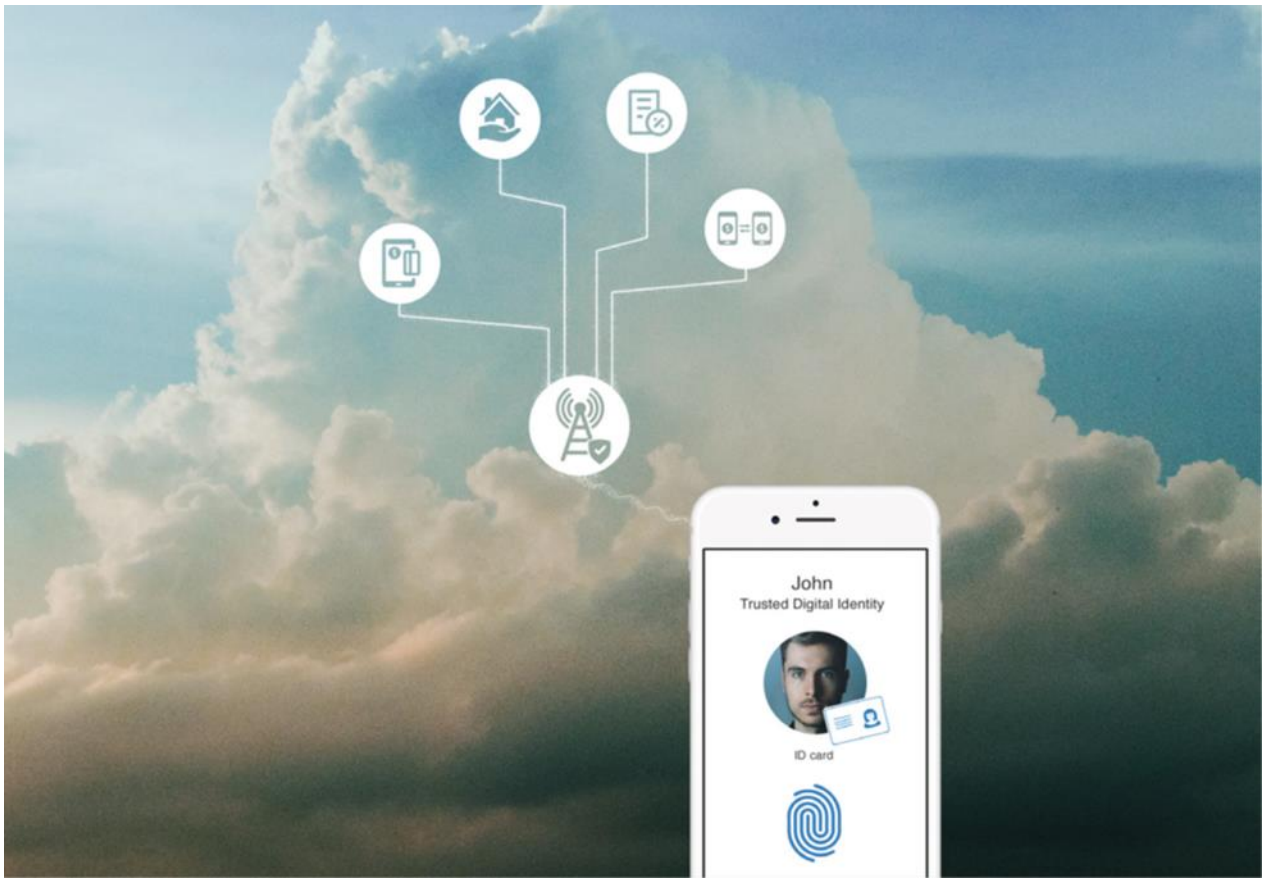
Security Target for
Gemalto Advanced Whitebox PKI SDK
(Public version)



CONTENT

FIGURES

TABLES

1. ST INTRODUCTION

1.1 SECURITY TARGET REFERENCE

Title :	Security Target for Gemalto Advanced Whitebox PKI SDK
Version :	1.7p (Public version extracted from evaluated version)
ST Reference :	R0R28657_AWPKI_SDK_ST
CC Version:	3.1 Revision 5
Assurance Level:	EAL 3 Augmented with AVA_VAN.3, ADV_FSP.4, ADV_TDS.3, ADV_IMP.1, ALC_TAT.1
General Status:	Evaluated
Origin :	Gemalto a Thales company
Author :	Francois GUERIN
ITSEF:	BRIGHTSIGHT
Certification Body :	NSCIB operated by NLNCSA
Certification ID	NSCIB-CC-230855

1.2 TOE REFERENCE

Product Name :	Gemalto Mobile ID Smart SDK
TOE Name :	Gemalto Advanced Whitebox PKI SDK for Android
TOE Version :	v1.0.1.300* for Android
TOE documentation:	Guidance [GUIDE_AWPKISDK]

TOE versioning is defined as following: V W.X.Y.Z where W.X.Y. represents a numeric value between 0 and 99 and Z a numeric value between 000 and 999.
Major version W: Starts from 1, incremented whenever a major change is implemented
Minor version X: Starts from 0, incremented whenever features are added/changed
Micro version Y: Starts from 0, incremented for bug fixes/minor patches
Build number Z: Starts from 001, set automatically incremented for each Jenkins build number

The Product and TOE identification details are provided in §TOE description.

1.3 SECURITY TARGET DOCUMENT OVERVIEW

The current Security Target document describes the TOE and its environment and the scope of the evaluation refining security objectives for TOE and its environment and TOE security features under evaluation.

The main objectives of this ST are:

- To introduce TOE and the relevant environment,
- To define the scope of the TOE and its security features,
- To describe the security environment of the TOE, including the assets to be protected and the threats to be countered by the TOE and its environment during the product development, production and usage.
- To describe the security objectives of the TOE and its environment.
- To specify the security requirements which includes the TOE security functional requirements, the TOE assurance requirements and TOE security functions.

The Target of Evaluation (TOE) addressed by the current Security Target is a software development kit named Gemalto Mobile ID Smart SDK including items depicted in following figure. The kit includes AWPKI SDK library to be embedded in any customer application working in conjunction with Gemalto server named AWPKI Server. It also includes a development guidance explaining rules for integration of library in customer application. A Gemalto sample application is also provided to help customer for integration.

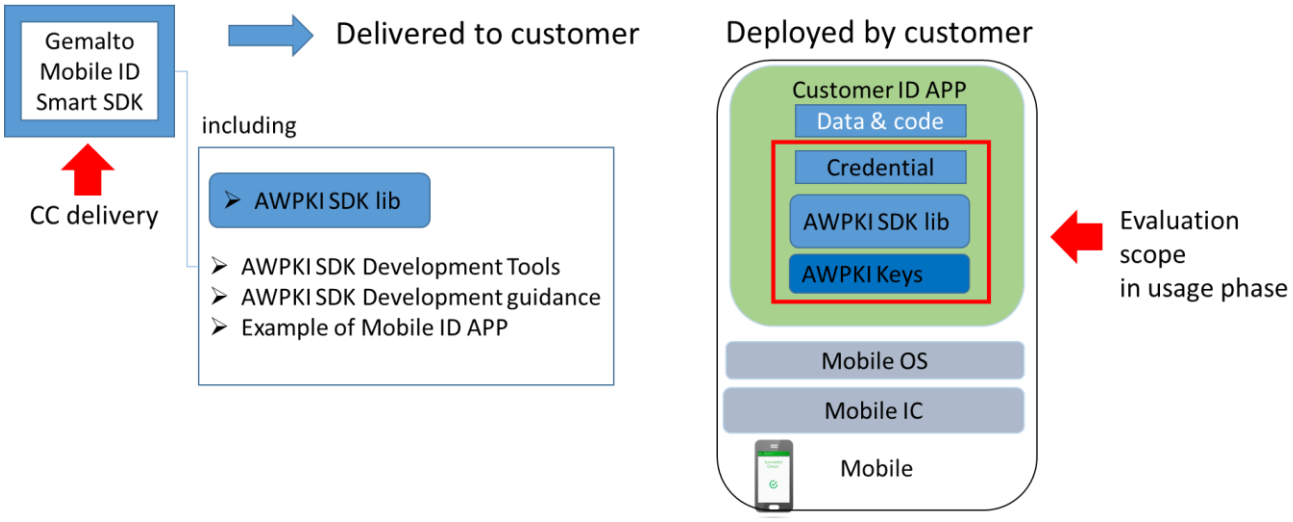


Figure 1: Gemalto Mobile ID Smart SDK as delivered to customer and AWPKI SDK as integrated in customer application

The key security feature is a set of operations implemented by a "state machine" (described using a proprietary representation in next figure) that runs cryptographic operations (ECDSA signature) using information from server, mobile and user and a key provisioned from server. Result of cryptographic operations is sent back to the server for verification. When verification is confirmed, user is considered as authenticated.

