Intelligent Waves Virtual Mobile Infrastructure Platform 4.1

Hypori Client (Android) Security Target

Hypori Client Android Version 4.1
August 2, 2018

Prepared by:
Intelligent Waves, Inc.
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 component of the Virtual Mobile Infrastructure Platform version 4.1 provided by Intelligent Waves, Inc.

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 – Intelligent Waves Virtual Mobile Infrastructure Platform 4.1 Hypori Client (Android) Security Target

ST Version – Version 4.1

ST Date – August 2, 2018

TOE Identification – Hypori Client (Android) 4.1

TOE Developer – Intelligent Waves, Inc.

Evaluation Sponsor – Intelligent Waves, Inc.

CC Identification – *Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 4, September 2012*

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 1.2, 22 April 2016 (PP APP SW) including DoD Annex for Protection Profile for Application Software v1.2, Version 1 Release 1, 21 February 2018

- The following NIAP Technical Decisions apply to evaluation assurance activities.
    - TD0107: FCS_CKM - ANSI X9.31-1998, Section 4.1 for Cryptographic Key Generation
    - TD0119: FCS_STO_EXT.1.1 in PP_APP_v1.2
    - TD0163: Update to FCS_TLSC_EXT.1.1 Test 5.4 and FCS_TLSS_EXT.1.1 Test
    - TD0172: Additional APIs added to FCS_RBG_EXT.1.1
    - TD0174: Optional Ciphersuites for TLS
    - TD0178: Integrity for installation tests in AppSW PP
    - TD0192: Update to FCS_STO_EXT.1 Application Note
    - TD0217: Compliance to RFC5759 and RFC5280 for using CRLs
    - TD0221: FMT_SMF.1.1 Assignments moved to Selections

- TD0238: User-modifiable files FTP_AEX_EXT.1.4
- TD0244: FCS_TLSC_EXT - TLS Client Curves Allowed
- TD0268: FMT_MEC_EXT.1 Clarification
- TD0283: Cipher Suites for TLS in SWApp v1.2
- TD0295: Update to FPT_AEX_EXT.1.3 Assurance Activities
- TD0300: Sensitive Data in FDP_DAR_EXT.1
- TD0304: Update to FCS_TLSC_EXT.1.2
- TD0305: Handling of TLS connections with and without mutual authentication
- TD0327: Default file permissions for FMT_CFG_EXT.1.2

- Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1, Revision 4, September 2012.
  - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1 Revision 4, September 2012.
  - 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.
- The PP APP SW uses an additional convention – the 'case' – which defines parts of an SFR that apply only when corresponding selections are made or some other identified conditions exist. Only the applicable cases are identified in this ST and they are identified using **bold** text.
- 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 SW 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
DEP	Data Execution Prevention
DoD	Department of Defense
OS	Operating System
PII	Personally Identifiable Information
PP	Protection Profile
PP APP SW	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 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 an Android Hypori Client 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. 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.

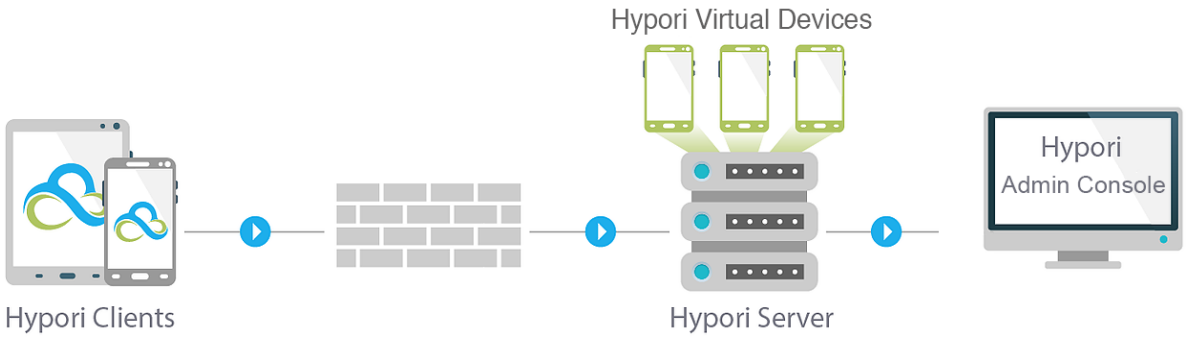

Figure 1 Hypori Virtual Mobile Infrastructure (VMI)

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.

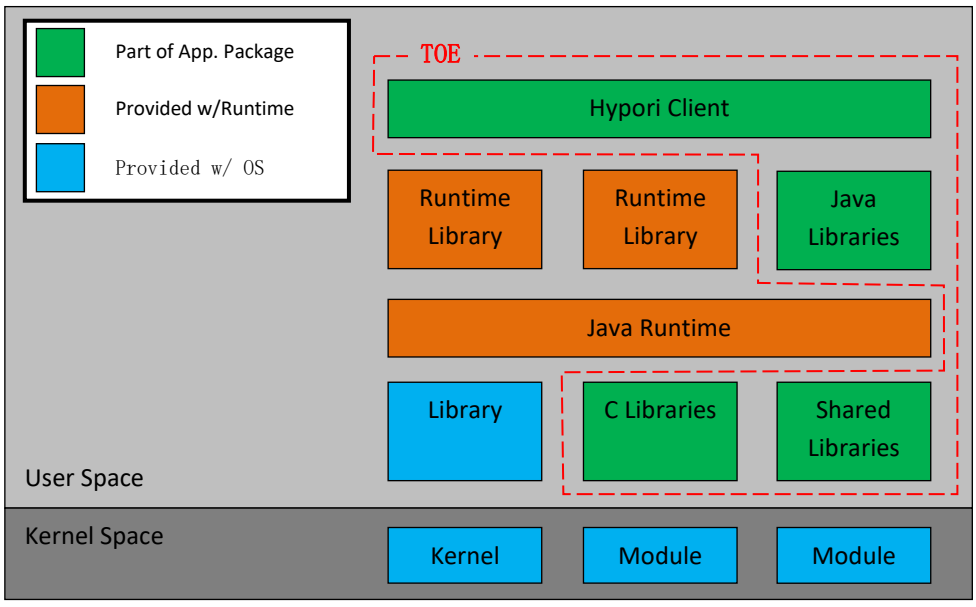

Figure 3 TOE Boundary for Android Devices

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 runs on Android versions 5.0, 5.1, 6.0, 7.0, and 7.1.

