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

Version 1.0
March 15, 2021

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

Prepared by:
Leidos Inc.
Common Criteria Testing Laboratory
6841 Benjamin Franklin Drive, Columbia, Maryland 21046

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 (Windows) component of the Virtual Mobile Infrastructure Platform version 4.2.0 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: Windows APIs (Section 9).

1.1 Security Target, TOE and CC Identification

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

ST Version – Version 1.0

ST Date – March 15, 2021

TOE Identification – Hypori Client (Windows) 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].

This ST and the TOE it describes are conformant to the following CC specifications:

- 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

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
- TD0543: FMT_MEC_EXT.1 evaluation activity update
- TD0521: Updates to Certificate Revocation (FIA_X509_EXT.1)
- 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
- TD0465: Configuration Storage for .NET Apps
- TD0445: User Modifiable File Definition
- TD0444: IPsec selections
- TD0437: Supported Configuration Mechanism
- TD0434: Windows Desktop Applications Test
- TD0427: Reliable Time Source
- TD0416: Correction to FCS_RBG_EXT.1 Test Activity

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

- TD0540: Expanded AES Modes in FCS_COP
- TD0519: Linux symbolic links and FMT_CFG_EXT.1
- TD0515: Use Android APK manifest in test
- TD0510: Obtaining random bytes for iOS/macOS
- TD0473: Support for Client or Server TOEs in FCS_HTTPS_EXT
- TD0435: Alternative to SELinux for FPT_AEX_EXT.1.3

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
UWP	Universal Windows Platform
VMI	Virtual Mobile Infrastructure
WNS	Windows Push Notification Services

2. TOE Description

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

2.1 Product Overview

The TOE is the Hypori Client (Windows) v4.2.0. The TOE is a software program as specified in the [PP_APP_v1.3] which is a Windows-based thin client that installs on an end user's Windows device and communicates only with the Hypori Virtual Device on the server through platform provided secure TLS 1.2 encrypted protocols.

The Hypori Client (Windows) v4.2.0 uses cryptographic services provided by the Windows platform. The user initiates a secure network connection to the Hypori server using the TOE, which uses the platform's certificate validation services to authenticate the X.509 certificate from the Hypori server as part of establishing a TLS connection.

Security management consists of setting Hypori Client configuration options while sensitive data resides on the Hypori server and not the Hypori Client, although the client does store credentials. The TOE does not transmit PII over a network. The TOE uses security features and APIs provided by the platform. The TOE leverages package management for secure installation and updates.

The TOE type (software application) is consistent with the TOE type described in [PP_APP_v1.3].

In the Hypori Virtual Mobile Infrastructure (VMI) platform, end users running a Hypori Client (Windows) on their Windows 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 (Windows). The Hypori Windows thin client application provides secure access to the remote Android virtual device and brokers access between the Windows 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.