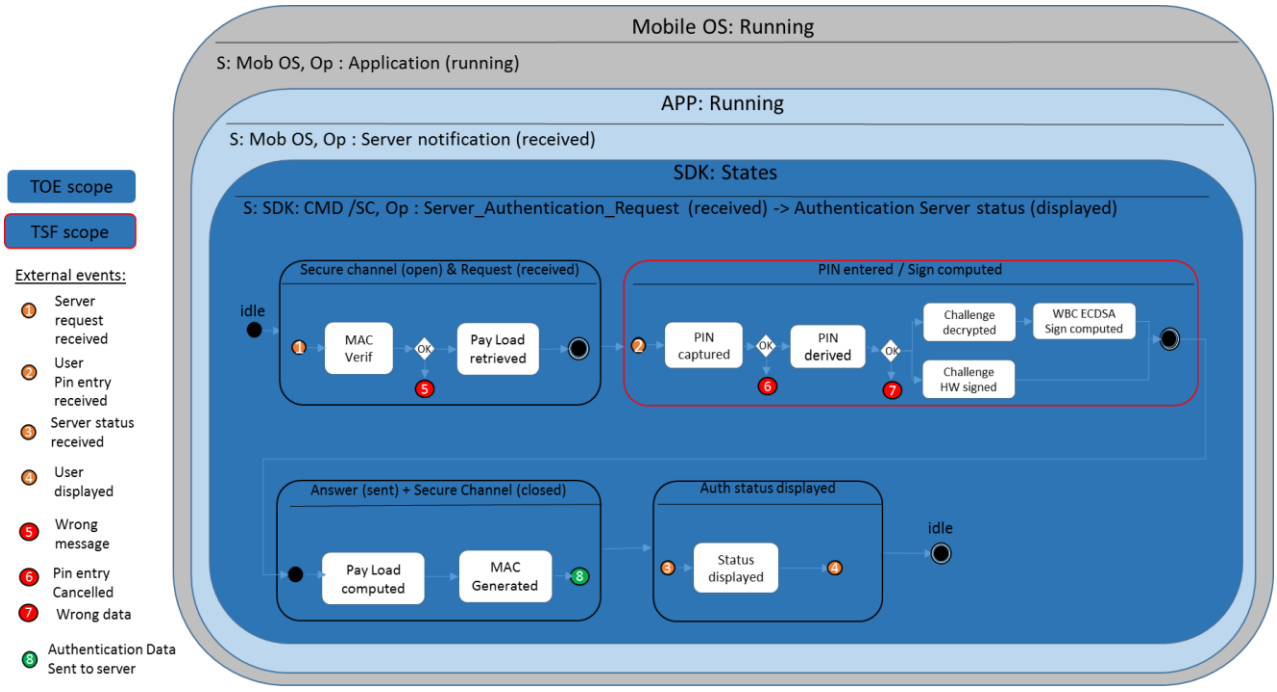


Figure 2: State Machine implementing Signature computation for User authentication by AWPKI Server

1.4 REFERENCES

1.4.1 External References

[CC-1]	Common Criteria for Information Technology Security Evaluation Part 1: Introduction and general model, CCMB-2017-04-001, version 3.1 rev 5, April 2017
[CC-2]	Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, CCMB-2017-04-002, version 3.1 rev 5, April 2017
[CC-3]	Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, CCMB-2017-04-003, version 3.1 rev 5, April 2017
[CEM]	Common Methodology for Information Technology Security Evaluation Methodology CCMB-2017-04-004, version 3.1 rev 5, April 2017
[FIPS186]	Federal Information Processing Standards Publication FIPS PUB 186-4, Digital Signature Standard (DSS), 2013-07
[FIPS197]	Federal Information Processing Standards Publication 197, Advanced Encryption Standard (AES), U.S. Department of Commerce/National Institute of Standards and Technology, 2001-11-26
[IETF 2104]	Internet Engineering Task Force (IETF) 2104, HMAC: Keyed-Hashing for Message Authentication.
[ISO15946-1]	ISO/IEC 15946: Information technology – Security techniques – Cryptographic techniques based on elliptic curves – Part 1: General, 2002
[ISO15946-2]	ISO/IEC 15946: Information technology – Security techniques – Cryptographic techniques based on elliptic curves – Part 2: Digital Signatures, 2002
[ISO9797-1]	ISO/IEC 9797: Information technology – Security techniques – Message Authentication Codes (MACs) – Part 1: Mechanisms using a block cipher, 1999
[NIST800-132]	NIST Special Publication 800-132 - Recommendation for Password-Based Key Derivation (December 2010)
[RFC2898]	PKCS #5: Password-Based Cryptography Specification Version 2.0 https://www.ietf.org/rfc/rfc2898.txt
[SP800-38A]	Recommendation for Block Cipher Modes of Operation: Methods and Techniques, NIST, Special Publication 800-38A, National Institute of Standards and Technology, December 2001
[SP800-38B]	Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication, NIST Special Publication 800-38B, National Institute of Standards and Technology, May 2005

1.4.2 Internal References

Reference ID	Description
[GUIDE_AWPKISDK]	[AGD_OPE_AWPKISDK] + [AGD_PRE_AWPKISDK]
[AGD_DEV_AWPKISDK]	Advanced Whitebox PKI Library Developer Guide, Version 1.13 (Proprietary only available under NDA)
[AGD_OPE_AWPKISDK]	[AGD_US_AWPKISDK] + [SDEV_GUIDE]
[AGD_PRE_AWPKISDK]	[PRE]+ [SEC_SA] + [AGD_DEV_AWPKISDK] + [SEC_SI] + [SDEV_GUIDE]
[AGD_US_AWPKISDK]	Security Rules for User of Application including Gemalto Advanced Whitebox PKI SDK V1.0.1, Version 1.5 (Proprietary only available under NDA)
[PRE]	Preparation Guidance for Gemalto Advance Whitebox PKI SDK V1.0.1, Version 1.5 (Proprietary only available under NDA)
[SEC_SA]	Security Rules for Application development based on Gemalto Advanced Whitebox PKI SDK V1.0.1, Version 1.6 (Proprietary only available under NDA)
[SEC_SI]	Security Rules for Server development & integration associated to Gemalto Advanced Whitebox PKI SDK V1.0.1, V1.2 (Proprietary only available under NDA)
[SDEV_GUIDE]	AWPKI Server Integration guide, V1.2 (Proprietary only available under NDA)
[DEP]	Gemalto Data Exchange Protocol (Proprietary only available under NDA)
[AESP]	The Design of Rijndael AES — The Advanced Encryption Standard From Joan Daemen, Vincent Rijmen November 26, 2001

1.5 ACRONYMS AND GLOSSARY

Acr.	Term	Definition
	ACO	Authentication Cryptographic Operation
	ACOR	Authentication Cryptographic Operation Result
	AR	Authentication Request
	CC	Common Criteria for IT Security Evaluation
	CEM	Common Methodology for Information Technology Security Evaluation
	Customer	Customer is the entity integrating SDK in Customer application and integrating AWPKI Server in the IT infrastructure
	DERIVED RAD	TOE uses a derivation function to compute Derived RAD from RAD provided by user. Derived RAD is transmitted to server for user authentication purpose.
	DERIVED VAD	TOE uses a derivation function to compute Derived VAD from VAD provided by user. Derived VAD is used as input for ACO.
	EAL	Evaluation Assurance Level
	ECC	Elliptic Curve Cryptography
	ENC	Encryption
	ECDSA	Elliptic Curve Digital Signature Algorithm
	ENC	Content Data Encryption
	IT	Information Technology
	MRA	Mutual Recognition Agreement
	NIST	National Institute of Standards and Technology
	PIN	Personal Identification Number (refer to Verification Authentication Data)
	PKI	Public Key Infrastructure
	RAD	Reference Authentication Data is provided during user enrolment by user, derived by TOE, transferred and stored in server once to contribute to user authentication. Note: RAD is not stored in TOE.
	SAR	Security Assurance Requirement
	SHA	Secure Hash Algorithm
	TOE	Target Of Evaluation (CC part 1 [CC-1]).
	TSF	TOE Security Functionality (CC part 1 [CC-1]).
	TSF data	Data created by and for the TOE that might affect the operation of the TOE ([CC-1]).
	TTBD	Text to be Displayed to the user on mobile screen
	User, external entity	Human or IT entity possibly interacting with the TOE from outside of the TOE boundary.
	VAD	Verification Authentication Data is data entered by user and derived by TOE to perform user authentication (e.g PIN, Password). Note: VAD is not stored in TOE.
	User data	Data created by and for the user that does not affect the operation of the [CC-1]).
	WBC	White Box Cryptography

1.6 TOE OVERVIEW

1.6.1 System Architecture

The following figure describes a system architecture as deployed by the service provider.

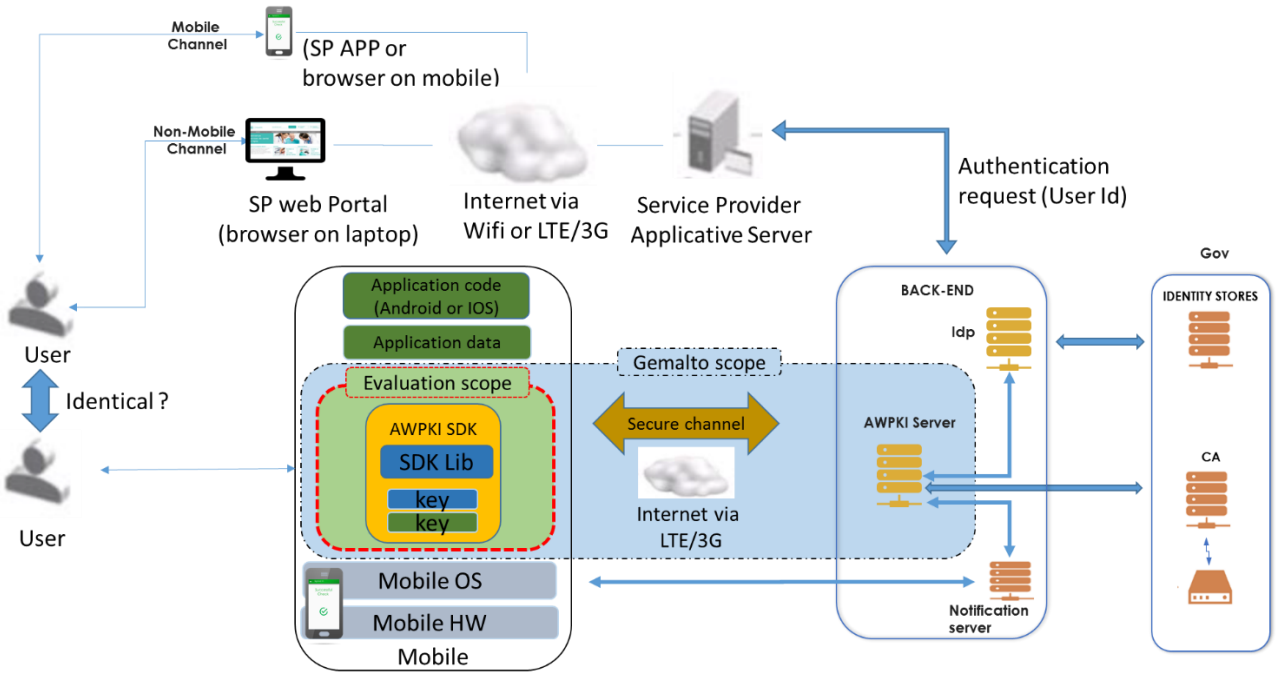


Figure 3: System architecture including TOE inside customer application

Gemalto provides items in the deployed solution:

- a SDK (also named AWPKI SDK) to be integrated in the service provider Mobile application,
- a AWPKI Server software deployed on AWPKI Server in the service provider infrastructure.