2.3.1.2 Hardware Requirements

The TOE imposes no hardware requirements beyond 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 Android Keystore System.

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 knowingly 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

TOE uses TLS 1.2 for all communication with the Hypori server.

2.4 TOE Documentation

The TOE includes the following Hypori Client documentation.

- *Hypori User Guide,* Version 4.1.0
- Hypori User Guide Common Criteria Configuration and Operation, Version 4.1

3. Security Problem Definition

This security target includes by reference the Security Problem Definition from the PP APP SW. 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 SW 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 SW 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 SW. The PP APP SW 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 SW 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 SW security objectives are suitable for the Hypori Client TOE.

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 1.2, 22 April 2016 (PP APP SW). 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 SW made a number of refinements and completed some of the SFR operations defined in the CC. PP APP SW should be consulted to identify those changes if necessary.

The security assurance requirements are the set of SARs specified in PP APP SW.

5.1 Extended Requirements

All of the extended requirements in this ST have been drawn from the PP APP SW. The PP APP SW defines the following extended SFRs. Since these SFRs are not redefined in this ST, readers should consult PP APP SW 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
- FCS_TLSC_EXT.1 TLS Client Protocol
- FCS_TLSC_EXT.2 TLS Client Protocol
- FCS_TLSC_EXT.4 TLS Client Protocol
- 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_LIB_EXT.1 Use of Third Party Libraries
- FPT_TUD_EXT.1 Integrity for Installation and Update
- 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: Cryptographic support	FCS_CKM_EXT.1 Cryptographic Key Generation Services
	FCS_RBG_EXT.1 Random Bit Generation Services
	FCS_STO_EXT.1 Storage of Credentials
	FCS_TLSC_EXT.1 TLS Client Protocol
	FCS_TLSC_EXT.2 TLS Client Protocol
	FCS_TLSC_EXT.4 TLS Client Protocol
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_LIB_EXT.1 Use of Third Party Libraries
	FPT_TUD_EXT.1 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[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[2] 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.

5.2.1.4 TLS Client Protocol (FCS_TLSC_EXT.1)

FCS_TLSC_EXT.1.1[3] The application shall [***invoke platform-provided TLS 1.2***] supporting the following cipher suites: [

TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246
TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289

[1] SFR modified per NIAP TD0107

[2] SFR Modified per NIAP TD0119

[3] This SFR was modified per NIAP TD0174 and TD0283

TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_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]

FCS_TLSC_EXT.1.2 The application shall verify that the presented identifier matches the reference identifier according to RFC 6125.

FCS_TLSC_EXT.1.3 The application shall establish a trusted channel only if the peer certificate is valid.

5.2.1.5 TLS Client Protocol (FCS_TLSC_EXT.2)

FCS_TLSC_EXT.2.1 The application shall support mutual authentication using X.509v3 certificates.

5.2.1.6 TLS Client Protocol (FCS_TLSC_EXT.4)

FCS_TLSC_EXT.4.1[4] The application shall present the **S**upported Elliptic Curves Extension in the Client Hello with the following NIST curves: [***secp256r1, secp384r1, secp521r1***].

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[5] 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 [

- ***[accounts on device]***

] .

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],***

[4] This SFR was modified per NIAP TD0244

[5] This SFR was modified per NIAP TD0300

- **respond to *[push notifications from Google's GCM platform by polling the Hypori server for notifications],***
- ***[application-initiated communication for 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 only provide enough functionality to set new credentials when configured with default credentials or no credentials.

FMT_CFG_EXT.1.2[6] The application shall be configured by default with file permissions which protect the application's binaries and data files from modification by normal unprivileged user.

5.2.3.2 Supported Configuration Mechanism (FMT_MEC_EXT.1)

FMT_MEC_EXT.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[7] 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***].

Application Note: *Note that as per the claimed PP, this requirement applies only to PII that is specifically requested by the application and does not apply if the user volunteers PII without prompting.*

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.

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*].**

[6] This SFR was modified per NIAP TD0327

[7] This SFR was modified per NIAP TD0221

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

FPT_AEX_EXT.1.4[8] 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 compiled 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 be distributed using the format of the platform-supported package manager.

FPT_TUD_EXT.1.3 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.1.4 The application shall not download, modify, replace or update its own binary code.

FPT_TUD_EXT.1.5 The application shall [***provide the ability***] to query the current version of the application software.

FPT_TUD_EXT.1.6 The application installation package and its updates shall be digitally signed such that its platform can cryptographically verify them prior to installation.

5.2.5.4 Use of Third Party Libraries (FPT_LIB_EXT.1)

FPT_LIB_EXT.1.1 The Android application shall be packaged with only [

3rd party (non-Android framework) libraries:

java libraries:

1. **protocol buffers (from Google)**
2. **json (via source code, from json.org)**
3. **ZXing (from Google for QR code scanning)**
4. **Spongycastle (renamed BC packages used for CMS message wrapping).**

C libraries:

5. **opus (audio compression)**

Shared libraries included with the TOE:

6. **libhyp_libopus-1.1.4.so**
7. **libhyp_librooted.so**

].

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 The application shall [***encrypt all transmitted data with [TLS]***] between itself and another trusted IT product.