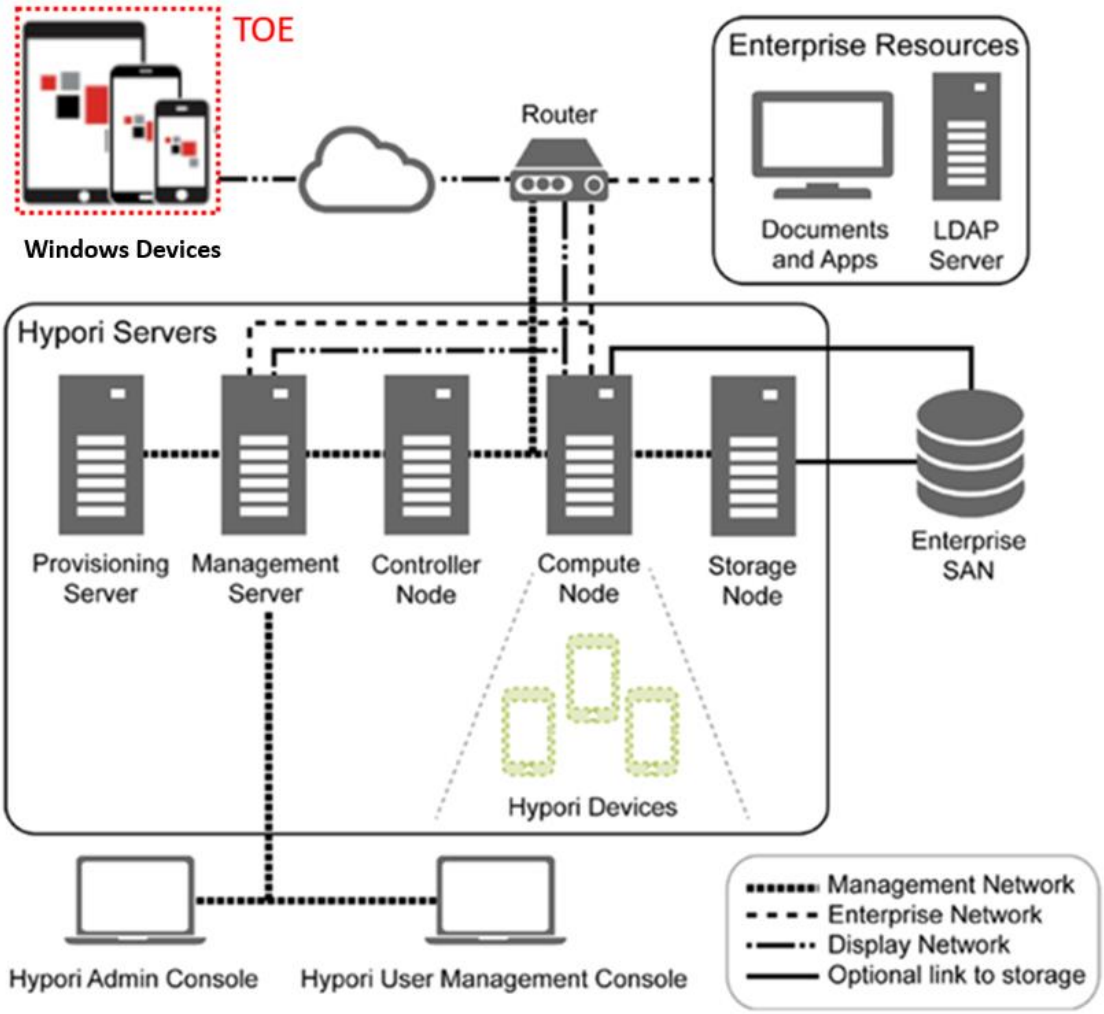

Figure 1 Hypori Virtual Mobile Infrastructure (VMI)

The Hypori VMI platform includes the following components:

- **Hypori Client:** This is a Windows-based thin client that installs on the end user's Windows device and communicates with the Hypori Virtual Device on the server through secure encrypted protocols.
- **Hypori Virtual Device:** This is an Android-based virtualized device executing on a server in the cloud.
- **Hypori Servers:** This is the RHEL 7.7 cloud server cluster that hosts the Hypori Android 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 Windows-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

This 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.

