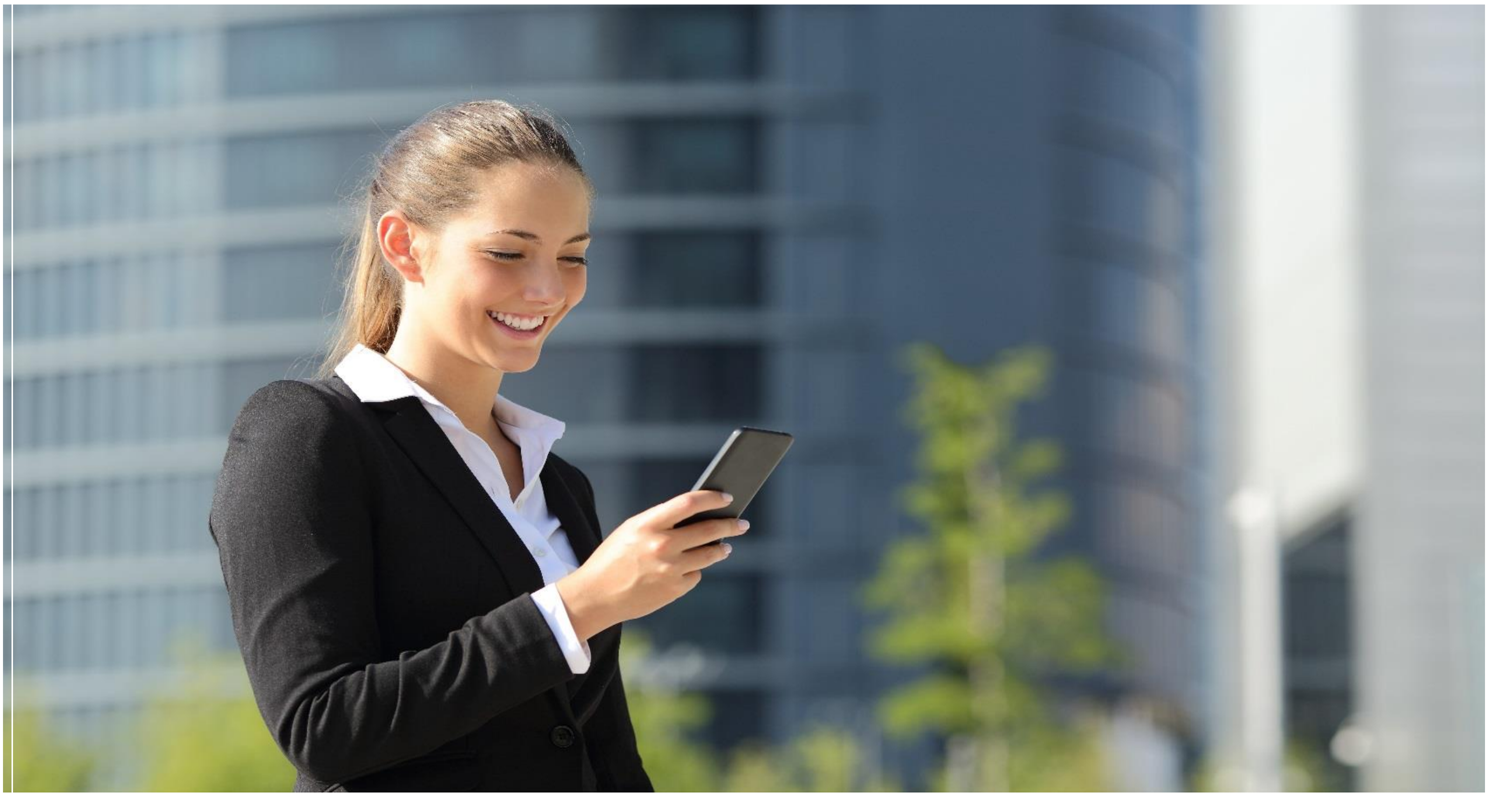


Mobile Protector v5.2.0

Security Target - iOS

All information herein is either public information or is the property of and owned solely by Gemalto NV. and/or its subsidiaries who shall have and keep the sole right to file patent applications or any other kind of intellectual property protection in connection with such information.

Nothing herein shall be construed as implying or granting to you any rights, by license, grant or otherwise, under any intellectual and/or industrial property rights of or concerning any of Gemalto's information.

This document can be used for informational, non-commercial, internal and personal use only provided that:

- The copyright notice below, the confidentiality and proprietary legend and this full warning notice appear in all copies.
- This document shall not be posted on any network computer or broadcast in any media and no modification of any part of this document shall be made.

Use for any other purpose is expressly prohibited and may result in severe civil and criminal liabilities.

The information contained in this document is provided "AS IS" without any warranty of any kind. Unless otherwise expressly agreed in writing, Gemalto makes no warranty as to the value or accuracy of information contained herein.

The document could include technical inaccuracies or typographical errors. Changes are periodically added to the information herein. Furthermore, Gemalto reserves the right to make any change or improvement in the specifications data, information, and the like described herein, at any time.

Gemalto hereby disclaims all warranties and conditions with regard to the information contained herein, including all implied warranties of merchantability, fitness for a particular purpose, title and non-infringement. In no event shall Gemalto be liable, whether in contract, tort or otherwise, for any indirect, special or consequential damages or any damages whatsoever including but not limited to damages resulting from loss of use, data, profits, revenues, or customers, arising out of or in connection with the use or performance of information contained in this document.