[8] Modified per NIAP TD0238

5.2.7 Identification and authentication (FIA)

5.2.7.1 X.509 Certificate Validation (FIA_X509_EXT.1)

FIA_X509_EXT.1.1[9] 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 and that the CA flag is set to TRUE for all CA certificates.
- The application shall validate the revocation status of the certificate using [***a Certificate Revocation List (CRL) as specified in RFC 5759***].
- The application shall validate the extendedKeyUsage 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.[10]
  - 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 extendedKeyUsage 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 extendedKeyUsage 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 extendedKeyUsage field.[11]
  - 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 extendedKeyUsage field. [12]
  - 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 extendedKeyUsage field.[13]

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

---

[9] SFR modified per NIAP TD0217

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

[11] 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.

[12] The Hypori Client does not check extended key usage for OCSP Signing, since the Hypori Client does not use OCSP to check certificate revocation status.

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

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***].[14]

5.3 TOE Security Assurance Requirements

The security assurance requirements in Table 2 are included in this ST by reference from the PP APP SW.

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 SW apply to the TOE evaluation.

[14] By default, the Hypori Client validates server certification using CRLs as the default Android implementation does not perform any validation using CRLs or OCSP. However, some Android implementations that have been evaluated against the PP_MD_V3.1 will perform revocation checking when operating in their evaluated configurations. This will result in two checks, one by the Android operating system and one by the Hypori Client.

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.

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

6.1.4 FCS_TLSC_EXT.1, FCS_TLSC_EXT.2, FCS_TLSC_EXT.4

The Hypori Client relies on the platform for TLS protection of communication with the Hypori server. This includes providing the TLS client certificate to authenticate the client to the server. The Hypori server authenticates a Hypori Client. Different versions of the Android operating system support distinct sets of ciphersuites, which are shown in Table 4. The column labeled Server shows the default Hypori server configuration, which may be changed to accommodate the other identified ciphersuites.

Table 4 Ciphersuite Support by Android Version

	Ciphersuite	5.0	5.1	6.0	7.0	7.1	Server
1	TLS_DHE_RSA_WITH_AES_128_CBC_SHA256	Yes	Yes	Yes	Yes	Yes	Default
2.	TLS_DHE_RSA_WITH_AES_256_GCM_SHA384	Yes	Yes	Yes	Yes	Yes	Configurable
3.	TLS_DHE_RSA_WITH_AES_256_CBC_SHA256	Yes	Yes	Yes	Yes	Yes	Default
4.	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	Yes	Yes	Yes	Yes	Yes	Default
5.	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	Yes	Yes	Yes	Yes	Yes	Default
6.	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	Yes	Yes	Yes	Yes	Yes	Default
7.	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	Yes	Yes	Yes	Yes	Yes	Default
8.	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	Yes	Yes	Yes	Yes	Yes	Configurable
9.	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	Yes	Yes	Yes	Yes	Yes	Configurable
10.	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	Yes	Yes	Yes	Yes	Yes	Configurable
11.	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	Yes	Yes	Yes	Yes	Yes	Default

For elliptic curve ciphersuites, the Hypori Client relies on the platform for elliptic curves. The Android platforms support NIST curves secp256r1, secp384r1, and secp521r1 and Supported Elliptic Curves Extension for TLS. No configuration is required by a Hypori Client user.

The Hypori Client establishes the reference identifier using the configured server host name. The Hypori Client validates the first CN and the subject alternative names against the configured reference identifier. It supports wildcards and IP addresses. Pinning is not supported in the client.

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 6.1.2. 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:

Table 5: Android Permissions

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.
C2D_MESSAGE	Prevents other Android apps from registering and receiving Hypori Client messages.
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.
GET_TASKS (Deprecated)	Allow the app to retrieve information about currently and recently running tasks
GET_ACCOUNTS (Deprecated)	Access to the list of accounts in the Accounts Service.

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.[15] 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.

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 certification path validation as part of the TLS service. The platform certificate path algorithm is described by its Android platform source code:

> https://android.googlesource.com/platform/external/conscrypt/+/android-5.0.0_r7/src/platform/java/org/conscrypt/TrustManagerImpl.java

See the checkTrusted() method at line 249 for the algorithm. 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 CRL provided by the CRL distribution point specified in the certificates. 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.

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.) 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. If the CRL 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

[15] 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, PP APP SW distinguishes credentials from PII.

key from a certificate file using the platform's certificate services. A user's IT group provides the user with a server account password.

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 protocols in a manner consistent with DoD Ports and Protocols guidance, including the DoD Ports Protocols Services Management (PPSM) Category Assurance List (CAL). 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 knowingly transmit PII over a network. A user may use the Hypori Client to transmit PII but it is an operation specifically requested by the user; the TSF does not include any data input interfaces that are designed specifically to obtain PII.

6.5.1 FPR_ANO_EXT.1

The Hypori Client does not knowingly 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

Intelligent Waves 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. Intelligent Waves 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.

6.6.3 FPT_LIB_EXT.1

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

6.6.4 FPT_TUD_EXT.1

Intelligent Waves distributes the Hypori Client as an .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. Intelligent Waves 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. The certificate information:

```
X.509, CN=Joe Smith, OU=Development, O=Intelligent Waves LLC, C=US
      [certificate is valid from 11/06/17 14:53 PM to 10/31/42 15:53 PM]
      [CertPath not validated: Path does not chain with any of the trust
anchors]
```

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

Intelligent Waves provides customers with timely updates. A customer chooses their preferred communication. The Intelligent Waves 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.

Intelligent Waves maintains a Security Portal online. Every customer is registered with the Support Portal. Intelligent Waves notifies each customer of a new security report on the Support portal using the customers preferred communication mechanism. Intelligent Waves 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.

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.