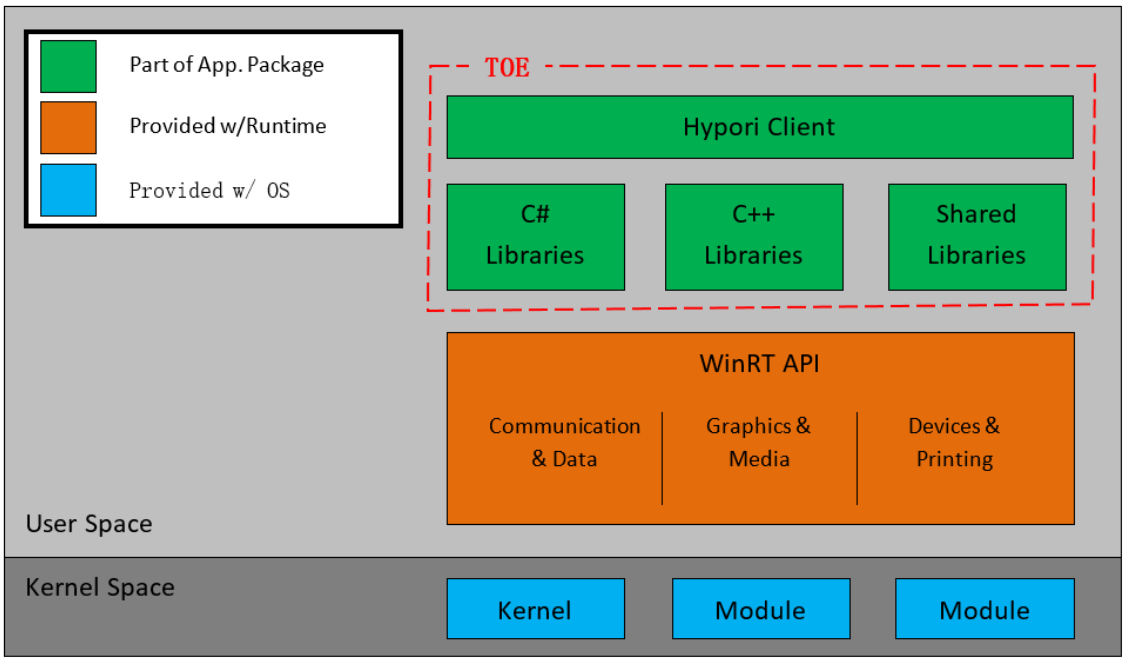

Figure 3 TOE Boundary for Windows Devices

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 a Windows-based thin client that only communicates with the Hypori Virtual Device. The Hypori Virtual Device, 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 is supported on Microsoft Windows 10 (64 bit), version 1809 (build 17763) and version 1903 (build 18362).

2.3.1.2 Hardware Requirements

The TOE imposes no hardware requirements beyond the Microsoft Windows 10 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 TOE uses cryptographic services provided by the Windows platform. The TOE stores credentials and certificates for mutual authentication in the Windows Credential Manager and the Windows Certificate Store. Certificates specific to an app are stored on a per user, per app Windows Certificate Store in an isolationist application model. The app has access to the "My" Certificate Store on the app level mentioned above, but also on the "Current User" Certificate Store.

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. There is some account provisioning information stored locally in the app's private data store (controlled by Windows). The data is located in a hidden directory with confidential data stored encrypted, which can only be unencrypted on that local device with that specific user logged in using Windows.Security.Cryptography.DataProtection. DataProtectionProvider class.

2.3.2.3 Identification and Authentication

The TOE uses the platform's certificate 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 – Windows, 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 Windows devices, as well as on desktop and server platforms. The Hypori Client is a Windows application running on a Windows 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 Windows devices, as well as on desktop and server platforms. Consequently, the [PP_APP_v1.3] security objectives are suitable for the Hypori Client TOE (Windows).

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
- 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

Requirement Class	Requirement Component
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,***
- ***Bluetooth,***
- ***[Graphics Capture,***
- ***Private Network usage,***
- ***Certificate Store Usage]***

] .

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 Microsofts's WNS (Windows Push Notification Services) 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 [

- **Microsoft .NET Framework v4.8.03752**
- **Universal Windows Apps v15.0.28307.1000**
- **Google Protobuf v2.5**
- **Microsoft.UI.XAML v2.3.191211002**
- **Microsoft.Toolkit.Uwp v5.1.1**
- **ZXing.Net v0.16.5**
- **Microsoft.ApplicationInsights v1.2.3**

[2] Modified per TD0561

[3] Modified per TD0561

- **Newtonsoft.Json v12.0.2**
- **TimeZoneConverter v3.2.0**
- **Opus-Windows v1.1.5**
- **MetroLog v1.0.1**
- **Microsoft.NETCore.UniversalPlatform v6.2.9**
- **libyuv-windows v1.0.1671**
- **ANGLE.WindowsStore v2.1.14**
- **Microsoft.Graphics.Canvas.dll**

].

5.2.5.6 Software Identification and Versions (FPT_IDV_EXT.1)

FPT_IDV_EXT.1.1 The application shall be versioned with [***[Microsoft's standards for packaging version numbering (Major, Minor, Maintenance Release, and a fourth number controlled by the Microsoft Store]***].

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 [***invoke 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, 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***].
- 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.[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 extendedKeyUsage field.

[4] The SFR was modified per TD0444.

[5] The SFR was 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.

- 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.[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 extendedKeyUsage field. [8]
- 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.[9]

FIA_X509_EXT.1.2 The application shall treat a certificate as a CA certificate only 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[10] 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

[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.

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

[10] The SFR was modified per TD0444.

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.