Gemalto does not and shall not warrant that this product will be resistant to all possible attacks and shall not incur, and disclaims, any liability in this respect. Even if each product is compliant with current security standards in force on the date of their design, security mechanisms' resistance necessarily evolves according to the state of the art in security and notably under the emergence of new attacks. Under no circumstances, shall Gemalto be held liable for any third party actions and in particular in case of any successful attack against systems or equipment incorporating Gemalto products. Gemalto disclaims any liability with respect to security for direct, indirect, incidental or consequential damages that result from any use of its products. It is further stressed that independent testing and verification by the person using the product is particularly encouraged, especially in any application in which defective, incorrect or insecure functioning could result in damage to persons or property, denial of service or loss of privacy.

September 9, 2019

Contents

This document describes the security target for first level security certification (CSPN) of product *Mobile Protector* by the “Agence nationale de la sécurité des systèmes d'information” (ANSSI).

Document History

Version	Author	Description
1.0	Thales DIS – Sébastien Petit	First version.
1.1	Thales DIS – Sébastien Petit	• Remove reverse engineering threat (not in perimeter anymore). • Remove self-protection from Security Functions (e.g. obfuscation), considered as support functions. • Clarify that on iOS platforms, biometric can be either fingerprint or face, depending on what the mobile equipment support. • Update product version to 5.2.0 to include latest patches. • Explicitly exclude EMV QR, MSP and DCVV from evaluation perimeter to avoid any ambiguity. • Update minimum iOS version to 10 for this product version.
1.2	Thales DIS – Sébastien Petit	Fix after review with CESTI during CSPN kick-off meeting • Removing Secure Storage and Password Manager from “Threat Description” chapter (bad information from Android Security Target copy). Those items are not in the perimeter for iOS platform as described into paragraphs “Product usage Description” and “Product Evaluation Perimeter” since first ST version. • Fix references on product documentation.

References

Reference	Description
[PG]	Mobile Protector Programmers’ Guide. On-line documentation available at https://idcloud-authentication.docs.stoplight.io
[SG]	Mobile Protector Security Guideline. On-line documentation available at https://idcloud-authentication.docs.stoplight.io
[HOTP]	https://tools.ietf.org/html/rfc4226
[TOTP]	https://tools.ietf.org/html/rfc6238
[OATH]	https://tools.ietf.org/html/rfc6287
[RGS_B1]	Référentiel Général de Sécurité, Annex B1 Règles et recommandations concernant le choix et le dimensionnement des mécanismes cryptographiques.
[RGS_B2]	Référentiel Général de Sécurité, Annex B2 Règles et recommandations concernant la gestion des clés utilisées dans les mécanismes cryptographiques.
[RGS_B3]	Référentiel Général de Sécurité, Annex B3 Règles et recommandations concernant les mécanismes d’authentification.

Product identification

Organization	Thales Group DIS
Organization website	www.thalesgroup.com
Product Name	Mobile Protector
Evaluated Product Version	5.2.0
Product Category	Identification, Authentication and Access Control
Product Programmer's Guide	https://idcloud-authentication.docs.stoplight.io

Product arguments

General product description

Mobile Protector is part of a solution for one time password (OTP) generation, secure storage and out of band messages exchange. The solution is composed by a library *Mobile Protector and Messenger* for mobile application and many server components: *Mobile EPS* (Enrolment and Provisioning Server), *MSM* (Mobile Secure Messenger) and an Authentication Server. The solution enables developer to integrate a strong authentication layer for mobile users.

The library *Mobile Protector*, target of this certification, provides to mobile application developers an abstraction layer for security functions. The library provides mechanisms for provisioning and storage of secret keys involved in OTP generation. The library provides also a service for messages exchange and transaction verification as well as a secure storage mechanism.

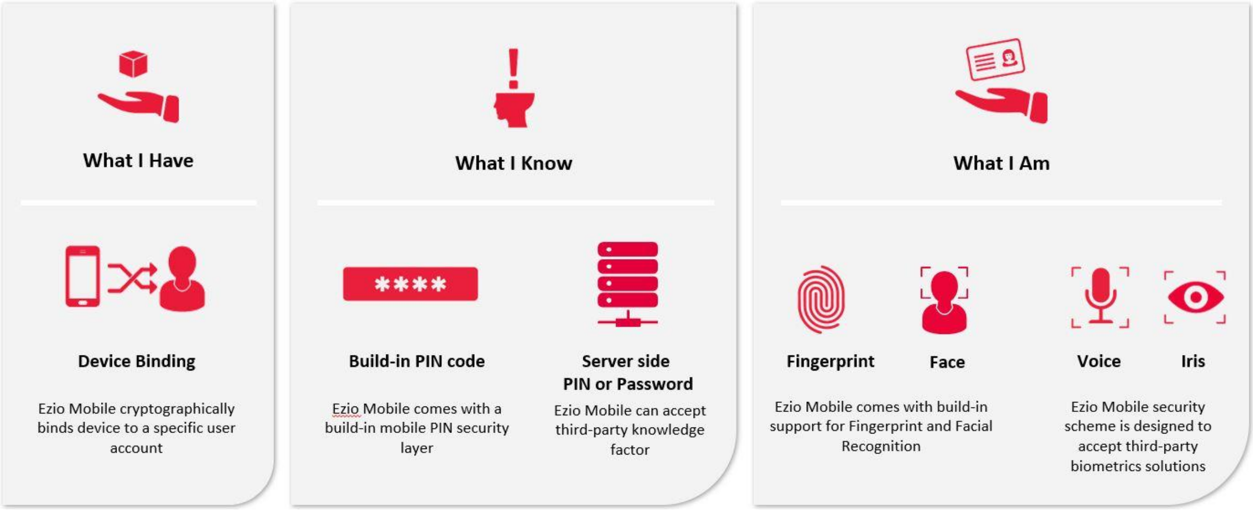
The evaluation perimeter contains the library *Mobile Protector*, the mobile equipment library, restricted to required services for "Identification, Authentication and Access Control" category.