Both items work together to provide user authentication service to service provider.

1.6.2 TOE description

AWPKI SDK is delivered to be integrated by Customer in a mobile application (named here after Customer APP). This application is developed by Customer using AWPKI SDK development tools according rules defined in TOE development guidance and in line with Gemalto application example provided as AWPKI SDK APP. AWPKI SDK APP is delivered also for the evaluation purpose because **AWPKI SDK** cannot be run alone in the mobile environment.

1.6.2.1 Product and TOE identification

For Product and TOE identification, refer to §1.2

The Product and TOE identification and configuration are provided by executing a dedicated command described in [AGD_US_AWPKISDK], §1.5).

The TOE and the product differ, as further explained in §1.7 TOE Boundaries:

- The TOE is the **AWPKI SDK.**
- The product is a Customer application running on mobile thanks to mobile and mobile OS service (Android). Some others applications are also executed on the mobile. Customer application including **AWPKI SDK** dealing with Gemalto AWPKI Server to provide user authentication service.

TOE can be identified using information from getAWPKIName() method providing unique identifier for SDK and getVersion() method to obtain version of SDK (see table 1):

SDK name	Version
”Gemalto Advanced Whitebox PKI SDK for Android”	”1.0.1.300”

Table 1: TOE Identification Data

1.6.3 TOE Usage and Security Features in Operational Use

1.6.3.1 Main functionalities of the TOE

As described on figure 2, the main security functionality of the TOE is to contribute to an authentication request launched by service provider applicative server received from AWPKI Server.

TOE provides the client part of the Authentication Request Protocol (ARP) that assures the control of the authentication request.

TOE performs (on AWPKI server request) an authentication cryptographic operation (ECDSA signature) using information from AWPKI Server, user mobile, and user. When operation is completed, authentication cryptographic operation result is transferred securely to AWPKI server to finalize the authentication request. The TOE also provides administration features to manage sensitive data.

The main functionalities of the TOE not in the evaluation scope are:

- Initialization and configuration of the SDK (sensitive data for Server authentication),
- User enrolment (User registration with sensitive data transfer to server and ECDSA keys provisioning from server to OS file system),
- Storage of ECDSA keys in OS file system,
- Data signing after user consent,
- Administration features (user secret update, ECDSA key renewal),
- User de registration,
- Connectivity to server,
- Secure channel between SDK and server using DEP protocol defined in [DEP].

1.6.3.2 Overview of Authentication Request Protocol client part

A standard management of authentication request is limited to receive a request from a server, asks authentication data to the user, performs operation to check validity of authentication data and send back a result to server. Such way to proceed may not resist to attacker with Enhanced-Basic attack potential.
Therefore, Gemalto has setup an authentication request protocol (ARP) involving AWPKI SDK and AWPKI Server to counter potential threats identified on standard management of authentication request as listed below.

The following lines describes the client part of the authentication request protocol (ARP):
Note that D.XX are described in §3.3 Assets.

1) AWPKI Server sends a notification for an authentication request (D.AR) to referenced application (APP) loaded and activated on a registered mobile (MOB) already associated to a registered User (User).
Therefore AWPKI Server already knows D.MOB_ID, D.USER_ID, D.APP_ID.

2) Mobile OS receives notification and transfers it to the accurate APP for an authentication request (D.AR) from AWPKI Server using D.APP_ID and D.MOB_ID linked to D.USER_ID and APP transfers the request to the AWPKI SDK.

3) AWPKI SDK setups a secure channel between SDK and Server
In case of error detection in secure channel establishment, request is cancelled and an error message is sent back to the APP and 11),
Otherwise, ARP operation continues.

4) AWPKI SDK receives AR from Server including D.Serv_Auth_Data (Challenge, DTBS, TTBD,…)
In case of error detection in data received, request is cancelled an error message is sent back to the Server and 11)
Otherwise, ARP operation continues.

5) AWPKI SDK asks User consent (request of credential) to perform AR using D.Serv_Auth_Data and obtains D.User_Auth_Data (PIN, Fingerprint*) from User. (*: Note Fingerprint is mentioned for future use)
In case of user denial, request is cancelled an error message is sent back to the AWPKI Server and 10) & 11)
Otherwise, ARP operation continues.

6) AWPKI SDK retrieves D.Mob_Auth_Data (data from Key store) from Mobile
In case of data are not available, request is cancelled an error message is sent back to the Server and 10) & 11)
Otherwise, ARP operation continues.

7) AWPKI SDK performs operation to prepare ACO and authorizes ACO when conditions are fulfilled (see details in FDP_ACF.1.2/AUTH-Comp from §6.2.2).
In case of conditions are not fulfilled, request is cancelled an error message is sent back to the AWPKI Server and 10) & 11)
Otherwise, ARP operation continues.

8) AWPKI SDK computes ACO and manages ACOR.
In case of ACO interruption, request is cancelled an error message is sent back to the Server and 10) & 11)
ARP operation continues.

9) AWPKI SDK send back ACOR to server using secure channel in order that AWPKI Server can finalize AR.
In case of ACOR not received, request is cancelled an error message is sent back to the Server

10) AWPKI SDK closes the secure channel and clean residual data from AR.

11) AWPKI SDK informs APP about AR completion (or not) of transfer of AR to server (APP is responsible to inform user)

1.7 TOE BOUNDARIES

TOE logical and physical boundaries are defined on the following figures.

TOE includes:

- The AWPKI SDK,
- The associated guidance documentation.

Note:
Secrets (ECDSA keys) are generated by Gemalto using a proprietary diversification and provisioned securely in a self-protected format during enrolment phase into the mobile.

Secrets (AESP key) are generated by Gemalto using a proprietary diversification, delivered in a self-protected format and stored securely in application and AWPKI server during development phase.

Verification authentication data (e.g PIN, password) is entered during enrolment phase by user and exported securely to AWPKI Server using a Gemalto proprietary method. Then Verification authentication data is

entered by user and used for authentication cryptographic operation done by AWPKI SDK but it is not stored locally.

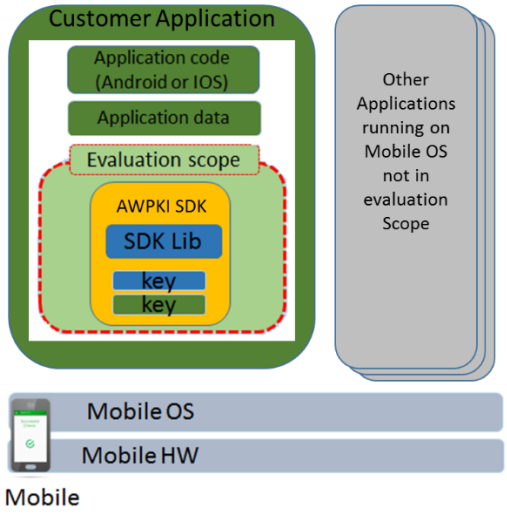


Figure 4: TOE Logical Boundaries

Physical boundaries are defined by AWPKI SDK code and data location in mobile volatile and non-volatile memory as part of Customer application and file system part located in Mobile OS non-volatile memory.

The main cryptographic functionalities provided by TOE are:

- ECDSA signature defined in [FIPS186], used for authentication service,
- Protected AES decryption defined in [AESP], used for authentication service.

Some additional cryptographic services are used internally but are not in the scope of the TSF.

The TSF is the state machine that prepares and runs a cryptographic operation (ECDSA signature) using inputs from server, mobile and user and a key in a self-protected format. The result of this computation is transferred to server for verification.

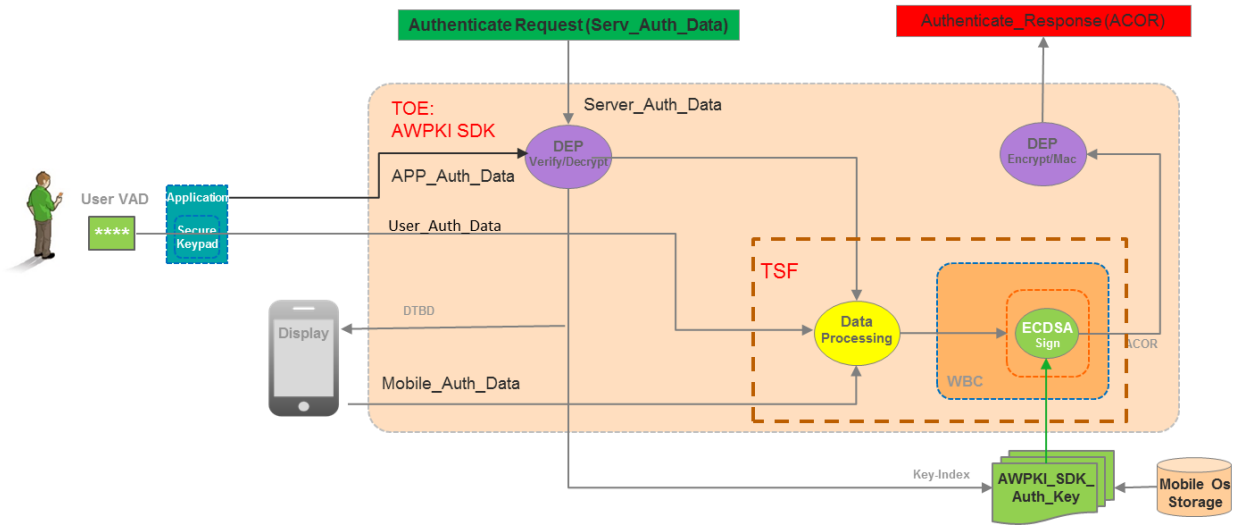


Figure 5: TOE and TSF scope

The result of this verification (out of TOE scope) allows the server to determine if signature is performed using the right input, key and cryptographic engine. In such case, the user is considered as authenticated.