6. TOE Summary Specification

This chapter describes the security functions:

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

6.1 Cryptographic support

The Hypori Client makes use of the Windows 10 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 the RSA or Elliptic Curve key pairs 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 is evaluated on Microsoft Windows 10 (64 bit), v1809 and v1903 and relies on these platforms to provide all of its cryptographic functionality. These Common Criteria evaluated versions of Windows 10 are identified on the NIAP Product Compliant List as follows:

- "Windows 10 and Windows Server 2019 version 1809", with the ST available here: https://www.commoncriteriaportal.org/files/epfiles/2018-61-ST_lite.pdf
- "Microsoft Windows 10 and Server version 1903 (May 2019 Update)", with the ST available here: https://www.commoncriteriaportal.org/files/epfiles/2019-22-ST_lite.pdf

The Windows 10 platform provides the following TLS ciphersuites:

- TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246,
- 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_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289,
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289,
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,
- 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.

For elliptic curve cipher suites, the Hypori Client relies on the platform for elliptic curves. The Windows platform supports NIST curves secp256r1 and secp384r1 and Supported Elliptic Curves Extension for TLS. No configuration is required by a Hypori Client user.

6.1.1 FCS_CKM_EXT.1

The Hypori Client does not generate cryptographic keys. As part of installation, a user adds a TLS client certificate and the RSA or Elliptic Curve key pairs 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	The RSA/Elliptic Curve key pairs and the X509 certificate are used for TLS mutual authentication (client side of TLS exchange) that is implemented by the platform.	Windows Certificate Store
Server account password	Authenticates user to Hypori server	Windows Credential Manager

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 consists of user TLS client key and server account password credentials. FCS_STO_EXT.1 Storage of Secrets specifies the platform's Windows Certificate Store and Windows Credential Manager for protecting keys and credentials.

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 to respond to 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. The table shows the permissions required by the Hypori Client:

Table 4 Permissions Required by the Hypori Client

Permission	Description
Internet Connectivity	Open network sockets.

Permission	Description
Bluetooth	Connect to paired Bluetooth devices.
Graphics Capture	Enables the user to take screen captures when connected to the Virtual Device.
Location	Access precise location.
Microphone	Provides access to the microphone and audio recording capabilities on the Windows device to support apps in the Virtual Device that require audio input.
Private Network Usage	Used to access Intranet networks that have an authenticated domain controller, or that the user has designated as either home or work networks.
Certificate Store Usage	Used to store the User TLS client key.
Camera	The Hypori Client uses remote access to the device's camera to support multimedia applications that use the camera in the Virtual Device. It can also use the camera when scanning a QR code during account provisioning.
Background Tasks (Notifications)	The background operations permission is required to receive notifications when the application is not active.

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 Windows 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 TOE does not access any sensitive information repositories as defined by the [PP_APP_v1.3].

Accounts are stored using Microsoft's preferred application settings method, Windows.Storage.ApplicationData.Current.LocalSettings class.

https://docs.microsoft.com/en-us/windows/uwp/design/app-settings/store-and-retrieve-app-data

This data is stored on the hard drive in isolated storage (i.e. its by user and by app). Only users with admin rights to the device may get to this data, or the UWP app itself. UWP apps, defined by system rules, are sandboxed, and can only affect app specific data for the current windows user.

The Hypori Client does not maintain PII. Hence, it does not transmit PII over any network.[11]

6.3 Identification and authentication

The Windows platform follows RFC 5280 for certificate path validation. The Hypori Client uses Windows certificate validation services to authenticate the X.509 certificate the Hypori server presents as part of the establishing a TLS connection.

[11] 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.

6.3.1 FIA_X509_EXT.1

The TOE uses X.509v3 certificates as defined by RFC 5280 to support authentication for TLS connections. All certificate validation is performed by the underlying Windows platform, and certificates are stored in the Windows Certificate Store. Certificate validation is done in conformance to RFC 5280. Certificate validation paths must terminate with a trusted CA certificate that contains the basicConstraints extension and a CA flag that is set to TRUE. ExtendedkeyUsage field validation is also performed. Checking is also done for the basicConstraints extension and the cA flag to determine whether they are present and set to TRUE. If they are not, the certificate is not accepted. The Windows 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.