Authentication

The *Mobile Protector* implements the following authentication factors:

- Category "What I have":
    - A mobile equipment that is running a strong authentication application built on top of the product *Mobile Protector*. The library manage and protect the secret key needed to provide strong authentication service.
- Category "What I know":
    - A Pin value known by the user and required in order to provide the strong authentication service.
- Category "What I am":
    - A biometric fingerprint or face Id, provided by the mobile equipment, optionally required to provide the strong authentication service.

In this document, mention of biometrics indicate either fingerprint or face depending on the support in the mobile equipment.



Device Binding

Ezio Mobile cryptographically binds device to a specific user account

Build-in PIN code

Ezio Mobile comes with a build-in mobile PIN security layer

Server side PIN or Password

Ezio Mobile can accept third-party knowledge factor

Fingerprint **Face**

Ezio Mobile comes with build-in support for Fingerprint and Facial Recognition

Voice **Iris**

Ezio Mobile security scheme is designed to accept third-party biometrics solutions

The *Mobile Protector* library is available for both iOS and Android mobile equipment. This evaluation perimeter covers iOS platform.

Product usage description

The service provided by the product and in the evaluation perimeter is the OTP generation. In addition, the Secure PinPad to input a Pin is in the perimeter.

Other features of *Mobile protector*, like the Out of Band messaging, Secure Storage, Password Manager and support functions are not in the perimeter.

One Time Password

The user, as a client of the remote service, downloads the strong authentication application that embeds the product *Mobile Protector*, from an application store.

Mobile Protector provides several mechanisms that need to be used by the application to setup and then use strong authentication service:

- Provisioning of a secret key from the back-end component into the mobile equipment. This is the process to inject the secret into user’s device. The secret will be protected by several security layers and sealed with device information before it is stored.
- Generation of OTP values after user authentication. The previously injected secret can be used to generate OTP for authentication. It is not possible to generate a valid OTP without user authentication. The generation requires the factor “What I have” and at least one of the categories “What I know” or “What I Am”.

The *Mobile Protector* supports OTP generation algorithms: CAP, OATH ([HOTP], [TOTP] and [OCRA]), Gemalto OATH and Dynamic Signature (a Gemalto proprietary algorithm). OATH OTP implementation can be standard or based on a white-box cryptography library.

This security target perimeter includes only the OATH OTP generation algorithm and implemented in the standard form.

Secure PinPad

The strong authentication application can optionally invoke Secure Pin Pad when a Pin is required from the user. Secure Pin Pad provides a way to manage and manipulate the Pin input process in a secured, controlled and verified manner.

Product environment description

Authentication

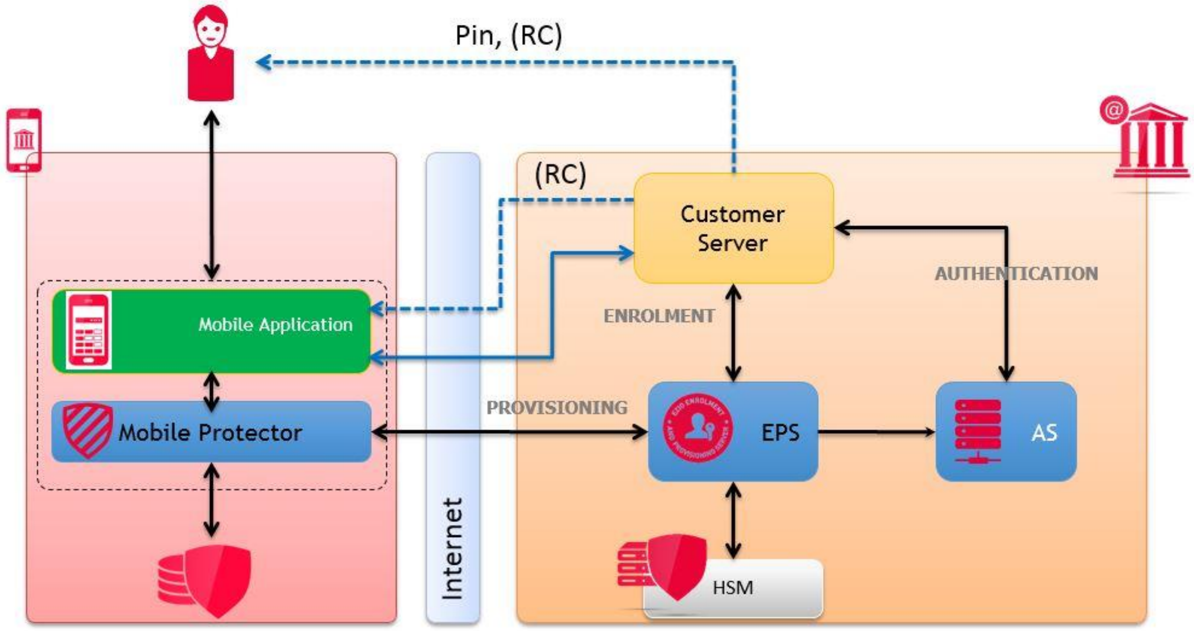


FIGURE 1 - SOLUTION FOR OTP GENERATION

The product environment is mainly related to provide a strong authentication to customer service.