1.8 AWPKI SDK NON-TOE HARDWARE/SOFTWARE/FIRMWARE

AWPKI SDK is included in Customer application.

Customer application in loaded, installed and stored in mobile non-volatile memory using mobile OS interface. It is executed thanks to mobile providing a hardware-backed keystore and Mobile OS services running Android 6.0 or above.

AWPKI Server initiates interaction with Secrets (e.g. ECDSA keys) used in AWPKI SDK are stored in proprietary format in Mobile OS file system located in mobile non-volatile memory.

There is no shared interface between AWPKI SDK and any other application than Customer application.

User interacts with mobile and mobile OS interface through screen to display information and virtual keyboard to enter information required by AWPKI SDK.

AWPKI SDK interacts with Gemalto AWPKI Server to perform operations required in authentication request.

Server is responsible to provide accurate inputs for authentication request, and it is responsible to verify the validity of answer provided by AWPKI SDK for authentication request. If valid, Server uses outputs from AWPKI SDK to complete the authentication service for Service Provider.

1.9 TOE LIFE-CYCLE

The TOE life cycle model can be described as following.

1.9.1 The phases prior the TOE delivery to Customer Application Integrator

The TOE life cycle is described with the following phases:

Phase 1 and 2 “TOE development”:
The AWPKI SDK developer is responsible of TOE development in phase 1 (in TOE scope). The AWPKI SDK developer develops the AWPKI SDK, the AWPKI SDK sample application and the AWPKI SDK guidance documentation.

Phase 2 “AWPKI Server software development”:

The AWPKI Server developer is responsible of AWPKI Server software working in conjunction with AWPKI SDK in phase 2 (not in TOE scope).
The AWPKI Server developer uses the guidance documentation for the AWPKI SDK and the APP sample application to integrate AWPKI Server software with AWPKI Server software (not in TOE scope).

Phase 3 “Packaging and Testing”:
AWPKI Server developer provides an instance of AWPKI Server software to AWPKI SDK developer to perform end to end validation (in TOE scope).

AWPKI SDK developer is responsible to provide TOE to Delivery Center for delivery to Customer Application Integrator (in TOE scope).

AWPKI Server developer is responsible to provide AWPKI Server software to Delivery Center for delivery to Customer Application Integrator (not in TOE scope).

Delivery Center is responsible to make Testing and Packaging of the AWPKI SDK and AWPKI Server Software and to prepare delivery to Customer Application Integrator (in TOE scope).

Phase 4: “Delivery”

Delivery Center is responsible to make TOE delivery to Customer Application Integrator (in TOE scope).

Customer Application Integrator is responsible to verify accuracy of items delivered by Delivery Center (in TOE scope).

Phase 5: “Customer Application Development, Integration and Delivery”

Customer Application Integrator is responsible to integrate AWPKI Server software in its Customer server software (not in TOE scope).

Customer Application Integrator is responsible to use AWPKI SDK to integrate AWPKI SDK in customer application. For such purpose, it can use the AWPKI SDK development guidance and AWPKI SDK sample application.
When ready, Customer Application is pushed on accurate store to be deployed on user mobile (not in TOE scope).

Phase 6: “Customer Application Initialization”
Customer Application Integrator is responsible to AWPKI SDK initialization guidance to perform AWPKI SDK initialisation.

Phase 7: “Customer Application usage and Administration”

Customer Application Administrator is responsible for Customer Application administration according to accurate guidance and AWPKI SDK administration according to AWPKI SDK administration guidance.

User is responsible to use Customer Application according to accurate guidance and AWPKI SDK according to AWPKI SDK usage guidance.

1.9.2 Actors

Actors	Identification	Scope
AWPKI SDK developer	Gemalto	In scope
AWPKI Server developer	Gemalto	Out scope
Delivery Center	Gemalto	In scope
Customer Application Integrator	The agent who integrates AWPKI SDK in Customer Application and AWPKI Server in Customer server software.	In scope
Customer Application Administrator	The agent who performs Customer Application and AWPKI SDK initialisation and administration.	Out of scope
Customer Application Server Administrator	The agent who performs Customer server initialisation and administration.	Out of scope
User of Authentication service	User of Authentication service using Mobile where Customer Application and AWPKI SDK are running.	In scope

Table 2: Identification of the actors

1.9.3 Sites used prior TOE Delivery

The lifecycle is described in following table.

Life cycle phase	Gemalto Involved sites
TOE development (Phase 1)	**Gemalto Singapore** site (MSG team, ZFS team and R&D team) **Gemalto Meudon** site (ZFS team) **Gemalto La Ciotat** site (SCS team)
AWPKI Server software development (Phase 2)	**Gemalto Praha** (Not in scope)
Packaging and Testing (Phase 3)	**Gemalto Praha** (Delivery Center team)
Delivery (Phase 4)	Delivery to Customer Application Integrator from **Gemalto Praha** (Delivery Center team)

Table 3: Sites used prior TOE delivery

2. CONFORMANCE CLAIMS

2.1 CC CONFORMANCE CLAIM

This security target claims conformance to

- [CC-1]
- [CC-2]
- [CC-3]

as follows

- Part 2 extended,
- Part 3 conformant.

The [CEM] has to be taken into account.

2.2 PP CLAIM

This security target claims no conformance to any protection profile.

2.3 PACKAGE CLAIM

This ST is conforming to assurance package EAL3 augmented with AVA_VAN.3, ADV_FSP.4, ADV_TDS.3, ADV_IMP.1, ALC_TAT.1 as defined in CC part 3 [CC-3].

2.4 PP CONFORMANCE CLAIM RATIONALE

Not applicable.

3. SECURITY PROBLEM DEFINITION

3.1 GENERAL

The subjects, assets, threats, OSP, and assumptions of the TOE are defined in following paragraphs.

3.2 SUBJECTS AND EXTERNAL ENTITIES

This ST considers the following external entities (EE) and subjects (S):

External Entity / Subject	Role	Definition
S.User_NotAuthenticated	User	User not authenticated and interacting with AWPKI SDK included in APP running on user mobile.
S.User_Authenticated	User	User being authenticated by AWPKI Server and AWPKI SDK included in APP running on user mobile.
S.AWPKI_Server	Authentication Server	AWPKI Server is an abstract role to manage authentication request to be performed by TSF.
S.Administrator	Administrator	Administrator is an abstract role for an authenticated user on server side with privilege able to manage administration operations to be performed by TSF.
S.Command_Manager	TOE Administrator (including TSF part managing commands)	Command_Manager is an abstract role to describe management of command by TSF.
EE: User mobile & mobile OS	Resource provider to AWPKI SDK	Application including AWPKI SDK is running on User mobile registered in AWPKI server, using support of mobile OS.
EE: Untrusted mobile Application S.External_world	Different application running on user mobile	Untrusted mobile application sharing or not resources of mobile OS with application
EE: Untrusted Software external to TOE S.External_world	software trying to deal with mobile application including SDK	Untrusted external software trying to deal with mobile application including SDK

Table 4: External Entities and Subjects

This table defines external entities and subjects in the sense of [CC1].

Subjects can be recognized by the TOE independent of their nature (human or technical user). As result of an appropriate identification and authentication process, the TOE creates – for each of the respective external entity – an ‘image’ inside and ‘works’ then with this TOE internal image (also called subject in [CC1]). From this point of view, the TOE itself perceives only ‘subjects’ and, for them, does not differ between ‘subjects’ and ‘external entities’. There is no dedicated subject with the role ‘attacker’ within the current security policy, whereby an attacker might ‘capture’ any subject role recognized by the TOE.

3.3 ASSETS

The next tables focus on the assets that are relevant for the TOE with a distinction related to their need for protection in view of confidentiality (Conf.), integrity (Int.) authenticity (Auth.) and Availability (Av).