7. Protection Profile Claims

This ST conforms to the *Protection Profile for Application Software,* Version 1.2, 22 April 2016 (PP APP SW)

As explained in Section 3, Security Problem Definition, the Security Problem Definition of the PP APP SW has been included by reference into this ST.

As explained in Section 4, Security Objectives, the Security Objectives of the PP APP SW 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 SW and operations completed as appropriate.

Table 6 SFR Protection Profile Sources

Requirement Class	Requirement Component	Source
FCS: Cryptographic support	FCS_CKM_EXT.1 Cryptographic Key Generation Services	PP APP SW
	FCS_RBG_EXT.1 Random Bit Generation Services	PP APP SW
	FCS_STO_EXT.1 Storage of Credentials	PP APP SW
	FCS_TLSC_EXT.1 TLS Client Protocol	PP APP SW
	FCS_TLSC_EXT.2 TLS Client Protocol	PP APP SW
	FCS_TLSC_EXT.4 TLS Client Protocol	PP APP SW
FDP: User data protection	FDP_DAR_EXT.1 Encryption of Sensitive Application Data	PP APP SW
	FDP_DEC_EXT.1 Access to Platform Resources	PP APP SW
	FDP_NET_EXT.1 Network Communications	PP APP SW
FIA: Identification and authentication	FIA_X509_EXT.1 X.509 Certificate Validation	PP APP SW
	FIA_X509_EXT.2 X.509 Certificate Authentication	PP APP SW
FMT: Security management	FMT_CFG_EXT.1 Secure by Default Configuration	PP APP SW
	FMT_MEC_EXT.1 Supported Configuration Mechanism	PP APP SW
	FMT_SMF.1 Specification of Management Functions	PP APP SW
FPR: Privacy	FPR_ANO_EXT.1 User Consent for Transmission of Personally Identifiable Information	PP APP SW
FPT: Protection of the TSF	FPT_AEX_EXT.1 AntiExploitation Capabilities	PP APP SW
	FPT_API_EXT.1.1 Use of Supported Services and APIs	PP APP SW
	FPT_LIB_EXT.1 Use of Third Party Libraries	PP APP SW
	FPT_TUD_EXT.1 Integrity for Installation and Update	PP APP SW
FTP: Trusted path/channels	FTP_DIT_EXT.1 Protection of Data in Transit	PP APP SW

8. Rationale

This security target includes by reference the PP APP SW Security Problem Definition, Security Objectives, and Security Assurance Requirements. The security target makes no additions to the PP APP SW assumptions. PP APP SW security functional requirements have been reproduced with the PP APP SW operations completed. Operations on the security requirements follow PP APP SW application notes and assurance activities. Consequently, PP APP SW 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

PP APP SW includes mandatory, selection-based, optional, selection based, and objective requirements. The ST includes all mandatory requirements as well as all selection-based requirements that apply to selections made in the ST. The ST includes optional elements (FCS_TLSC_EXT.2.1 and FCS_TLSC_EXT.4.1) and no objective requirements.

Table 7 presents rationale demonstrating that this ST includes all applicable selection-base requirements. The requirements column lists all PP APP SW security functional requirements. SFRs listed in italics are not included in this ST. The type of requirement is identified by M, Op, SB, and Ob, which stand for mandatory, optional, selection-based, and objective, respectively. The table lists dependencies of each requirement together with the condition that establishes the dependency. Finally, the table includes whether each dependency is satisfied or not applicable along with a brief rationale.

Table 7 Rationale for Selection-Based Requirements

Requirement	Type	Dependency	Condition	Satisfied	Rationale
FCS_RBG_EXT.1	M	FCS_RBG_EXT.2	Select implement DRBG	Not applicable	Not selected
FCS_RBG_EXT.2	*SB*	*None*	*None*	*Not applicable*	
FCS_STO_EXT.1	M	FCS_COP.1(1)	Select implement secure store	Not applicable	Not selected
FCS_TLSC_EXT.1	SB	FCS_CKM_EXT.1	Unconditional	Satisfied	Included in ST
FCS_TLSC_EXT.4	SB	FCS_TLSC_EXT.4.1	Select elliptic curve ciphersuites	Satisfied	Included in ST
FCS_TLSC_EXT.1	SB	FCS_CKM.1	Select implement TLS	Not applicable	Not selected
FCS_TLSC_EXT.1	SB	FCS_CKM.2	Select implement TLS	Not applicable	Not selected
FCS_TLSC_EXT.1	SB	FCS_COP.1(1)	Select implement TLS	Not applicable	Not selected
FCS_TLSC_EXT.1	SB	FCS_COP.1(2)	Select implement TLS	Not applicable	Not selected
FCS_TLSC_EXT.1	SB	FCS_COP.1(3)	Select implement TLS	Not applicable	Not selected
FCS_TLSC_EXT.1	SB	FCS_COP.1(4)	Select implement	Not applicable	Not selected