During **enrolment** the customer back-end server request the *Mobile EPS* to create a new user. It implies at least the generation of the secret that will be injected (provisioning) into the end user mobile equipment in next steps, an initial Pin value and a Registration Code (RC) that uniquely identify the secret. Pin and Registration Code will be send to the user or to the application, depending on the deployment use case. The *Mobile EPS* will also push the secret into the Authentication Server deployed for the solution for further authentication of the user. Although enrolment is a mandatory step to create a new user into the solution, the *Mobile Protector* is not involved into this first step.

During **provisioning** the *Mobile Protector* will securely retrieve the secret from the *Mobile EPS*, uniquely identified by the Registration Code.

During **nominal** usage (user authentication), the user can use the strong application to authenticate himself to the service. First the user needs to authenticate himself by one of the activated mechanism (Pin or Biometric in this evaluation perimeter), then the application will use *Mobile Protector* and this authentication mean to request an OTP generation. Depending on the deployment use case, the OTP can be sent to the customer server by the application or displayed to the user for further input on a website for example.

When an OTP is received by the customer server, the service checks the OTP validity against the Authentication Server linked to the *Mobile EPS* to grant access or not to the user.

Environment hypothesis

- HM1: The user is managing is mobile equipment in a way to minimize security risks:
    - The mobile equipment operating system is up-to-date and the latest security patches available are applied.
    - An operating system lock, considered as robust, is active.
    - The user doesn't record his Pin in the mobile equipment nor transmits it to a third party. The Pin is not used for any other usage than the strong authentication application.
- HM2: The mobile equipment enforces a first level of protection:
    - The root certificate authority of the mobile equipment is considered as trusted.
    - The mobile equipment has capacity to connect to the solution servers.
    - The random number generator has sufficient entropy.
    - The cryptography primitives provided by the operating system are resistant to state-of-the-art "basic" attacks.
- HM3: The strong authentication application using *Mobile Protector* follows guidelines defined into [PG] and [SG].
- HM4: The biometric service provided by the operating system is designed to protect assets bound to the biometric authentication and minimize false positive. It is compliant with the [RGS_B3] rules about User Authentication and Machine Authentication with a trusted local environment.
- HM5: The remote service must have an operational and trusted *Mobile EPS* and Authentication Server. The remote service usage by a client must enable the association of an OTP with the according user profile recorded in the Authentication Server.
- HM6: The platform's provided random number generator has sufficient quality to be used as source of entropy to a random number generator designed to be compliant with [RGS_B1] rules.

Dependency description

At least iOS version 10 is required for the *Mobile Protector* to run.

To activate and use the biometric authentication, the mobile equipment must have a hardware sensor (Touch ID) for fingerprint and the user have enrolled one or more fingerprint. Face ID is supported on iOS 11.1.1 and above. To activate and use the Face ID, the mobile equipment must have the Face ID support and the user have enrolled with the Face ID.

Users and typical roles

- MU1: the user has access to the strong authentication application developed with the *Mobile Protector* and is able to authenticate himself by one of the activated mean (Pin or Biometric). The user is typically a client of the service that need to authenticate himself to the remote service. The user is not considered malicious or excessively careless (HM1).

Product evaluation perimeter

The evaluation perimeter is the *Mobile Protector* product, restricted to the following features, linked to the authentication and secure storage services, in particular the various authentication factors and the authentication code generation (the proof of identity):

Authentication

- OTP OATH generation (OCRA included).
- OATH secret provisioning from the *Mobile EPS* (restricted to the Provisioning Protocol v3).
- OATH secret storage into the mobile equipment.
- Secure PinPad to input securely the Pin.
- User authentication by Pin.
- User authentication by biometric (Touch ID or Face ID, depending of platform support).

The following elements are not considered within the evaluation perimeter:

- OTP CAP: computation and provisioning.
- OTP OATH in white-box cryptography format.
- EMV QR.
- MSP (Mobile Signing Protocol).
- DCVV (Dynamic CVV).
- Provisioning Protocols other than v3 (in particular PPv5 required for WBC).
- Dynamic Signature (proprietary protocol conforming CAP).
- Offline provisioning, seed importation inheritance (migration from *Mobile Protector* 1.x to 2.x).

- OTP VIC verification.
- “Dual seed” support for OATH.
- Out Of Band service.
- Secure Storage.
- Password Manager.

Technical environment for product usage

The *Mobile Protector* library is developed to be run on standard iOS mobile equipment.

Hardware compatibility

The technical environment to use the strong authentication application, developed with the *Mobile Protector*, requires a physical handset (smartphone or tablet) supporting mobile application execution environment and IP networking (through SIM card with data subscription or through WIFI).