In the following table, the User Data to be protected by the TOE (as long as in scope of the TOE) are described:

Asset / User Data	Description	Need for Protection			
		Conf	Int.	Auth	Av
D.User_ID_Data	These data correspond to User identification data to be used for identity claim (if required). These data are supposed to be imported from AWPKI Server during enrolment phase.		X		
D.User_Auth_Data	It corresponds to verification authentication data (VAD) provided by User (e.g. PIN, password) and to be used in a derived form in authentication cryptographic operation. Note: Reference Authentication Data provided by user during enrolment is not included in this asset as user enrolment in out of evaluation scope.	X	X	X	

Table 5: User Data

In the next table, the TSF Data to be protected by the TOE (as long as in scope of the TOE) are described:

Asset / User Data	Description	Need for Protection			
		Conf	Int.	Auth	Av
D.AWPKI_SDK_Auth_Key	This asset corresponds to the WBC private key associated to the AWPKI SDK specialized for a device and a user. This key is generated outside of the TOE and imported in the TOE. The private key is associated with a public key and a public-key certificate. Only WBC private key is stored in file system by OS mobile in a self-protected format*. Note: self-protected format means that access to the file does not help to gain access to the key secret and usage of this file with associated WBC cryptographic library gives erroneous result.	X	X	X	
D.AWPKI_SDK_Auth_Algo	This asset corresponds to the authentication algorithm implemented using WBC in AWPKI SDK. This algorithm is required to generate the result of cryptographic operation used by authentication request. Note: usage of this WBC cryptographic implementation without associated WBC private key in self-protected format produces erroneous result.	X	X	X	
D.AWPKI_SDK_ID_Data	These data correspond to AWPKI SDK identification data. These data are supposed to be imported during enrolment phase and stored in the TOE.		X		
D.Server_Auth_Data	It corresponds to authentication data provided by AWPKI Server including (but not limited to):	X	X		X

Asset / User Data	Description	Need for Protection			
		Conf	Int.	Auth	Av
	D.TTBD: Text-to-be-displayed as data to be displayed (DTBD) to the user in plaintext on Mobile output (e.g screen). **D.DTBD**: Data-to-be-displayed to the user on Mobile output (e.g screen) **D.Auth_Challenge:** Data used as input in signature generation.				
D.Mobile_Auth_Data	It corresponds to authentication data linked to mobile designed to allow (but not limited to) detection of wrong hardware to prevent cloning of AWPKI SDK.	X	X	X	
D.APP_Auth_Data	It corresponds to data linked to customer application designed to deter cloning of AWPKI SDK.	X		X	
D.AR	This asset corresponds to Authentication Request sent by AWPKI Server leading to Authentication Cryptographic Operation (ACO) performed by AWPKI SDK. This request is received from an authenticated server and associated response including D.ACOR is transferred to the server for verification.		X		X
D.ACOR	This asset corresponds to result of Authentication Cryptographic Operation Result (ACOR) performed by AWPKI SDK and transferred to AWPKI Server for verification.		X		
D.ACO	This asset corresponds to execution of Authentication Cryptographic Operation (ACO) performed by AWPKI SDK. Such execution is only possible when conditions are fulfilled.				X

Table 6: TSF Data

3.4 ASSUMPTIONS

In the following, according to the threat model as outlined in the following chapter 0, assumptions about the environment of the TOE that need to be taken into account in order to ensure a secure operation of the TOE are listed.

The assumptions for the TOE (A) will be defined in the following manner:

A.Name	Short title Description of the assumption.
A.MobileOS_Mgt	**Mobile Operating System management** Mobile Operating System offers services to interact with final user, to store key file in file system with restricted access to Customer application and API to perform operation for binding SDK with mobile.
A.Application	**Application services used by SDK** Application provides services to interact securely with AWPKI server as connectivity, access to server RSA public key and application rights to store ECDSA keys. Application provides services to interact securely with user to display message for authentication request and to manage user authentication data (RAD and VAD).
A.User_Auth_Data _Mgt	**User Authentication Data management by user and AWPKI SDK Server** User keeps its Authentication Data (e.g. PIN, password) securely and do not use for a different purpose than authentication service with AWPKI SDK. This data is provided by user during enrolment phase and transferred securely to AWPKI SDK Server. **D.User_Auth_Data** is stored securely in Server. A secure update of **D.User_Auth_Data** is provided in administration features.
A.Key_Mgt	**Authentication Key management by administrator** AWPKI SDK Server securely generates and keeps User authentication keys (private key also named **D.AWPKI_SDK_Auth_Key** and public key and associated certificate), and then **D.AWPKI_SDK_Auth_Key** is transferred securely to the mobile and it is not use for a different purpose than authentication service with AWPKI SDK. A secure update of **D.AWPKI_SDK_Auth_Key** is provided in administration features.
A.AWPKI_Server_Mgt	**AWPKI Server management** AWPKI Server is considered to be trusted for user and mobile registration, credential generation and authentication service management
A.User_Registration	**Secure User Registration and credential generation** User Registration and credential generation has been performed according security rules defined in Operational guidance
A.Neglected_User	**User neglected behaviour** User is considered to have no neglected or malicious behavior as leaving Mobile unlocked and unattended.
A.PKI	**PKI Usage** It is assumed that the server environment is secure, providing a PKI that generates a certificate for the AWPKI SDK authentication public key. The PKI

	also manages the validity of certificate, their end of validity, their possible revocation, in such a way that the AWPKI Server can rely on the certificate provided by the PKI.
A.Administration	**Administration of the TOE** The administration of the TOE is done securely and under the control of the AWPKI Server administrator. It is assumed that the Administrator is trustworthy and well-trained, in particular in view of the correct and secure usage of the TOE.
A.Key_Generation	**Authentication Key generation** It is assumed that key pair generation is performed by AWPKI Server under control of an authorized person in a way that preserves the integrity and confidentiality of the private keys and integrity of public keys and relevant certificate. The cryptographic keys are supposed to be generated in conformance to the rules and recommendations defined by the relevant Certification Body.
A.ExtData_Protection	**Protection of Sensitive data outside TOE** Where copies of sensitive data protected by the TOE are managed outside of the TOE, other entities are supposed to provide appropriate protection for copies of that data that may exist outside the TOE.

3.5 THREATS

This section describes the threats to be averted by the TOE independently or in collaboration with its IT environment. These threats result from the TOE method of use in the operational environment and the assets stored in or protected by the TOE.

Find here a refinement of our security model

The “Attacker” role is defined as follows:
Attacker: An attacker is any individual who is attempting to perform actions, which they are not allowed by their access rights. For such purpose, attacker can try to obtain direct access to TOE (under control of User) or Server (under control of Administrator) or to access to communication channel between TOE and Server without notice of User. Attacker can try to obtain TOE code and data and try to use them in an alternative environment. Such person may have expertise, resources and motivation as expected from an entity with given potential in relevant AVA_VAN.3 assurance requirement.

As defined in [CC-3] APE_SPD.1.2C, All threats shall be described in terms of a threat agent, an asset, and an adverse action. In the following threat description, several items (marked with *) are all part of the adverse action.
We refine assets using definition of primary assets and secondary assets.
Primary assets are the most valuable assets for the Customer (integrating SDK in application) and User of the TOE, it is the direct target for the attacker.
Secondary assets are the ones used in security features to protect the primary assets. The attacker tries to gain access or knowledge on secondary assets to obtain access to primary asset.
(e.g.: Money is Primary asset, Money can be protected by storage in a Safe. Combination or Key of the safe is the main secondary asset allowing to open the safe and gain access to money)

In our threat model, we have to consider which assets are directly Compromised [C] versus their security properties (Confidentiality, Integrity, Authenticity, Availability, Non Repudiation) and which one are subject to Unauthorized use [U] or Denial of use [D].
Therefore, in following threat description, asset is potentially associated to letters:
[C] for Compromised, [U] for Unauthorized use, [D] for Denial of use.

The threats to the TOE (T) will be defined in the following manner:

Threat ID	Threat description	Compromise asset
T.ACOR_STOLEN_MOBILE_BF_VAD	Attacker succeeds to compute a fake ACOR (D.ACOR) that will be inserted in a genuine authentication request (D.AR) considered as valid by the server in the following conditions: * on a stolen the user mobile with SDK installed and user enrolled on AWPKI SDK, * giving usage in genuine context of ACO, access to Data used to compute ACO and * using guessed user VAD by conducting brute force attack against VAD (D.User_Auth_Data) on stolen user mobile connected to APWKI server.	Primary Assets: D.ACOR [C], D.AR [C] Secondary Assets: D.User_Auth_Data [C] D.AWPKI_SDK_Auth_Key [U] D.AWPKI_SDK_Auth_Algo [U] D.Server_Auth_Data [U] D.Mobile_Auth_Data [U] D.APP_Auth_Data [U]
T.ACOR_ATT_DEV_CLONING_ROOTING	Attacker succeeds to compute a fake ACOR (D.ACOR) that will be inserted in a fake authentication request (D.AR) considered as valid by the server in the following conditions: * using a malicious app installed on user mobile with SDK installed and user enrolled on AWPKI SDK, to retrieve memory context including application binary, Auth_key using hacking techniques (as rooting) * giving usage in genuine context of ACO, access to Data used to compute ACO, * on the attacker device with stolen binary and data (cloning) * using social engineering to obtain user VAD	Primary Assets: D.ACOR[C], D.AR [C] Secondary Assets: D.ACO [C] D.AWPKI_SDK_Auth_Key [U] D.AWPKI_SDK_Auth_Algo [U] D.Server_Auth_Data [U] D.Mobile_Auth_Data [U] D.APP_Auth_Data [U] D.User_Auth_Data [C]
T.ACOR_ATT_DEV_FAKE_SDK_HOOKING_ROOTING	Attacker succeeds to compute a fake ACOR (D.ACOR) that will be inserted in a fake authentication request (D.AR) considered as valid by the server in the following conditions: * installing genuine SDK on attacker device and performed reverse engineering combined with hacking techniques (hooking, tampering) to create a fake SDK. * using a malicious app installed on user mobile with SDK installed and user enrolled on AWPKI SDK to retrieve memory context including Data used to compute ACO using hacking techniques (as hooking and rooting), * on the attacker device with a fake SDK (reverse) and stolen data (cloning).	Primary Assets: D.ACOR [C], D.AR [C] Secondary Assets: D.ACO [C] D.AWPKI_SDK_Auth_Key [C] D.AWPKI_SDK_Auth_Algo [C] D.Server_Auth_Data [U] D.Mobile_Auth_Data [U] D.APP_Auth_Data [U] D.User_Auth_Data [C]
T.ACOR_ATT_EAVESDROPS	Attacker succeeds to predict and to compute a fake ACOR (D.ACOR) that will be inserted in an authentication request (D.AR) considered as valid by the server in the following conditions: * Eavesdropping the communication exchanges between the genuine user mobile and AWPKI server, * analyzing protocol exchange in order to predict the ACOR for any authentication request * illegitimately replays steps of authentication request protocol allowing unauthorized authentication cryptographic operation by SDK	Primary Assets: D.ACOR [C], D.AR [C]

Table 7: Threat Definition

3.6 ORGANIZATIONAL SECURITY POLICIES

This section specifies the organizational security policies (OSP) that the TOE and its environment shall comply.