Certificates must have a valid and established chain of trust by verifying the root certificate and are verified using the Windows.Security.Cryptography.Certificates Namespace.

https://docs.microsoft.com/en-us/uwp/api/windows.security.cryptography.certificates?view=winrt-19041

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

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 the `Certificate setting` on the `Connection 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 Windows Certificate Store during installation. (The user need not install a CA certificate when the CA is a platform trusted CA.)

On Windows devices, the Hypori Client uses Windows platform certificate path validation services with the CA certificate to validate the certificate presented by the Hypori server. The Windows platform will send a status request to an OCSP responder and receive information if the certificate is valid or revoked. A good response will indicate the certificate is valid and not revoked. A revoked status will indicate the certificate has been revoked. If the OCSP responder 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 the Windows 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 binaries are stored in a protected location under %ProgramFiles%\WindowsApps.

6.4.2 FMT_MEC_EXT.1

The Hypori Client invokes the recommended Windows mechanisms for storing account settings files. Accounts are stored using Microsoft's preferred application settings method, Windows.Storage.ApplicationData.Current.LocalSettings class. https://docs.microsoft.com/en-us/windows/uwp/design/app-settings/store-and-retrieve-app-data

The namespace used is Windows.Storage. The primary class to manage this data is the ApplicationData.Current.LocalSettings. This class will persist settings (LocalSettings is a dictionary like mechanism using key/value pairs) to a system managed folder created at the time the application is installed. This same data will be removed when the application is uninstalled. The system is responsible for maintaining the physical storage and insures that the data is isolated from other apps and users. The system preserves this data when the app is updated.

This data is stored on the hard drive in isolated storage (i.e. it's by user and by app). Only users with admin rights to the device may get to this data, or the UWP app itself. UWP apps, defined by system rules, are sandboxed, and can only affect app specific data for the current windows user.

The binaries are stored in a protected location under %ProgramFiles%\WindowsApps.

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

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 Windows .NET UWP applications, and stack-based buffer overflow protection. The client leverages Windows UWP 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

The Hypori client handles ASLR in its C++ libraries by setting the linker option "/DYNAMICBASE". The C# code is enabled by default, so the client code is ASLR compliant. The application does not allocate any memory region with both write and execute permissions nor does the TOE request to map memory to an explicit address. The TOE does not write user-modifiable files to directories that contain executable files. The application is built with stack-based buffer overflow protection enabled.

The TOE is compatible with security features provided by the platform vendor. The TOE OS platform supports Windows Defender Exploit Guard.

6.6.2 FPT_API_EXT.1

The Hypori Client uses the Windows APIs listed in section 9 Appendix: Windows APIs.

6.6.3 FPT_IDV_EXT.1

The Hypori Client Windows application uses the major.minor.maintenance release format, with the exception that the Microsoft Store reserves a last integer-based number for itself (major.minor.maintenance.internalwindowsuseonly) (e.g. 4.2.0.0). For Windows 10 (UWP) packages, the last (fourth) section of the version number is reserved for Microsoft Store use and must be left as 0 when the package is built, although the Microsoft Store may change the

value in this section. A description of Microsoft's standards for package version numbering by the Microsoft Store is available at:

https://docs.microsoft.com/en-us/windows/uwp/publish/package-version-numbering?redirectedfrom=MSDN.

The Hypori Client Windows application will display the TOE version in the major.minor.maintenance release format in the lower right corner of the display window.

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 Windows standard .appx file for Windows devices.

The TOE relies on the Microsoft Store to provide application updates. Updates are automatically handled by the Windows Operating System, so notifications will be given to the user about existing application updates. Hypori digitally signs the installation package as well as updates and includes the corresponding public key certificate in the package. Windows 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 Microsoft Store or the enterprise IT group of the user.

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 Windows platform.

The Hypori Client will open the socket, designate the host, port, and socket protection level (TLS 1.2). The following Microsoft platform API invokes this functionality:

```
await _socket.ConnectAsync(new_HostName(_serverAddress), _serverPort.ToString(), SocketProtectionLevel.Tls12).
```

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 Microsoft 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 Support 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 TLS 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 5 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 6 demonstrates the relationship between security requirements and security functions.