The optional user authentication mechanism with biometric requires a hardware sensor on the physical handset.

Operating system

The strong authentication application developed with the *Mobile Protector* under evaluation requires mobile equipment running iOS version 10 up to version 12 on architectures `armv7` or `arm64`.

Assets to be protected by the product

The sensitive assets to be protected are ones involved into the authentication (via Pin or Biometric), the secret key generated by the *Mobile EPS* that is associated to a unique user.

While provisioning of the secret key on the mobile equipment, other local keys are generated to ensure the secret key confidentiality and provide security services. All these assets are used in further steps to generate an OTP, enabling user authentication on a remote service.

In the following description, names match the schemes described into the Cryptographic Mechanisms document.

The following sensitive assets are identified for the **Authentication** perimeter:

- B1: Pin, this asset is volatile, provided by the user.
- B2: the secret key `TOK` used to generate OTP values, stored in persistent memory on the mobile and wrapped by several layers of encryption. Each layer provides a different security service, platform security, device binding and user authentication:
    - B2.1: the storage key `sek.` This asset is volatile, derived from `MK` (asset B3) and is unique to each secret key `TOK`. It is used into platform security layer.
    - B2.3: the environment key `eek`. This asset is volatile, derived from data collected from the mobile equipment. It is used into device binding security layer.
    - B2.3: the pin key `kek`. This asset is volatile, derived from user Pin. When `TOK` is provisioned and not upgraded yet to multi-authentication mode `kek` is used to wrap `TOK` (B2). After upgrade to multi-authentication mode the `kek` is used to wrap `awk` (B2.4).

  In case the secret key `TOK` (B2) has been upgraded to support multi-authentication mode the secure scheme is updated and new assets are introduced:

    - B2.4: the authentication wrapper key `awk.` This asset is stored encrypted in persistent memory. It is used to wrap the `TOK` (B2). The `awk` is wrapped by the different authentication scheme (Pin or Biometric).
      For Pin authentication the `awk` is wrapped with `kek` (B2.3).
      For biometric authentication the `awk` is wrapped with `biofpkek` (B2.7).
    - B2.5: the asymmetric biometric key (`biofpk_pub, biofp_priv`). This asset is persistent, the private part `biofp_priv` is protected by the platform, with access control bound to biometric service. The product uses the match on device as authentication factor.

- B2.6: the biometric data `biofpdata`. This asset is stored encrypted in persistent memory with the public part of the asymmetric biometric key `biofpk_pub` (B2.5).
- B2.7: the biometric enciphering key `biofpkek`. This asset is volatile, derived from `biofpdata` (B2.6). It is used to wrap `awk` (B2.4) when biometric authentication mode is activated on the `TOK` (B2).

During the provisioning the secret key `TOK` is transferred from the server to the mobile equipment.

- B2.8: the session key for confidentiality `psk`. This asset is volatile, generated by the mobile equipment, wrapped with server's public key `eps_pub` (B2.10) and sent to the server for response encryption.
- B2.9: the session key for authentication `pak`. This asset is volatile, generated by the mobile equipment, wrapped with server's public key `eps_pub` (B2.10) and sent to the server for message authentication.
- B2.10: the *Mobile EPS* public key `eps_pub`. The asset is persistent, provided by the application to the *Mobile Protector* product.

- B3: the master key `MK` (OTP domain). This asset is securely stored into the iOS keychain.

Those assets usage depends on the activation or not of authentication modes. After provisioning the `TOK` is always protected only with user Pin authentication. Application that uses *Mobile Protector* needs first to upgrade it to support multi-authentication mode and then activate the biometric if needed. The table below presents these steps and list the assets involved in `TOK` protection in persistent memory at the different time.

	Provisioning	**Upgrade to multi-authentication mode**	**Activate biometric**
T0	**T1**	**T2**	**T3**
Mobile Protector doesn't contain any `TOK` to compute an OTP.	A `TOK` has been provisioned into the mobile equipment. It is by default protected by user Pin (and the other security layers). Persistent state: • `(((TOK)kek)eek)sek`	The `TOK` security scheme has been upgraded to support several authentication mode but only Pin is active so far. Persistent state: • `(((TOK)awk)eek)sek` • `(awk)kek`	Biometric has been activated. Persistent state: • `(((TOK)awk)eek)sek` • `(awk)kek` • `(awk)biofpkek`

			• (biofpdata) biofpk_pub • (biofpk_pub, biofp_priv)
	User can generate OTP with his Pin.	User can generate OTP with his Pin.	User can generate OTP with his Pin or Biometric.

Threat description

The security model of the *Mobile Protector* has been designed to counter-act the following attack vectors:

- Attack during secret key exchange between the mobile equipment and the EPS server;
- Attack on secret key stored on the mobile handset (offline attack);
- Attack on user authentication;
- Attack during cryptographic computation (for the various domains, OTP computation and Secure Storage)

The following threat scenarios are identified:

- M1 - mobile equipment theft: a malicious user can try to impersonate the legitimate user or try to access application’s secret (physical access to the mobile equipment).
- M2 - Brute force on a secret or authentication factor:
    - Pin.
    - Biometric.
    - Secret keys.
- M3 - Pin or password or secret key access in real time during the cryptographic operations.
- M4 - Threats on the authentication code:
    - OTP replay.
    - Forge a new OTP from a valid one.
    - Reverse engineering of authentication factor from a valid OTP.
- M5 - Secret key interception during provisioning.
- M6 - Pin and/or password theft;
- M7 - Secret key cloning;

Security function description

FS1 - Pin management

User Pin is one factor that can be used in the authentication scheme. The secret injected into the mobile equipment from the *Mobile EPS* always comes protected by the user Pin. In case other factors are activated (biometric for example), they will be in a second step, after provisioning.