The organizational security policies for the TOE (P) will be defined in the following manner:

P.Name	Short title Description of the organizational security policy.
P.Protocol	**Authentication protocol** The AWPKI SDK implements an authentication protocol based on robust algorithm with an accurate strength of function. ACOR received from SDK is decoded and verified to complete user authentication protocol.
P.Credential	**Credentials** The user credentials (ECDSA keys) are generated specifically for each final user in a self-protected format by Server and transferred to SDK and stored in Mobile file system to be used by AWPKI SDK according to AWPKI SDK administration guidance. Key value cannot be extracted from self-protected format. Result of user credential modification will affect ACOR resulted in failure of AWPKI server verification process. It allows user to be sure that SDK cannot be used by any other user if its mobile remains on its control. The customer credential (AESP key) is generated specifically for each customer in a self-protected format by TOE developer and transferred to Customer to be integrated in Customer application. Such key is imported in SDK at each initialization and erased after usage. It allows customer to be sure that SDK cannot be used by other application than the one defined by customer. A policy of credential update is defined in operational guidance in consistency with Customer risk management.
P.Random	**Random number generation** The Server shall generate random numbers for its own use (e.g. challenge and the generation of ECC key pairs) with an accurate process.
P.Secure_Channel	A Secure channel between TOE and AWPKI server is established to authenticate the server and to manage confidentiality and integrity of exchange between TOE and AWPKI server.
P.SDK_Init	SDK initialization is done securely thanks to support of application providing data as server RSA public key used for secure channel establishment.

4. SECURITY OBJECTIVES

4.1 GENERAL

This chapter describes the security objectives for the TOE and the security objectives for the operational environment.

The security objectives for the TOE (O) and the security objectives for the operational environment (OE) will be defined in the following manner:

O/OE.Name **Short title**

Description of the objective.

This section identifies and defines the security objectives for the TOE and its environment. Security objectives reflect the stated intent and counter the identified threats, as well as comply with the identified organisational security policies and assumptions.

4.2 SECURITY OBJECTIVES FOR THE TOE

This chapter describes the security objectives for the TOE which address the aspects of the identified threats to be countered by the TOE independently of the operational environment as well as the organizational security policies to be met by the TOE independently of the operational environment.

O.Integrity **Integrity of User Data or TSF Data**

The TOE shall ensure the integrity of Data used as input for ACO, D.AR received by TSF and D.ACOR produced by TSF in TSF scope of control.

O.Confidentiality **Confidentiality of User Data or TSF Data**

The TOE shall ensure the confidentiality of D.AR received by TSF and D.ACOR produced by TSF and confidential TSF Data during usage (especially authentication private keys and user authentication data) under the TSF scope of control.

O.AccessControl **Access control to functionality and objects**

The TOE shall provide and enforce the authorization to sensitive operations and access control to objects. The TOE shall enforce that only authenticated entities with sufficient access control rights can access restricted objects and services. The access control policy of the TOE shall bind the access control right to an object to authenticated entities.

O.Prot_Malfunction **Protection against malfunction of the TOE**

The TOE shall ensure its correct operation or detect operation outside the normal operating conditions to preserve a secure state of ACO state machine.

O.ACO_SM	**Authentication Cryptographic Operation State Machine** The TOE shall manage authorization to compute authentication cryptographic operation using a state machine.
O.ACO	**Authentication Cryptographic Operation** The TOE shall securely compute authentication cryptographic operation based on a robust algorithm with an accurate strength of function to obtain a reliable result.

4.3 SECURITY OBJECTIVES FOR THE OPERATIONAL ENVIRONMENT

The following security objectives for the operational environment of the TOE are defined:

OE.Trusted_Admin	**Trustworthiness of the Administrator** The administrator shall be trustworthy and well-trained, in particular in view of the correct and secure usage of the TOE and AWPKI Server.
OE.Trusted_Channel	**Trusted Channel** A trusted channel between the Server and TOE for protection of the confidentiality and integrity of the sensitive data transmitted between the Server and the TOE. It also allows TOE to authenticate the AWPKI server.
OE.User_Awareness	**User Awareness about protection of user authentication data** User is aware about how to protect its user authentication data against disclosure and misuse. More precisely, user authentication data shall not be used for different purpose than authentication service with AWPKI SDK. User is able to request a user authentication data update through SDK interface as defined in operational guidance.
OE.PKI	**PKI usage** PKI is used for generating and managing certificate associated to SDK authentication public key is used and manages certificate according to security rules defined in Operation Guidance.
OE.Trusted_Server	**Trustworthiness Server** The server implements security services as defined in AWPKI SDK integration guidance. In particular, Random number generation, ECDSA key generation and provisioning, user enrolment including RAD import, binding with user identity and storage, User authentication request. It also performs ACOR reception, decoding and verification to complete user authentication protocol.
OE.MobileOS	**Secure Service from Mobile OS** The Mobile OS implements security services as defined in AWPKI SDK integration guidance. In particular, interaction with user for capture of User authentication data, for file storage and access used for ECDSA key usage, for binding Mobile with SDK.
OE.Application	**Application Service used SDK** The Application implements services as defined in AWPKI SDK integration guidance. In particular, it provides connectivity to SDK through available network to receive AR from server and provides AWPKI Server public key to allow setup of secure channel. It also provides application context to store ECDSA key in mobile OS file system and user interface to capture securely user authentication data (RAD and VAD) or display message for user authentication.
OE.Credential_Renewal	**Renewal of Credentials** SDK administration (on request of Customer) is able to produce new Credentials for a given Final user and to provision it in user mobile with notification.

OE.Application_Update	Update of Application and SDK Gemalto has its own update policy for the SDK (defined on Guidance) that Customer has to conform. SDK update will require that Customer performs an update of any application containing SDK in addition to its own regular update policy. Customer has supposed to inform final user about its policy of periodical update of any Application containing SDK using regular store update. Final user has to conform to Customer policy of periodical update otherwise usage of application will be blocked for such final user.

4.4 SECURITY OBJECTIVE RATIONALE

Security objectives rationale is not provided in PUBLIC version.
Refer to complete version (only delivered in specific case under NDA) for details of this section.

5. EXTENDED COMPONENT DEFINITION

This Security Target uses components defined as extensions to CC Part 2 [CC2]. The component FPT_MUL is a new component to be used for application evaluation used in mobile environment.

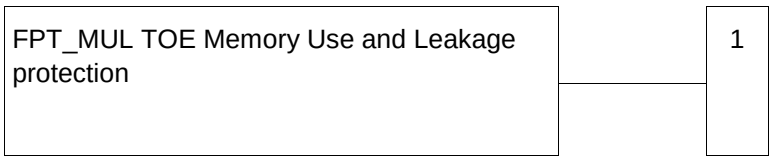
5.1 Definition of the Family FPT_MUL

The family FPT_MUL (TOE Memory Use and Leakage protection) of the Class FPT (Protection of the TSF) is defined here to describe the IT security functional requirements of the TOE related to leakage of information based on inspection of memory use during cryptographic operation. The TOE shall prevent attacks against the TOE and other secret data where the attack is based on deduction of useful information from analysis of memory use (also named {DCA}). This family describes the functional requirements for the limitation of intelligible exploitation of usage of memory for cryptographic operation which are not directly addressed by any other component of CC Part 2 [CC2].

Family Behaviour

This family defines requirements to mitigate intelligible exploitation of usage of memory for cryptographic operation.

Component Levelling



FPT_MUL.1 TOE Memory Use and Leakage protection related to TSF and user data.

Management

FPT_MUL.1 There are no management activities foreseen.

Audit

FPT_MUL.1 There are no actions defined to be auditable.

FPT_MUL.1 TOE Memory Use and Leakage protection

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FPT_MUL.1.1	The TOE shall avoid leakage of data computed in memory during cryptographic operation in excess of [assignment: specified limits] enabling access to [assignment: list of types of TSF data] and [assignment: list of types of user data].

FPT_MUL.1.2	The TSF shall deter inspection of memory usage by protecting simple usage of memory trace to gain access to [assignment: list of types of TSF data] and [assignment: list of types of user data].

6. Security Requirements

6.1 OVERVIEW

This part of the ST defines the detailed security requirements that shall be satisfied by the TOE.

The CC allows several operations to be performed on security requirements (on the component level); refinement, selection, assignment and iteration are defined in sec. 8.1 of CC Part 1 [CC1].

The **refinement** operation is used to add detail to a requirement, and, thus, further restricts a requirement. Refinements of security requirements are denoted in such a way that added words are in **bold text** and removed words are ~~crossed out~~. In some cases an interpretation refinement is given. In such a case an extra paragraph starting with "Refinement" is given. In this ST, refinements made by the author will be noted using ***bold italic and underlined text.***

Note: Detailed refinement for references are included in Refinement note rather than in SFR for clarity reason.

The **selection** operation is used to select one or more options provided by the CC in stating a requirement. Selections to be filled in by the ST author appear in square brackets with an indication that a selection is to be made, [selection:], and are is underlined and italicized like *this*.

The **assignment** operation is used to assign a specific value to an unspecified parameter, such as the length of a password. Assignments to be filled in by the ST author appear in square brackets with an indication that an assignment is to be made [assignment:], and are *italicized*.

The **iteration** operation is used when a component is repeated with varying operations. Iteration is denoted by showing a slash "/", and the iteration indicator after the component identifier. For the sake of a better readability, the iteration operation may also be applied to some single components (being not repeated) in order to indicate belonging of such SFRs to same functional cluster. In such a case, the iteration operation is applied to only one single component.

It should be noted that the requirements in the following chapters are not necessarily be ordered alphabetically. Where useful the requirements have been grouped.

6.2 SECURITY FUNCTIONAL REQUIREMENTS FOR THE TOE

The following table summarizes all TOE security functional requirements (SFR) of this ST organized in 3 SFR groups: Verification authentication data management, AUTH Signature computation, TSF protection.