Table 6 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.2						X	
FTP_DIT_EXT.1							X

9. Appendix: Windows APIs

The Hypori Client uses the following Windows APIs:

1. Internal.Runtime.CompilerServices
2. internal::down_cast
3. internal::implicit_cast
4. internal::Mutex
5. internal::MutexLock
6. internal::MutexLockMaybe
7. internal::ReaderMutexLock
8. internal::scoped_array
9. internal::scoped_ptr
10. internal::WriterMutexLock
11. Mcg.System
12. MetroLog.Targets
13. MetroLog
14. Microsoft.Build.Framework
15. Microsoft.Build.Utilities
16. Microsoft.Graphics.Canvas.Brushes
17. Microsoft.Graphics.Canvas.UI.Xaml
18. Microsoft.Graphics.Canvas.UI
19. Microsoft.Graphics.Canvas
20. Microsoft.Toolkit.Uwp.Connectivity
21. Microsoft.VisualStudio.TestPlatform.UnitTestFramework
22. Newtonsoft.Json.Converters
23. Newtonsoft.Json.Linq
24. Newtonsoft.Json
25. OrientationEventRpc
26. Platform
27. std::ifstream
28. std::ios
29. std::string
30. std::stringstream
31. std::uint64_t
32. std::unique_ptr
33. std::vector
34. System.Collections.Concurrent
35. System.Collections
36. System.ComponentModel
37. System.Diagnostics
38. System.Drawing.Color
39. System.Globalization
40. System.IO.Compression
41. System.IO
42. System.Linq

43. System.Net.Http
44. System.Net.Security
45. System.Net.Sockets
46. System.Net
47. System.Reflection
48. System.Resources
49. System.Runtime.CompilerServices
50. System.Runtime.InteropServices.WindowsRuntime
51. System.Runtime.InteropServices
52. System.Runtime.Serialization.Formatters
53. System.Runtime.Serialization.Json
54. System.Runtime.Serialization
55. System.Security.Authentication
56. System.Security.Cryptography.X509Certificates
57. System.Security.Cryptography
58. System.Text.RegularExpressions
59. System.Text
60. System.Threading.Tasks
61. System.Threading
62. System.Timers
63. System.Windows.Input
64. System.Xml
65. System
66. TimeZoneConverter
67. Windows.ApplicationModel.Activation
68. Windows.ApplicationModel.Background
69. Windows.ApplicationModel.Core
70. Windows.ApplicationModel.DataTransfer
71. Windows.ApplicationModel.Resources
72. Windows.ApplicationModel
73. Windows.Data.Json
74. Windows.Devices.Geolocation
75. Windows.Devices.Input
76. Windows.Devices.Sensors
77. Windows.Devices.SmartCards
78. Windows.Devices.WiFi
79. Windows.Foundation.Collections
80. Windows.Foundation.Diagnostics
81. Windows.Foundation
82. Windows.Graphics.Display
83. Windows.Media.Audio
84. Windows.Media.Capture.Frames
85. Windows.Media.Capture
86. Windows.Media.Devices

87. Windows.Media.MediaProperties
88. Windows.Media.Render
89. Windows.Media
90. Windows.Networking.Connectivity
91. Windows.Networking.PushNotifications
92. Windows.Networking.Sockets
93. Windows.Networking
94. Windows.Security.Credentials
95. Windows.Security.Cryptography.Certificates
96. Windows.Security.Cryptography.Core
97. Windows.Security.Cryptography.DataProtection
98. Windows.Security.Cryptography
99. Windows.Security.ExchangeActiveSyncProvisioning
100. Windows.Storage.Pickers
101. Windows.Storage.Streams
102. Windows.Storage
103. Windows.System.Power
104. Windows.System.Profile
105. Windows.System.Threading
106. Windows.System
107. Windows.UI.Color
108. Windows.UI.Core
109. Windows.UI.Input
110. Windows.UI.Notifications
111. Windows.UI.Popups
112. Windows.UI.Text.Core
113. Windows.UI.ViewManagement
114. Windows.UI.Xaml.Controls.Primitives
115. Windows.UI.Xaml.Controls
116. Windows.UI.Xaml.Data
117. Windows.UI.Xaml.Input
118. Windows.UI.Xaml.Media.Imaging
119. Windows.UI.Xaml.Media
120. Windows.UI.Xaml.Navigation
121. Windows.UI.Xaml
122. Windows.Web.Http.Headers
123. Windows.Web.Http
124. Windows::Foundation::Collections
125. Windows::Foundation
126. Windows::System::Diagnostics
127. Windows::UI::Xaml::Controls
128. ZXing