Secure Pin Pad

Secure Pin Pad is a single visual view that provides security for data entries. The purpose of Secure Pin Pad is to ensure that the management and manipulation of the Pin code are conducted in a secured, controlled, and verified way. It ensures that the protection mechanisms are in place to defeat or mitigate key logger, over shoulder attacks, memory dump and screen capture. Secure Pin Pad helps a software solution to prevent targeted attacks more efficiently.

The Secure Pin Pad feature uses internally a Gemalto library to handle Secure Key Pad. You can configure elements of the Secure Pin Pad using the initialization parameters or through APIs offered by the product. On Pin entry success, the Secure Pin Pad feature outputs a `SecurePin` object that can be used later into change Pin or OTP generation flows.

Pin blockage

The secret key is encrypted with the Pin on the mobile. The application cannot influence this fundamental principal. The security relies on the property that a bad Pin must generate a bad OTP that is not distinguishable from a good OTP.

The attacker cannot validate an OTP without submitting it to the authentication server for checking. The authentication server enforces a threshold policy to limit the number of wrong OTP and disable the user service if required.

There isn’t any mechanism in *Mobile Protector* that permits to know the entered Pin is the good one or not and no way to block locally the Pin in the *Mobile protector*.

Pin change

The Pin can be changed in the strong authentication application, developed with the *Mobile SDK* library. This operation involves a modification of the encryption layer of the secret key that integrates the new Pin.

FS2 - Biometric management

Once the secret to generate OTP is provisioned into the mobile equipment, it is possible to activate the authentication with biometric, if hardware and operating system requirements match the minimum required ones.

Multi-Authentication mode

By default the secret to compute OTP comes protected by the user Pin value and cannot be used with other authentication factors. In order to allow usage of other authentication factors the secure scheme used to protect the secret needs first to be upgraded.

During upgrade the user needs to authenticate himself by using his Pin. The secure scheme is then upgraded to allow both Pin and Biometric to unlock the secret. This upgrade step is needed only once per secret and cannot be reversed.

As there isn’t any way to know locally the Pin is correct, providing a wrong Pin during secure scheme upgrade will lead to an invalid state that cannot be reversed, the secret will not be able to generate valid OTP anymore. It is then strongly recommended to validate an OTP with the authentication server with the provided Pin before starting the upgrade. This is the only possible way to validate the Pin was correct.

Biometric activation

Biometric can be activated after the secure scheme has been upgraded. It can be activated and used per secret to generate OTP if the platform support it and if user has preciously enrolled himself into the mobile equipment biometric system. The user needs to provide his Pin in order to activate the new authentication factor.

Any secret that has been upgraded to the multi-authentication mode secure scheme can support the biometric authentication factors.

Biometric deactivation

At any time the application can request to deactivate the biometric support for any of the secret.

FS3 - Confidentiality protection of secret key during provisioning

The secret key provisioning on mobile handset from the *Mobile EPS* involves two security layers, a TLS connection that is used to transport messages of a proprietary protocol. The following steps are involved into the proprietary protocol, to protect the secret key during every exchange operations:

1. The strong authentication application starts and detects that the secret key is missing.
   This state triggers a process to get a secret key for OTP generation from the back-end, identified by a registration code. Depending on the deployment scenario this registration code can be entered by the user or retrieved by the application from a back-end server.
   The registration code, bound to the user, uniquely identify the secret key on the back-end side.

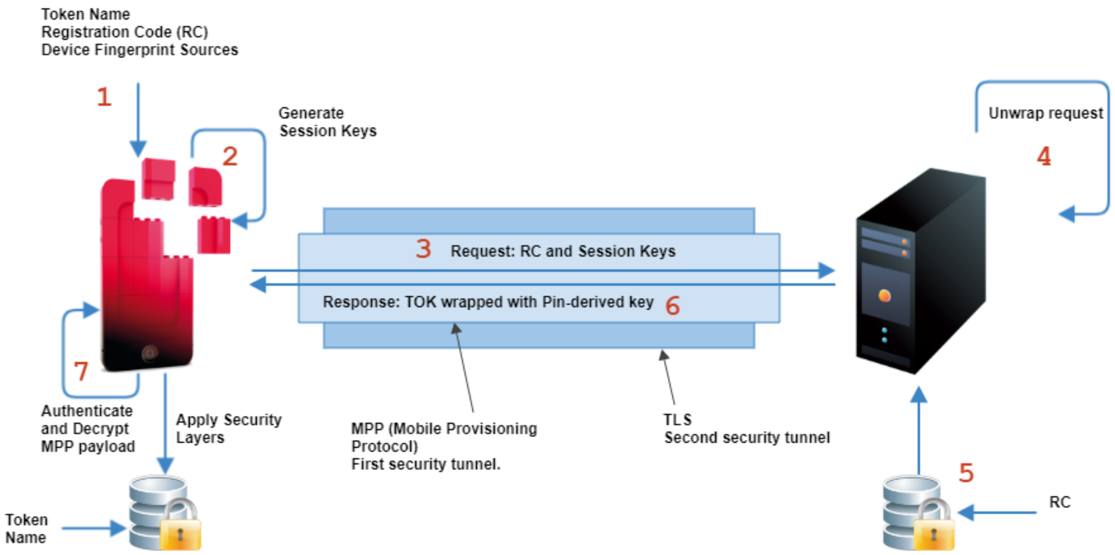
2. The *Mobile Protector* generates two symmetric session keys, for encryption and authentication.
3. Those two keys and the registration code are wrapped with the public key of the *Mobile EPS* and send to it.
4. On the *Mobile EPS*, the security module unwrap the two symmetric keys and the registration code with the private key of the *Mobile EPS*.
5. On the *Mobile EPS*, the secret key (already encrypted with the user Pin) corresponding to the registration code is selected.
6. The security module encrypts the secret key again with symmetric key for encryption and computes a HMAC with the authentication key. The *Mobile EPS* gets the encrypted and authenticated message and sent it to the mobile application.
7. On reception in the mobile equipment the message is authenticated with the HMAC value and then decrypted.