Requirement	Type	Dependency	Condition	Satisfied	Rationale
			TLS		
FCS_TLSC_EXT.1	SB	FIA_X509_EXT.1	Unconditional	Satisfied	Included in ST
FCS_TLSC_EXT.2.1	Op	FIA_X509_EXT.2	Unconditional	Satisfied	Included in ST
FCS_CKM_EXT.1	SB	FCS_CKM.1	Select invoke or implement key generation	Not applicable	Not selected
FCS_CKM.1	*Op*	*None*	*None*	*Not applicable*	
FCS_CKM.2	*Op*	*None*	*None*	*Not applicable*	
FCS_COP.1(1)	*Op*	*None*	*None*	*Not applicable*	
FCS_COP.1(2)	*Op*	*None*	*None*	*Not applicable*	
FCS_COP.1(3)	*Op*	*None*	*None*	*Not applicable*	
FCS_COP.1(4)	*Op*	*None*	*None*	*Not applicable*	
FCS_DTLS_EXT.1	*Op*	*FCS_TLSC_EXT.1*	*Unconditional*	*Not applicable*	*SFR not in ST*
FCS_HTTPS_EXT.1	*Op*	*FCS_TLSC_EXT.1*	*Unconditional*	*Not applicable*	*SFR not in ST*
FDP_DEC_EXT.1	M	None	None	Not applicable	
FDP_DAR_EXT.1	M	File Encryption EP	Select implement encrypt sensitive data	Not applicable	Not selected
FDP_NET_EXT.1	M	None	None	Not applicable	
FIA_X509_EXT.1	SB	None	None	Not applicable	
FIA_X509_EXT.2	SB	None	None	Not applicable	
FMT_MEC_EXT.1.1	M	None	None	Not applicable	
FMT_CFG_EXT.1	M	None	None	Not applicable	
FMT_SMF.1	M	None	None	Not applicable	
FPR.ANO.EXT.1	M	None	None	Not applicable	
FPT_API_EXT.1.1	M	None	None	Not applicable	
FPT_API_EXT.1.2	*Ob*	*None*	*None*	*Not applicable*	
FPT_AEX_EXT.1	M	None	None	Not applicable	
FPT_TUD_EXT.	M	None	None	Not applicable	
FPT_LIB_EXT.1	M	None	None	Not applicable	
FPT_IDV_EXT.1	*Ob*	*None*	*None*	*Not applicable*	
FTP_DIT_EXT.1	M	FCS_HTTPS_EXT.1	Select HTTPS	Not applicable	Not selected
FTP_DIT_EXT.1	M	FCS_TLSC_EXT.1	Select TLS	Satisfied	Included in ST
FTP_DIT_EXT.1	M	FCS_DTLS_EXT.1	Select DTLS	Not applicable	Not selected

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 8 demonstrates the relationship between security requirements and security functions.

Table 8 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						
FCS_TLSC_EXT.1	X						
FCS_TLSC_EXT.2	X						
FCS_TLSC_EXT.4	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_LIB_EXT.1						X	
FPT_TUD_EXT.1						X	
FTP_DIT_EXT.1							X

9. Appendix: Android APIs

The Hypori Client uses the following Android APIs:

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

41. android.bluetooth.BluetoothAdapter
42. android.bluetooth.BluetoothClass
43. android.bluetooth.BluetoothDevice
44. android.bluetooth.BluetoothGatt
45. android.bluetooth.BluetoothGattCallback
46. android.bluetooth.BluetoothGattCharacteristic
47. android.bluetooth.BluetoothGattDescriptor
48. android.bluetooth.BluetoothGattService
49. android.bluetooth.BluetoothProfile
50. android.bluetooth.BluetoothSocket
51. android.content.AbstractThreadedSyncAdapter
52. android.content.ActivityNotFoundException
53. android.content.BroadcastReceiver
54. android.content.ClipData
55. android.content.ComponentName
56. android.content.ContentProvider
57. android.content.ContentProviderClient
58. android.content.ContentResolver
59. android.content.ContentUris
60. android.content.ContentValues
61. android.content.Context
62. android.content.DialogInterface
63. android.content.DialogInterface.OnClickListener
64. android.content.Intent
65. android.content.Intent.ShortcutIconResource
66. android.content.IntentFilter
67. android.content.ServiceConnection
68. android.content.SharedPreferences
69. android.content.SharedPreferences.Editor
70. android.content.SharedPreferences.OnSharedPreferenceChangeListener
71. android.content.SyncResult
72. android.content.pm.ActivityInfo
73. android.content.pm.ApplicationInfo
74. android.content.pm.PackageInfo
75. android.content.pm.PackageManager
76. android.content.pm.PackageManager.NameNotFoundException
77. android.content.pm.ResolveInfo
78. android.content.pm.Signature
79. android.content.res.AssetFileDescriptor
80. android.content.res.AssetManager
81. android.content.res.Configuration
82. android.content.res.Resources
83. android.content.res.TypedArray
84. android.database.ContentObserver

85. android.database.Cursor
86. android.database.DataSetObserver
87. android.database.MatrixCursor
88. android.database.sqlite.SQLiteDatabase
89. android.database.sqlite.SQLiteException
90. android.database.sqlite.SQLiteOpenHelper
91. android.graphics.Bitmap
92. android.graphics.Bitmap.CompressFormat
93. android.graphics.BitmapFactory
94. android.graphics.Canvas
95. android.graphics.Color
96. android.graphics.ImageFormat
97. android.graphics.Matrix
98. android.graphics.Paint
99. android.graphics.Path
100. android.graphics.PixelFormat
101. android.graphics.Point
102. android.graphics.PointF
103. android.graphics.PorterDuff
104. android.graphics.Rect
105. android.graphics.RectF
106. android.graphics.SurfaceTexture
107. android.graphics.SurfaceTexture.OnFrameAvailableListener
108. android.graphics.Typeface
109. android.graphics.YuvImage
110. android.graphics.drawable.BitmapDrawable
111. android.graphics.drawable.ColorDrawable
112. android.graphics.drawable.Drawable
113. android.graphics.drawable.ShapeDrawable
114. android.graphics.drawable.TransitionDrawable
115. android.graphics.drawable.shapes.RoundRectShape
116. android.hardware.Camera
117. android.hardware.Camera.Area
118. android.hardware.Camera.CameraInfo
119. android.hardware.Camera.Face
120. android.hardware.Camera.FaceDetectionListener
121. android.hardware.Camera.Parameters
122. android.hardware.Camera.PictureCallback
123. android.hardware.Camera.Size
124. android.hardware.Sensor
125. android.hardware.SensorEvent
126. android.hardware.SensorEventListener
127. android.hardware.SensorManager
128. android.hardware.fingerprint.FingerprintManager