SFR Group for Verification Authentication Data (VAD) Management

Class FDP: User Data Protection	
FDP_ACC.2/VAD	Complete access control
FDP_ACF.1/VAD	Security attribute based access control
FDP_ITC.1/VAD	Import of user data without security attributes
FDP_RIP.1/VAD	Subset residual information protection
Class FCS: Cryptographic Support	
FCS_COP.1/DerivedVAD	Cryptographic operation performed to derive a VAD used as input for ACO

Table 8: List of Security Functional Requirements for VAD Management

SFR Group for AUTH Signature Computation

Class FCS: Cryptographic Support	
FCS_COP.1/AESP-Dec	Cryptographic operation / AESP Decryption of data used for user authentication
FCS_COP.1/AUTH-User	Cryptographic operation / ECDSA Signature generation for user authentication
Class FDP: User Data Protection	
FDP_ACC.1/AUTH-Comp	Subset access control
FDP_ACF.1/AUTH-Comp	Security attribute based access control
FDP_ITC.1/AESP-Key	Import of user data (AESP-Key) without security attributes
FDP_ITC.1/AUTH-Key	Import of user data (AUTH-Key) without security attributes
Class FPT: Protection of the TSF	
FPT_MUL.1	TOE Memory Use and Leakage protection
FPT_FLS.1	Failure with preservation of secure state

Table 9: List of Security Functional Requirements for Signature Computation

SFR Group for TSF Protection

Class FPT: Protection of the TSF	
FPT_FLS.1	Failure with preservation of secure state

FPT_MUL1	TOE Memory Use and Leakage protection

Table 10: List of Security Functional Requirements for TSF Protection

6.2.1 Class FCS: Cryptographic Support

The AWPKI SDK serves as a cryptographic service provider for the AWPKI Server and provides services in the following cryptographic areas:

- ECDSA Signature Generation used for user authentication and
- AESP deciphering of data contributing to protect ACO.

The TOE supports the following standardized elliptic curve domain parameters (see table 15) for the SFRs of the family FCS_COP.1

Name	Size	Reference
NIST P-256 (secp256r1)	*256*	*[FIPS186, D.1.2.3]*

Table 11: List of standardized elliptic curve domain parameters

The TOE shall meet the requirement “Cryptographic operation” as specified below:

Cryptographic Operation (FCS_COP)
The TOE shall meet the requirement “Cryptographic operation (FCS_COP.1/ AESP-Dec)” as specified below:

FCS_COP.1/ AESP-Dec	Cryptographic operation / AESP Decryption as input for ECDSA signature generation
Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]: fulfilled FDP_ITC.1/AESP-Key FCS_CKM.4 Cryptographic key destruction: not fulfilled.
FCS_COP.1.1/ AESP-Dec	The TSF shall perform [assignment: *decryption of data*] in accordance with a specified cryptographic algorithm [assignment: *AESP*] and cryptographic key sizes [assignment: *128 bits*] that meet the following: *[AESP]*.

Application note: AESP is refined from AES using method described in [AESP] §3.4.

The TOE shall meet the requirement “Cryptographic operation (FCS_COP.1/AUTH-User)” as specified below:

FCS_COP.1/ AUTH-User	Cryptographic operation / ECDSA Signature generation for User Authentication
Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]: fulfilled by FDP_ITC.1/Auth-Key FCS_CKM.4 Cryptographic key destruction: not fulfilled.

FCS_COP.1.1/ AUTH-User	The TSF shall perform [assignment: *signature generation for the interface (sign)*] in accordance with a specified cryptographic algorithm [assignment: *ECDSA*] and cryptographic key sizes [assignment: *256 bits*] that meet the following: *[ISO15946-2] and [FIPS186]* using curves described in Table 11*.

Application note: result of ECDSA signature is encoded in ACOR to prevent malicious operation as spying ACOR and it is decoded and verified by AWPKI Server.

FCS_COP.1/ DerivedVAD	Cryptographic operation / derivation function for VAD
Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]: FDP_ITC.1/VAD FCS_CKM.4 Cryptographic key destruction: not fulfilled.
FCS_COP.1.1/ DerivedVAD	The TSF shall perform [assignment: *a derivation function with input between 4 to 20 bytes and output as a derived secret of 128 bits*] in accordance with a specified cryptographic algorithm [assignment: *PBKDF2*] and cryptographic key sizes [assignment: *128 bits*] that meet the following: *[*assignment: [NIST800-132] referring to [*RFC2898]]*. Application note: Parameters of PBKDF2 are: • PRF: HMAC-SHA256 • P password, an octet string [4-20] • S salt, an octet string [8] • c iteration count, a positive integer [3] • dkLen intended length in octets of the derived key, a positive integer [16]

6.2.2 Class FDP: User Data Protection

The TOE shall meet the requirement “User Data Protection (FDP_ACC.1)” as specified below:

FDP_ACC.1/AUTH-Comp	AWPKI SDK Data subset access control
Hierarchical to:	No other components.
Dependencies:	[FDP_ACF.1 Security attribute based access control] : fulfilled by FDF_ACF.1/AUTH-Comp
FDP_ACC.1.1/ AUTH-Comp	The TSF shall enforce the [assignment: *AUTH-Comp access control SFP*] on [assignment: *list of subjects, objects, and operations among subjects and objects covered by the AUTH-Comp SFP*].

Subject	Object	Operation
S.Command_Manager	*Authentication Computation*	*Access to ECDSA computation for authentication service*
S.Command_Manager	*AESP key*	*Access to key for Challenge deciphering*
S.Command_Manager	*Authentication private key*	*Access to key for ECDSA computation*

S.Command_Manager	*Challenge for ACO*	*Access to Challenge for ECDSA computation*
S.Command_Manager	*Derived VAD*	*Access to Derived VAD for ECDSA computation*
S.External_world	*Authentication Computation*	*No Access to ECDSA computation*
S.External_world	*AESP key*	*No Access to key*
S.External_world	*Authentication private key*	*No Access to key*
S.External_world	*Challenge for ACO*	*No Access to Challenge for ACO*
S.External_world	*Derived VAD*	*No Access to Derived VAD*

Table 12: SDK items for AUTH-Comp access control SFP

FDP_ACC.2/VAD **SDK Verification Authentication Data complete access control**

Hierarchical to: FDP_ACC.1 Subset access control.

Dependencies: [FDP_ACF.1 Security attribute based access control] :
Fulfilled by FDF_ACF.1/VAD

FDP_ACC.2.1/VAD The TSF shall enforce the [assignment: SDK VAD access control SFP] on [assignment: *list of subjects and objects] and all operations among subjects and objects covered by the* SFP described in following table.

Subject	Object	Operation
S.Command_Manager	*Verification Authentication Data (VAD)*	*Access to VAD for derivation function generating the Derived VAD*
S.external_world	*Verification Authentication Data (VAD)*	*No Access to VAD entered by user*
S.external_world	*Derived VAD*	*No Access to Derived VAD computed from User VAD*

Table 13: SDK items for SDK VAD access control SFP

Application note: User consent for data signature operation (using PIN) is not in the scope of FDP_ACC.2.1/VAD

FDP_ACC.2.2/VAD The TSF shall ensure that all operations between any subject controlled by the TSF and any object controlled by the TSF are covered by an access control SFP.

FDP_ACF.1 /AUTH-Comp **AUTH-Comp Security attribute based access control**

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control,
FMT_MSA.3 Static attribute initialization]:
Fulfilled by FDF_ACC.1/AUTH-Comp & FMT_MSA.3 not fulfilled

FDP_ACF.1.1/ AUTH-Comp: The TSF shall enforce [assignment: the *AUTH-Comp access control SFP*] to objects based on the following: [assignment: list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes *as defined in next table*].

Subject/Object	Security attribute
S.Command_Manager	*AT.Phase*
S.Command_Manager	*AT.VAD_Captured*
S.Command_Manager	*AT.AuthKey_Loaded*
S.Command_Manager	*AT.AUTH_Computed*
S.Command_Manager	*AT.TTBD*
S.Command_Manager	*AT.Auth_Requested*

Table 14: SDK security attributes for Auth-Comp Access Control SFP

FDP_ACF.1.2/ AUTH-Comp: The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [assignment: rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects *as defined in next table*].

Operation	Rule
Access to Auth Key	*Phase [S.Command_Manager]==Usage AND Auth_Requested [S.Command_Manager]== YES AND VAD_Captured [S.Command_Manager]== YES AND AuthKey_Loaded [S.Command_Manager]== YES*
Access to AESP Key	*Phase [S.Command_Manager]==Usage AND Auth_Requested [S.Command_Manager]== YES*
Access to Derived VAD	*Phase [S.Command_Manager]==Usage AND Auth_Requested [S.Command_Manager]==Yes AND VAD_Captured [S.Command_Manager]== YES*
Access to Challenge for ACO	*Phase [S.Command_Manager]==Usage AND Auth_Requested [S.Command_Manager]==Yes*
Access to AuthComp	*Phase [S.Command_Manager]==Usage AND Auth_Requested [S.Command_Manager]== YES AND VAD_Captured [S.Command_Manager]== YES AND AuthKey_Loaded [S.Command_Manager]== YES AND AT.TTBD [S.Command_Manager]== YES*

Table 15: SDK security operations for Auth-Comp Access Control SFP

FDP_ACF.1.3/Auth-Comp: The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: [assignment *None*].

FDP_ACF.1.4/Auth-Comp: The TSF shall explicitly deny access of subjects to objects based on the following rules: [assignment: *Modification (other than the import operations) of the authentication private key and the Authentication Protocol sensitive data]*

FDP_ACF.1/VAD **SDK Verification Authentication Data Security attribute based access control**

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control,
FMT_MSA.3 Static attribute initialization]:

fulfilled by FDF_ACC.1/VAD, FMT_MSA.3 not fulfilled

FDP_ACF.1.1/VAD The TSF shall enforce the [assignment: the *VAD access control SFP*] to objects based on the following:

Subject/Object	Security attribute
S.Command_Manager	*AT.Phase*
S.Command_Manager	*AT.Auth_Requested*

Table 16: SDK security attributes for VAD Access Control SFP

FDP_ACF.1.2/VAD The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [assignment: rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects *as defined in next table*].