The secret key confidentiality is ensured during all the steps since an encryption layer with the Pin is enforced during all the process. The Pin is not needed during the provisioning, and the secret key is never in plaintext. Moreover a mutual authentication between the mobile application and the *Mobile EPS* is enforced:

- The mobile application is authenticated when the registration code is submitted for validation. Hence the secret key is delivered to an authorized user.
- The registration code is encrypted with the public key of the server targeted. This ensures that the registration code is used only by the authorized server.

The *Mobile Protector* rejects any plaintext communication (HTTP), self-signed certificate, host mismatch and enforce the server certificate to be signed by a root-CA trusted in the mobile. Those security checks cannot be deactivated.

The *Mobile Protector* apply then several security layers to provide anti-cloning and platform security features before storing the secret into the persistent memory.



FS4 - Confidentiality protection of secret key stored for OTP

The secret key is received to the mobile with one encryption layer from the Pin. Without the Pin, an attacker that eavesdrop the secret key cannot read it. Knowing the value of encrypted secret key, there isn't any mechanism or structure that allow to retrieve the correct Pin.

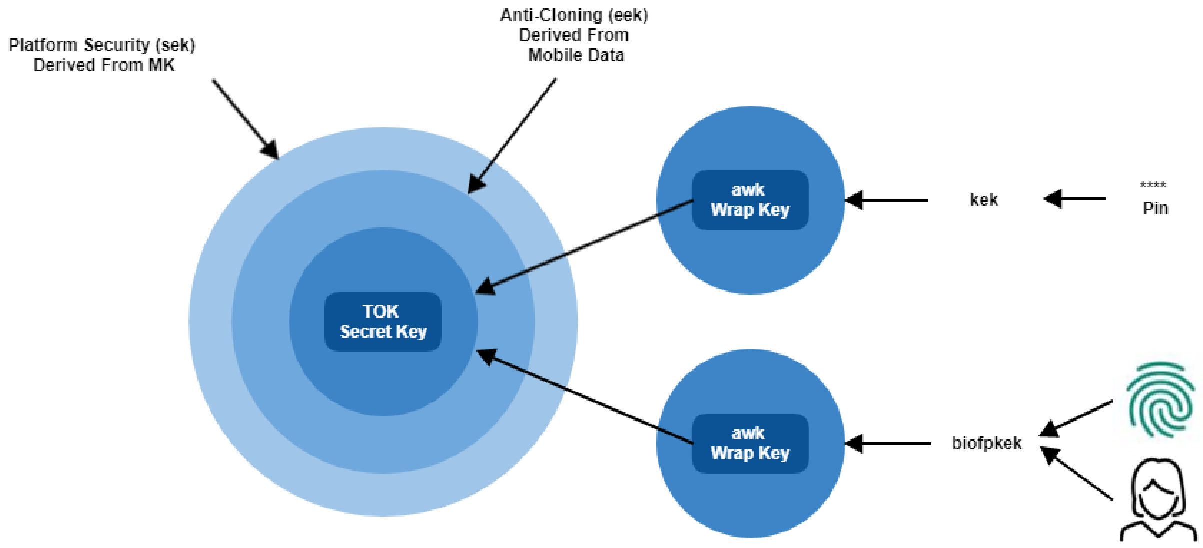
The confidentiality protection of the keys stored on the mobile equipment follow these principles:

1. The enciphered secret key is encrypted with a second layer using a volatile environment key built with device collected data and a salt generated randomly.
2. The double enciphered secret key is then encrypted with a third layer using a key derived from the master key MK (asset B3) and a salt generated randomly.
3. The data is natively protected by the segregation of mobile execution environment managed by the operating system that prevents an application to access to the data of another application.

In case of multi-authentication mode upgrade, the first security layer provided by the Pin-derived key is replaced by a wrap key. The wrap key is stored in the persistent memory encrypted:

- By the Pin-derived key. It will allow further user authentication with its Pin.
- By a biometric key with access control bound to the Biometric mechanism. It will allow further user authentication through the Biometric sensor.