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.MediaFormat
148.android.media.MediaMetadataRetriever
149.android.media.MediaPlayer
150.android.media.MediaRecorder
151.android.media.ThumbnailUtils
152.android.media.audiofx.AcousticEchoCanceler
153.android.media.audiofx.AutomaticGainControl
154.android.media.audiofx.NoiseSuppressor
155.android.net.ConnectivityManager
156.android.net.LocalSocket
157.android.net.LocalSocketAddress
158.android.net.NetworkInfo
159.android.net.Uri
160.android.net.Uri.Builder
161.android.net.wifi.WifiConfiguration
162.android.net.wifi.WifiInfo
163.android.net.wifi.WifiManager
164.android.net.wifi.WifiManager.WifiLock
165.android.opengl.EGL14
166.android.opengl.EGLConfig
167.android.opengl.EGLContext
168.android.opengl.EGLDisplay
169.android.opengl.EGLExt
170.android.opengl.EGLSurface
171.android.opengl.GLES11Ext
172.android.opengl.GLES20

173.android.opengl.GLSurfaceView
174.android.opengl.GLSurfaceView.Renderer
175.android.opengl.GLUtils
176.android.opengl.Matrix
177.android.os.AsyncTask
178.android.os.BatteryManager
179.android.os.Binder
180.android.os.Build
181.android.os.Bundle
182.android.os.CancellationSignal
183.android.os.Environment
184.android.os.Handler
185.android.os.HandlerThread
186.android.os.IBinder
187.android.os.Looper
188.android.os.Message
189.android.os.Parcel
190.android.os.ParcelFileDescriptor
191.android.os.ParcelFileDescriptor.AutoCloseOutputStream
192.android.os.Parcelable
193.android.os.PowerManager
194.android.os.Process
195.android.os.StatFs
196.android.os.SystemClock
197.android.os.UserHandle
198.android.os.UserManager
199.android.os.Vibrator
200.android.preference.CheckBoxPreference
201.android.preference.EditTextPreference
202.android.preference.Preference
203.android.preference.Preference.OnPreferenceChangeListener
204.android.preference.Preference.OnPreferenceClickListener
205.android.preference.PreferenceFragment
206.android.preference.PreferenceGroup
207.android.preference.PreferenceManager
208.android.provider.ContactsContract
209.android.provider.MediaStore
210.android.provider.MediaStore.Images
211.android.provider.MediaStore.Images.ImageColumns
212.android.provider.MediaStore.MediaColumns
213.android.provider.MediaStore.Video
214.android.provider.MediaStore.Video.VideoColumns
215.android.provider.Settings
216.android.provider.Settings.Secure

217.android.provider.Settings.System
218.android.security.KeyChain
219.android.security.KeyChainAliasCallback
220.android.security.KeyChainException
221.android.service.notification.StatusBarNotification
222.android.support.annotation.NonNull
223.android.support.multidex.MultiDex
224.android.support.multidex.MultiDexApplication
225.android.support.v13.app.FragmentCompat
226.android.support.v4.app.ActivityCompat
227.android.support.v4.app.DialogFragment
228.android.support.v4.app.Fragment
229.android.support.v4.app.FragmentActivity
230.android.support.v4.app.FragmentManager
231.android.support.v4.app.FragmentStatePagerAdapter
232.android.support.v4.app.NotificationCompat
233.android.support.v4.content.ContextCompat
234.android.support.v4.content.LocalBroadcastManager
235.android.support.v4.view.GravityCompat
236.android.support.v4.view.MotionEventCompat
237.android.support.v4.view.ViewConfigurationCompat
238.android.support.v4.view.ViewPager
239.android.support.v4.widget.DrawerLayout
240.android.support.v7.widget.LinearLayoutManager
241.android.support.v7.widget.RecyclerView
242.android.telephony.PhoneNumberUtils
243.android.telephony.PhoneStateListener
244.android.telephony.ServiceState
245.android.telephony.SignalStrength
246.android.telephony.TelephonyManager
247.android.text.InputFilter
248.android.text.Spannable
249.android.text.SpannableString
250.android.text.Spanned
251.android.text.TextUtils
252.android.text.format.DateFormat
253.android.text.format.Time
254.android.text.style.StyleSpan
255.android.util.AttributeSet
256.android.util.Base64
257.android.util.Base64OutputStream
258.android.util.DisplayMetrics
259.android.util.Log
260.android.util.SparseArray

261.android.util.TypedValue
262.android.util.Xml
263.android.view.Display
264.android.view.DragEvent
265.android.view.GestureDetector
266.android.view.Gravity
267.android.view.InflateException
268.android.view.KeyEvent
269.android.view.LayoutInflater
270.android.view.Menu
271.android.view.MenuInflater
272.android.view.MenuItem
273.android.view.MenuItem.OnMenuItemClickListener
274.android.view.MotionEvent
275.android.view.OrientationEventListener
276.android.view.SoundEffectConstants
277.android.view.Surface
278.android.view.SurfaceHolder
279.android.view.SurfaceView
280.android.view.VelocityTracker
281.android.view.View
282.android.view.View.DragShadowBuilder
283.android.view.View.OnClickListener
284.android.view.View.OnDragListener
285.android.view.View.OnTouchListener
286.android.view.ViewConfiguration
287.android.view.ViewGroup
288.android.view.ViewGroup.LayoutParams
289.android.view.ViewParent
290.android.view.Window
291.android.view.WindowManager
292.android.view.accessibility.AccessibilityEvent
293.android.view.animation.AccelerateInterpolator
294.android.view.animation.AlphaAnimation
295.android.view.animation.Animation
296.android.view.animation.Animation.AnimationListener
297.android.view.animation.AnimationUtils
298.android.view.animation.DecelerateInterpolator
299.android.view.inputmethod.EditorInfo
300.android.view.inputmethod.InputMethodManager
301.android.webkit.WebSettings
302.android.webkit.WebView
303.android.widget.AbsListView
304.android.widget.AdapterView