Operation	Rule
Access to User VAD	*Phase [S.Command_Manager]= Usage AND* *Auth_Requested [S.SecureChannel]= YES AND*

Table 17: SDK security operations for VAD Access Control SFP

FDP_ACF.1.3/VAD The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: [assignment *None*].

FDP_ACF.1.4/VAD The TSF shall explicitly deny access of subjects to objects based on the following rules: : [assignment: *Read, Modification, Storage in NVM of the User VAD and Derived VAD*]

FDP_ITC.1/VAD **Import of user data without security attributes**
Hierarchical to: No other component
Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]
FMT_MSA.3 Static attribute initialization:
Fulfilled by FDF_ACC.1/VAD, FMT_MSA.3 not fulfilled

FDP_ITC.1.1/VAD The TSF shall enforce the [assignment: *VAD access control SFP*] when importing user data, controlled under the SFP, from outside of the TOE

FDP_ITC.1.2/VAD The TSF shall ignore any security attributes associated with the user data when imported from outside the TOE.

FDP_ITC.1.3/VAD The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: [assignment: *any value of byte is valid in User VAD and only VAD with invalid length is checked*].

FDP_ITC.1/AESP-Key **Import of user data without security attributes**
Hierarchical to: No other component
Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]
FMT_MSA.3 Static attribute initialization:
Fulfilled by FDF_ACC.1/AUTH-Comp

FDP_ITC.1.1/ AESP-Key	The TSF shall enforce the [assignment: AUTH-Comp *access control SFP*] when importing user data, controlled under the SFP, from outside of the TOE.
FDP_ITC.1.2/ AESP-Key	The TSF shall ignore any security attributes associated with the user data when imported from outside the TOE.
FDP_ITC.1.3/ AESP-Key	The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: [assignment: *any value of byte is valid in* AESP-*Key and only* AESP-*Key with invalid version is checked*].
	Note: such SFR does not manage the key provisioning and storage of key in application container but the access to the key from application interface to be imported in RAM to be used for AES decryption operation.

FDP_ITC.1/AUTH-Key Import of user data without security attributes

Hierarchical to:	No other component
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_MSA.3 Static attribute initialization: Fulfilled by FDF_ACC.1/AUTH-Comp

FDP_ITC.1.1/ AUTH-Key	The TSF shall enforce the [assignment: AUTH-Comp *access control SFP*] when importing user data, controlled under the SFP, from outside of the TOE.
FDP_ITC.1.2/ AUTH-Key	The TSF shall ignore any security attributes associated with the user data when imported from outside the TOE.
FDP_ITC.1.3/ AUTH-Key	The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: [assignment: *any value of byte is valid in Auth-Key and only Auth-Key with invalid version is checked*].
	Note: such SFR does not manage the key provisioning and storage of key in a file is OS file system but the access to the key from file system to be imported in RAM to be used during ACO.

FDP_RIP.1/VAD Subset residual information protection

Hierarchical to:	No other component
Dependencies:	No dependencies

FDP_RIP.1.1/VAD	The TSF shall ensure that any previous information content of a resource is made unavailable [selection: ~~allocation of the resource to~~, *deallocation of the resource from*] the following objects: [assignment: *User VAD, Derived VAD*].

6.2.3 Class FPT: Protection of the TSF

FPT_MUL.1 **TOE Memory Use and Leakage protection**
Hierarchical to: No other components.
Dependencies: No dependencies.

FPT_MUL.1.1 The TOE shall avoid leakage of data computed in memory during cryptographic operation in excess of [assignment: *non useful information*] enabling access to [assignment: *D.AWPKI_SDK_Auth_Key, D.ACOR*] and [assignment *D.User_Auth_Data*].

FPT_MUL.1.2 The TSF shall deter inspection of memory usage by protecting simple usage of memory trace to gain access to [assignment: *D.AWPKI_SDK_Auth_Key, D.ACOR*] and [assignment: *D.User_Auth_Data*].

Application note: It is based on a statistical analysis of usage of runtime memory.

FPT_FLS.1 **Failure with preservation of secure state**
Hierarchical to: No other components.
Dependencies: No dependencies.

FPT_FLS.1.1 The TSF shall preserve a secure state when the following types of failures occur: [assignment: *see table 23*].

Event detection	Mode	Protection
Hook	*Suspicious mode*	*Normal process & Server notification*
	Confirmed mode	*Authentication request abort & Server notification*
Root	*Suspicious mode*	*Normal process & Server notification*
	Confirmed mode	*Authentication request abort & Server notification*
Debug	*Confirmed mode*	*Authentication request abort & Server notification*
	Anti-debug	*Application delayed crash*
Tamper	*Anti-Tampering*	*Application delayed crash*
	File inconsistency	*Exception throwing and management*

Table 18: Failure with preservation of secure state

6.3 SECURITY ASSURANCE REQUIREMENTS FOR THE TOE

The SAR for the evaluation of the TOE and its development and operating environment are those taken from the Evaluation Assurance Level 3 (EAL3) and augmented by taking the following components: AVA_VAN.3 and dependencies ADV_FSP.4, ADV_TDS.3, ADV_IMP.1, ALC_TAT.1.

The following table lists the assurance components which are therefore applicable to this ST.

Assurance Class	Assurance Component	
Class ADV: Development	Architectural design	(ADV_ARC.1)
	Complete Functional specification	(ADV_FSP.**4**)
	Basic modular design	(ADV_TDS.**3**)
	Implementation representation of the TSF	(**ADV_IMP.1**)
Class AGD: Guidance documents	Operational user guidance	(AGD_OPE.1)
	Preparative user guidance	(AGD_PRE.1)
Class ALC: Life-cycle support	CM capabilities	(ALC_CMC.3)
	CM scope	(ALC_CMS.3)
	Delivery	(ALC_DEL.1)
	Development security	(ALC_DVS.1)
	Life-cycle definition	(ALC_LCD.1)
	Well-defined development tools	(**ALC_TAT.1**)
Class ASE: Security Target evaluation	Conformance claims	(ASE_CCL.1)
	Extended components definition	(ASE_ECD.1)
	ST introduction	(ASE_INT.1)
	Security objectives	(ASE_OBJ.2)
	Derived security requirements	(ASE_REQ.2)
	Security problem definition	(ASE_SPD.1)
	TOE summary specification	(ASE_TSS.1)
Class ATE: Tests	Evidence of coverage	(ATE_COV.2)
	Testing: basic design	(ATE_DPT.1)
	Functional testing	(ATE_FUN.1)
	Independent testing - sample	(ATE_IND.2)
Class AVA: Vulnerability Assessment	Focused Vulnerability analysis	(AVA_VAN.**3**)

Table 19: List of Security Assurance Requirements

6.4 SECURITY REQUIREMENTS RATIONALES

6.4.1 Security Functional Requirements Rationale

Security Functional Requirements rationale is not provided in PUBLIC version.
Refer to complete version (only delivered in specific case under NDA) for details of this section.

6.4.2 Security Assurance Requirements Rationale

Security Assurance Requirements rationale is not provided in PUBLIC version.
Refer to complete version (only delivered in specific case under NDA) for details of this section.

6.4.3 Security Requirements – Internal Consistency

Content of Security Requirements internal consistency chapter is not provided in PUBLIC version.
Refer to complete version (only delivered in specific case under NDA) for details of this section.

7. TOE SUMMARY SPECIFICATION

7.1 TOE SECURITY FUNCTIONS

TOE Security Functions are provided by the SDK in combination with mobile OS.

Identification	Name
SF.USER_VAD_MANAGEMENT	User VAD is received from application char by char and derived on the flight in Derived VAD. Then it is performed Derived VAD object management, with VAD initialization, VAD access, VAD value wiping from volatile memory. There is no VAD storage in NVM. All operations are done in a way to avoid to reveal VAD value. (Note: operation for User RAD initialization with server is under responsibility of Application and it is out of the scope).
SF.MOBILE_BINDING	Implement operations binding sensitive data to the mobile associated to the authorized user in order to allow detection of cloning of application from other device. (Note: management of binding key is under responsibility of OS and is out of the scope).
SF.CUSTOMER_APP_BINDING	Implement operations binding sensitive data used in user authentication to the Customer application in order to deter cloning of application in other device. (Note: management of AESP key is under responsibility of application and is out of the scope).
SF.ACO_ACCESS_CONTROL	It performs access control to ACO using a dedicated state machine using check on security attributes and access control to inputs in ACO. It also includes management of operations (as clear, access and update) on sensitive data involved in ACO.
SF.ACO	Performs ECDSA signature using proprietary scheme using authentication data and authentication key. Result of operation is provided to server to complete the authentication request. Cryptographic operation is done in a way to mitigate brute force attack on authentication data or key. Cryptographic operation is done to avoid any disclosure of computation result and authentication key value during computation. (Note: it includes a white box cryptographic library to perform ACO and obfuscation techniques to be protected against reverse engineering techniques).
SF.AUTH_KEY_MANAGEMENT	Performs key object management, with key initialization, key access and key usage and index for authentication purpose. All operations are done in a way to avoid to wiring key usage and to reveal authentication key value. (Note: operation for key generation in server and provisioning in mobile are out of the scope).
SF.OPERATION_PROTECTION	Implement security protections (as code obfuscation, white box crypto usage and anti-debug feature) to increase complexity of reverse engineering of code and to deter any change on control flow execution.

Table 20: TOE Security Function List

7.2 TOE SUMMARY SPECIFICATION RATIONALE

7.2.1 TOE Security Functions Rationale

The TOE security functions rationale is not provided in PUBLIC version.
Refer to complete version (only delivered in specific case under NDA) for details of this section.