomicBoolean
526.java.util.concurrent.locks.Condition
527.java.util.concurrent.locks.Lock
528.java.util.concurrent.locks.ReentrantLock
529.java.util.regex.Matcher
530.java.util.regex.Pattern
531.javax.crypto.Cipher
532.javax.microedition.khronos.egl.EGLConfig
533.javax.microedition.khronos.opengles.GL10
534.javax.net.ssl.HandshakeCompletedEvent
535.javax.net.ssl.HandshakeCompletedListener
536.javax.net.ssl.HostnameVerifier
537.javax.net.ssl.HttpsURLConnection
538.javax.net.ssl.KeyManager
539.javax.net.ssl.SSLContext
540.javax.net.ssl.SSLException
541.javax.net.ssl.SSLHandshakeException
542.javax.net.ssl.SSLPeerUnverifiedException
543.javax.net.ssl.SSLProtocolException
544.javax.net.ssl.SSLSession
545.javax.net.ssl.SSLSocket
546.javax.net.ssl.TrustManager
547.javax.net.ssl.TrustManagerFactory
548.javax.net.ssl.X509ExtendedKeyManager
549.javax.net.ssl.X509TrustManager
550.javax.security.auth.x500.X500Principal
551.javax.security.cert.CertificateException
552.javax.security.cert.X509Certificate

10. Appendix: Java Library APIs

The Hypori Client uses the following library APIs from the zxing, org.json, and spongycastle java libraries:

1. Google Protocol Buffers, zxing, yubico, spongycastle
2. com.google.protobuf.ByteString
3. com.google.protobuf.CodedInputStream
4. com.google.protobuf.CodedOutputStream
5. com.google.protobuf.GeneratedMessageLite
6. com.google.protobuf.InvalidProtocolBufferException
7. com.google.zxing.BarcodeFormat
8. com.google.zxing.BinaryBitmap
9. com.google.zxing.DecodeHintType
10. com.google.zxing.MultiFormatReader
11. com.google.zxing.PlanarYUVLuminanceSource
12. com.google.zxing.ReaderException
13. com.google.zxing.Result
14. com.google.zxing.ResultMetadataType
15. com.google.zxing.ResultPoint
16. com.google.zxing.ResultPointCallback
17. com.google.zxing.client.android.CaptureFragment
18. com.google.zxing.client.android.Contents
19. com.google.zxing.client.android.Intents
20. com.google.zxing.client.android.LocaleManager
21. com.google.zxing.client.android.camera.CameraManager
22. com.google.zxing.client.android.camera.FrontLightMode
23. com.google.zxing.client.android.camera.open.OpenCameraInterface
24. com.google.zxing.client.android.result.ResultHandler
25. com.google.zxing.client.android.result.ResultHandlerFactory
26. com.google.zxing.client.android.wifi.WifiConfigManager
27. com.google.zxing.client.result.AddressBookParsedResult
28. com.google.zxing.client.result.CalendarParsedResult
29. com.google.zxing.client.result.EmailAddressParsedResult
30. com.google.zxing.client.result.ExpandedProductParsedResult
31. com.google.zxing.client.result.GeoParsedResult
32. com.google.zxing.client.result.ISBNParsedResult
33. com.google.zxing.client.result.ParsedResult
34. com.google.zxing.client.result.ParsedResultType
35. com.google.zxing.client.result.ProductParsedResult
36. com.google.zxing.client.result.ResultParser
37. com.google.zxing.client.result.SMSParsedResult
38. com.google.zxing.client.result.TelParsedResult
39. com.google.zxing.client.result.URIParsedResult
40. com.google.zxing.client.result.WifiParsedResult
41. com.google.zxing.common.HybridBinarizer
42. com.yubico.yubikit.YubiKitManager

43. com.yubico.yubikit.apdu.ApduCodeException
44. com.yubico.yubikit.apdu.ApduException
45. com.yubico.yubikit.piv.Algorithm
46. com.yubico.yubikit.piv.InvalidPinException
47. com.yubico.yubikit.piv.PivApplication
48. com.yubico.yubikit.piv.Slot
49. com.yubico.yubikit.transport.usb.UsbConfiguration
50. com.yubico.yubikit.transport.usb.UsbSession
51. com.yubico.yubikit.transport.usb.UsbSessionListener
52. org.spongycastle.asn1.ASN1InputStream
53. org.spongycastle.asn1.ASN1ObjectIdentifier
54. org.spongycastle.asn1.ASN1Primitive
55. org.spongycastle.asn1.DERIA5String
56. org.spongycastle.asn1.DEROctetString
57. org.spongycastle.asn1.x509.AlgorithmIdentifier
58. org.spongycastle.asn1.x509.CRLDistPoint
59. org.spongycastle.asn1.x509.DistributionPoint
60. org.spongycastle.asn1.x509.DistributionPointName
61. org.spongycastle.asn1.x509.Extension
62. org.spongycastle.asn1.x509.GeneralName
63. org.spongycastle.asn1.x509.GeneralNames
64. org.spongycastle.cert.X509CertificateHolder
65. org.spongycastle.cert.jcajce.JcaCertStore
66. org.spongycastle.cms.CMSException
67. org.spongycastle.cms.CMSProcessableByteArray
68. org.spongycastle.cms.CMSSignedData
69. org.spongycastle.cms.CMSSignedDataGenerator
70. org.spongycastle.cms.CMSTypedData
71. org.spongycastle.cms.jcajce.JcaSignerInfoGeneratorBuilder
72. org.spongycastle.operator.ContentSigner
73. org.spongycastle.operator.DefaultDigestAlgorithmIdentifierFinder
74. org.spongycastle.operator.DefaultSignatureAlgorithmIdentifierFinder
75. org.spongycastle.operator.OperatorCreationException
76. org.spongycastle.operator.jcajce.JcaContentSignerBuilder
77. org.spongycastle.operator.jcajce.JcaDigestCalculatorProviderBuilder
78. org.spongycastle.util.Store