The figure below presents the various security layers involved into the secret protection (note that all intermediate assets are not represented).



FS5 - Confidentiality protection of keys during OTP computation

The OTP computation requires secrets from 3 different environments:

- The operating system: the master key MK (asset B3) is used to derive the storage key in order to access the secret key enciphered three times. After deciphering, the secret key is still encrypted two times.
- The execution environment: device data is collected and used to derive the environment key in order to access the secret key enciphered two times. After deciphering, the secret key is still encrypted one time.
- The user authentication: the Pin or the Biometric is needed to remove the last encryption layer.
    - The Pin is used to derive a key used to decrypt the final wrap key.
    - The biometric is used to unlock a key to decrypt the final wrap key.

Finally the secret for OTP computation is unwrapped with the latest key obtained from the user authentication layer. When the secret key is accessible in memory, the OTP generation is achieved.

If the mobile equipment is stolen (M1) but the Pin is not disclosed, the attacker is not able to brute force (M2) the secret key in a feasible time. The attack on Pin (M2) is not realistic since no stop condition is involved and that the validation is done on the Authentication server.

If the mobile equipment is stolen (M1) and the biometric has been activated, the attacker is not able to unlock the secret thanks to the platform biometric security enforced by the platform.

When the Pin is stolen (M6) but the mobile equipment is not accessible, the attack to generate valid OTP is not possible as the secret key cannot be brute forced.

In case of secret key eavesdropping during the provisioning (M5), i.e. knowing the secret key but not the Pin, the attack is not feasible to distinguish a valid Pin since no stop condition is involved.

In case of application cloning (M7) on another mobile equipment and without the Pin, the attack to recover the secret key is not possible thanks to the anti-cloning security layer.

If the Pin or the secret key is accessed in real time during the cryptographically operations (M3), the attack is difficult and requires a rooted mobile with a malicious application whereas the sensitive data are wiped after use. The support functions for environment detection (rooted detection, hooked detection, emulator detection and debugger detection) increase the difficulty of the attack.

Integrity protection of sensitive data

The sensitive assets for OTP generation are not protected in integrity by construction. Indeed, according to the fundamental principle used in the mobile solution, a bad OTP must not be distinguished from a good OTP. The sensitive assets might be altered by an attacker without breaking the security model, any attempt to alter assets will lead to invalid OTP generation and invalid authentication will be detected on the authentication server.

FS6 – One Time Password algorithm

Mobile Protector, restricted to this evaluation perimeter, uses standard OTP algorithms [HOTP], [TOTP] and [OATH].

Threat coverage by product security functions

	M1: Mobile equipment theft	**M3:** Brute force	**M4:** Access to asset during cryptographic	**M5:** Threat on authentication code	**M6:** Secret Interception during provisioning	**M7:** Pin or password theft	**M8:** Secret key cloning
FS1 – Pin Management	X	X				X	
FS2 – Biometric management	X	X					
FS3 – Confidentiality of secret key during provisioning			X		X		
FS4 – Confidentiality of secret key stored for OTP	X	X	X				X
FS5 – Confidentiality of keys during OTP computation	X	X	X				
FS6 – One Time Password algorithm			X	X			

Protection rationale

Threat	Protection
M1: mobile handset theft	The secret key is protected by the Pin or the Biometric. • The Pin is not known by the attacker. No data derived from the Pin is stored in persistent memory, mitigating any brute force attack on it even with full mobile equipment reverse and analysis. Pin can only be validated by submitting the generated OTP to the authentication server, the server enforce strict retry counter management in a secure environment, blocking the account after a defined number of false OTP. • Biometric verification is provided by the mobile equipment manufacturer. Access control to keys to unlock the secret to

	generate OTP is enforced by the platform into hardware-backed keystore. • Secret key for OTP as well as application's data are sotred encrypted into the mobile equipment.
M2: Brute force	• No data derived from the Pin is stored in persistent memory, mitigating any brute force attack on it even with full mobile equipment reverse and analysis. Pin can only be validated by submitting the generated OTP to the authentication server, the server enforce strict retry counter management in a secure environment, blocking the account after a defined number of false OTP. • The product relies on mobile equipment for biometric • Cryptographic secret uses state-of-the art recommendation on key strength and cannot be brute forced.
M3: Access to asset during cryptographic operation	This attack requires a rooted mobile handset and a dedicated malicious application installed on the mobile handset. The *Mobile Protector* enforce a proper wiping of all assets at the end of usage. Secret key for OTP computation are managed in the native part of the product (not in Java layers). Support security functions make attacks on the field more complex (root detection, hook detection, tamper detection, debugger detection, emulator detection, overlay detection).
M4: Threats on the authentication code	*Mobile Protector* uses standard OTP algorithms, reviewed by experts. Replay is not possible and is enforced in Authentication Server.
M5: Secret key interception during provisioning	All the communications between the *Mobile Protector* and the *Mobile EPS* are encrypted with MPP (the Gemalto proprietary Mobile Provisioning Protocol) and TLS protocols.
M6: Pin and/or password theft	Useless without the secret key stored on the mobile equipment.
M7: Secret key cloning	A copy of the secret key stored on the mobile equipment cannot be used on another one since the volatile environment key is unique by mobile. A copy of the secret key without Pin is useless and keys involves into the Biometric are protected into hardware-backed keystore.