305.android.widget.AdapterView.OnItemClickListener
306.android.widget.AdapterView.OnItemLongClickListener
307.android.widget.ArrayAdapter
308.android.widget.BaseAdapter
309.android.widget.Button
310.android.widget.CheckBox
311.android.widget.CompoundButton
312.android.widget.CompoundButton.OnCheckedChangeListener
313.android.widget.CursorAdapter
314.android.widget.EditText
315.android.widget.FrameLayout
316.android.widget.GridView
317.android.widget.HorizontalScrollView
318.android.widget.ImageButton
319.android.widget.ImageView
320.android.widget.ImageView.ScaleType
321.android.widget.LinearLayout
322.android.widget.ListView
323.android.widget.PopupMenu
324.android.widget.PopupWindow
325.android.widget.ProgressBar
326.android.widget.RelativeLayout
327.android.widget.ScrollView
328.android.widget.SearchView
329.android.widget.SimpleAdapter
330.android.widget.Switch
331.android.widget.TabWidget
332.android.widget.TextView
333.android.widget.Toast
334.com.google.android.gms.common.ConnectionResult
335.com.google.android.gms.common.GoogleApiAvailability
336.com.google.android.gms.gcm.GcmListenerService
337.com.google.android.gms.gcm.GcmPubSub
338.com.google.android.gms.gcm.GoogleCloudMessaging
339.com.google.android.gms.iid.InstanceID
340.com.google.android.gms.iid.InstanceIDListenerService
341.com.google.protobuf.ByteString
342.com.google.protobuf.CodedInputStream
343.com.google.protobuf.CodedOutputStream
344.com.google.protobuf.GeneratedMessageLite
345.com.google.protobuf.InvalidProtocolBufferException
346.java.beans.PropertyChangeEvent
347.java.beans.PropertyChangeListener
348.java.io.BufferedInputStream

349.java.io.BufferedOutputStream
350.java.io.BufferedReader
351.java.io.BufferedWriter
352.java.io.ByteArrayInputStream
353.java.io.ByteArrayOutputStream
354.java.io.Closeable
355.java.io.DataInputStream
356.java.io.DataOutputStream
357.java.io.File
358.java.io.FileDescriptor
359.java.io.FileInputStream
360.java.io.FileNotFoundException
361.java.io.FileOutputStream
362.java.io.FileReader
363.java.io.FileWriter
364.java.io.FilenameFilter
365.java.io.IOException
366.java.io.InputStream
367.java.io.InputStreamReader
368.java.io.ObjectInputStream
369.java.io.ObjectOutputStream
370.java.io.OutputStream
371.java.io.PrintStream
372.java.io.PrintWriter
373.java.io.RandomAccessFile
374.java.io.Serializable
375.java.io.StringWriter
376.java.io.UnsupportedEncodingException
377.java.io.Writer
378.java.lang.Thread.UncaughtExceptionHandler
379.java.lang.annotation.ElementType
380.java.lang.annotation.Retention
381.java.lang.annotation.RetentionPolicy
382.java.lang.annotation.Target
383.java.lang.ref.WeakReference
384.java.lang.reflect.Array
385.java.lang.reflect.Constructor
386.java.lang.reflect.Field
387.java.lang.reflect.InvocationTargetException
388.java.lang.reflect.Method
389.java.math.BigInteger
390.java.net.ConnectException
391.java.net.HttpURLConnection
392.java.net.InetAddress

393.java.net.MalformedURLException
394.java.net.Socket
395.java.net.SocketException
396.java.net.URL
397.java.net.URLEncoder
398.java.net.UnknownHostException
399.java.nio.BufferOverflowException
400.java.nio.BufferUnderflowException
401.java.nio.ByteBuffer
402.java.nio.ByteOrder
403.java.nio.CharBuffer
404.java.nio.DoubleBuffer
405.java.nio.FloatBuffer
406.java.nio.IntBuffer
407.java.nio.LongBuffer
408.java.nio.ShortBuffer
409.java.security.GeneralSecurityException
410.java.security.InvalidKeyException
411.java.security.InvalidParameterException
412.java.security.Key
413.java.security.KeyFactory
414.java.security.KeyManagementException
415.java.security.KeyPair
416.java.security.KeyPairGenerator
417.java.security.KeyStore
418.java.security.KeyStoreException
419.java.security.NoSuchAlgorithmException
420.java.security.NoSuchProviderException
421.java.security.Principal
422.java.security.PrivateKey
423.java.security.Provider
424.java.security.PublicKey
425.java.security.SecureRandom
426.java.security.SecureRandomSpi
427.java.security.Security
428.java.security.Signature
429.java.security.SignatureException
430.java.security.UnrecoverableKeyException
431.java.security.cert.CertPathValidatorException
432.java.security.cert.Certificate
433.java.security.cert.CertificateEncodingException
434.java.security.cert.CertificateException
435.java.security.cert.CertificateExpiredException
436.java.security.cert.CertificateNotYetValidException

437.java.security.cert.X509Certificate
438.java.security.interfaces.RSAPublicKey
439.java.security.spec.AlgorithmParameterSpec
440.java.security.spec.X509EncodedKeySpec
441.java.text.DateFormat
442.java.text.ParseException
443.java.text.SimpleDateFormat
444.java.util.ArrayDeque
445.java.util.ArrayList
446.java.util.Arrays
447.java.util.Calendar
448.java.util.Collection
449.java.util.Collections
450.java.util.Comparator
451.java.util.Date
452.java.util.EmptyStackException
453.java.util.EnumMap
454.java.util.EnumSet
455.java.util.Enumeration
456.java.util.Formatter
457.java.util.HashMap
458.java.util.HashSet
459.java.util.Hashtable
460.java.util.Iterator
461.java.util.LinkedList
462.java.util.List
463.java.util.Locale
464.java.util.Map
465.java.util.Map.Entry
466.java.util.Random
467.java.util.Set
468.java.util.Stack
469.java.util.StringTokenizer
470.java.util.TimeZone
471.java.util.Timer
472.java.util.TimerTask
473.java.util.TreeMap
474.java.util.TreeSet
475.java.util.UUID
476.java.util.WeakHashMap
477.java.util.concurrent.ArrayBlockingQueue
478.java.util.concurrent.Callable
479.java.util.concurrent.CopyOnWriteArrayList
480.java.util.concurrent.CountDownLatch

481.java.util.concurrent.ExecutorService
482.java.util.concurrent.Executors
483.java.util.concurrent.Future
484.java.util.concurrent.LinkedBlockingQueue
485.java.util.concurrent.RejectedExecutionException
486.java.util.concurrent.Semaphore
487.java.util.concurrent.TimeUnit
488.java.util.concurrent.TimeoutException
489.java.util.concurrent.atomic.AtomicBoolean
490.java.util.concurrent.locks.Condition
491.java.util.concurrent.locks.Lock
492.java.util.concurrent.locks.ReentrantLock
493.java.util.regex.Matcher
494.java.util.regex.Pattern
495.javax.crypto.Cipher
496.javax.crypto.KeyGenerator
497.javax.crypto.SecretKey
498.javax.crypto.spec.IvParameterSpec
499.javax.crypto.spec.SecretKeySpec
500.javax.microedition.khronos.egl.EGLConfig
501.javax.microedition.khronos.opengles.GL10
502.javax.net.ssl.HandshakeCompletedEvent
503.javax.net.ssl.HandshakeCompletedListener
504.javax.net.ssl.HostnameVerifier
505.javax.net.ssl.HttpsURLConnection
506.javax.net.ssl.KeyManager
507.javax.net.ssl.SSLContext
508.javax.net.ssl.SSLException
509.javax.net.ssl.SSLHandshakeException
510.javax.net.ssl.SSLPeerUnverifiedException
511.javax.net.ssl.SSLProtocolException
512.javax.net.ssl.SSLSession
513.javax.net.ssl.SSLSocket
514.javax.net.ssl.SSLSocketFactory
515.javax.net.ssl.TrustManager
516.javax.net.ssl.TrustManagerFactory
517.javax.net.ssl.X509ExtendedKeyManager
518.javax.net.ssl.X509TrustManager
519.javax.security.auth.x500.X500Principal
520.javax.security.cert.CertificateException
521.javax.security.cert.X509Certificate
522.org.xmlpull.v1.XmlPullParser
523.org.xmlpull.v1.XmlPullParserException

10. Appendix: Java Library APIs

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

1. com.google.zxing.BarcodeFormat
2. com.google.zxing.BinaryBitmap
3. com.google.zxing.DecodeHintType
4. com.google.zxing.MultiFormatReader
5. com.google.zxing.PlanarYUVLuminanceSource
6. com.google.zxing.ReaderException
7. com.google.zxing.Result
8. com.google.zxing.ResultMetadataType
9. com.google.zxing.ResultPoint
10. com.google.zxing.ResultPointCallback
11. com.google.zxing.client.android.CaptureFragment
12. com.google.zxing.client.android.Contents
13. com.google.zxing.client.android.Intents
14. com.google.zxing.client.android.LocaleManager
15. com.google.zxing.client.android.camera.CameraManager
16. com.google.zxing.client.android.camera.FrontLightMode
17. com.google.zxing.client.android.camera.open.OpenCameraInterface
18. com.google.zxing.client.android.result.ResultHandler
19. com.google.zxing.client.android.result.ResultHandlerFactory
20. com.google.zxing.client.android.wifi.WifiConfigManager
21. com.google.zxing.client.result.AddressBookParsedResult
22. com.google.zxing.client.result.CalendarParsedResult
23. com.google.zxing.client.result.EmailAddressParsedResult
24. com.google.zxing.client.result.ExpandedProductParsedResult
25. com.google.zxing.client.result.GeoParsedResult
26. com.google.zxing.client.result.ISBNParsedResult
27. com.google.zxing.client.result.ParsedResult
28. com.google.zxing.client.result.ParsedResultType
29. com.google.zxing.client.result.ProductParsedResult
30. com.google.zxing.client.result.ResultParser
31. com.google.zxing.client.result.SMSParsedResult
32. com.google.zxing.client.result.TelParsedResult
33. com.google.zxing.client.result.URIParsedResult
34. com.google.zxing.client.result.WifiParsedResult
35. com.google.zxing.common.HybridBinarizer
36. org.dcjson.JSONArray
37. org.dcjson.JSONException
38. org.dcjson.JSONObject
39. org.dcjson.JSONStringer
40. org.spongycastle.asn1.ASN1ObjectIdentifier
41. org.spongycastle.cert.jcajce.JcaCertStore

42. org.spongycastle.cms.CMSException
43. org.spongycastle.cms.CMSProcessableByteArray
44. org.spongycastle.cms.CMSSignedData
45. org.spongycastle.cms.CMSSignedDataGenerator
46. org.spongycastle.cms.CMSTypedData
47. org.spongycastle.cms.jcajce.JcaSignerInfoGeneratorBuilder
48. org.spongycastle.operator.ContentSigner
49. org.spongycastle.operator.OperatorCreationException
50. org.spongycastle.operator.jcajce.JcaContentSignerBuilder
51. org.spongycastle.operator.jcajce.JcaDigestCalculatorProviderBuilder
52. org.spongycastle.util.Store

