Corporate Signatures Mediator Server Protection Profile

Aleksander Lasecki, 236371
Grzegorz Zaborowski, 236447
Paulina Jałosińska, 232892
Mikołaj Grzegrzółka, 241073

Contents

1 Protection Profile Introduction

This section introduces the description of the TOE(Target of Evaluation). It also describes major security features of the TOE and required non-TOE software or hardware.

1.1 TOE Description

1.1.1 General description

The Target of Evaluation(TOE), addressed by this Protection Profile(PP), is Signatures Mediator Server used for generating corporate signatures. This PP is for the software being the component of the mediator service.

1.1.2 User's authorisation

The software will provide such services as safe authorisation of the employees with a usage of digital signatures. Privet keys will be divided into two parts and held by a user and assigned server.

1.1.3 User's accesses

Access to services will depend on the needs of employees and corporation and some modifications to the services will not require interference from the System Administrator.

1.2 Major Security Features

1.2.1 Usage of private keys

Major security features of the TOE includes usage of partial private keys. The key will be divided into two parts, where the first part will be held on a device managed directly by the employee and the second part will be on the mediator's server.

1.2.2 Control over user's data

Additional security feature is the control over time window that will allow users to use their keys and constant revocation of the parameter's database, which delivers information about blocked or expired users.

1.2.3 User's settings

Users(employees) will have partial control over certain system settings and functions what means, that some changes will not require System Administrator interaction.

1.3 Required non-TOE software/hardware

The TOE will require few additional components which are necessary for the operation of the entire system. The base, additional components that provides functionality for other programs, devices or services are servers.

1.3.1 Open Authorisation Server

The OAuth(Open Authorisation) will be used to give access to the page that allows modifying employee's accesses. OAuth is an open standard for authorisation. It provides specific authorisation flows for web applications or desktop applications to access to an HTTP service. OAuth allows distribution access tokens to external clients through an authorisation server. Then the external client uses the received token to gain access to protected resources stored on the server. Only the system administrators will have full control over the settings, with some changes requiring

the approval of two administrators. Some of the settings can be modified by employees, but their range will differ and depend on many factors. There will be possibility of changing settings only at some time window.

1.3.2 Trusted Platform Module

The required non-TOE hardware is a Trusted Platform Module(TPM) that is a computer chip that is conforming TPM standard. Task of TPM is usually to secure hardware through integrated cryptographic keys. TPM makes processes such as digital signing more secure and can help ensure that the platform remains trustworthy. Common TPM functions are key creation(and use) and system integrity measurements. The function that will ensure safety of the TOE will be checking if the integrity of the system haven't been compromised by a virus or malware or some unauthorised changes hasn't been made. If an abnormality is detected, it can run in the quarantine mode and remain in it until the proper operation of the system will be restored.

1.3.3 Hardware Security Module

The last required non-TOE hardware is Hardware Security Module(HSM). HSM is a physical device, dedicated crypto processor that main task is to safeguard and manage digital keys and perform encryption and decryption for digital signatures. The functions that HSM will provide are secure cryptographic key storage, cryptographic key generation and cryptographic key management. HSM will also perform encryption and decryption functions for digital signatures. As a result of all cryptographic operations taking place at HSM, which has access control, there will be better prevention from accessing sensitive cryptographic data.

2 Conformance Claims

2.1 CC Conformance Claim

- This Protection Profile has been developed and claims conformance to version **3.1. Release 5** of Common Criteria (CC).
- This Protection Profile is **conformant** with Parts 2 and 3 of the CC. **No extended** SFRs and SARs have been defined.

2.2 PP Claim

This PP does not claim conformance to any other Protection Profile.

2.3 Package Claim

This PP claims assurance package EAL3.

TODO: This section needs to be finished by the addition of augmentations if necessary (after SARs are done).

2.4 Conformance statement

The PP requires **demonstrable** conformance of any PPs/STs to this PP.

3 Security Problem Definition

This section introduces the subjects, objects and assets of the TOE. It also describes threats, operational security policies and assumptions for the TOE.

3.1 Introduction

3.1.1 Subjects

- **System Administrator** - Authorised person responsible for administration and with administrative functions. Admins the TOE, has full access to the TOE, and has the ability to change attributes for the TOE functionality.
- **Employee** - An employee of a company wishing to access the TOE.
- **Auditor** - A person with the ability to view operations of the TOE, verifying them on the basis of data from the log.
- **Adversary** - Any individual who is attempting to subvert the operation of the TOE. The intention may be to gain unauthorised access to the assets protected by the TOE.

3.1.2 Objects

- **Authorisation data base** - A data base holding all information necessary to authorise an employee
- **Revocation parameters data base** - A data base holding information about which employees are no longer able to create signatures
- **User preferences data base** - A data base holding information about when a particular employee can create signatures
- **Partial private key storage** - A secure data base holding the mediator's partial private keys for each employee
- **WWW server** - A portal allowing users to modify their entries in the user preferences data base
- **Cryptographic module** - A hardware module used for creating and verifying signatures
- **Cryptographic policies** - A file contains parameters for cryptographic computations, such as minimum TLS version and key lengths

3.1.3 Assets

- **Private keys used for creating signatures** - Each employee has to have their own partial secret key stored on a computing device owned by them. The mediator server has to store a partial private key for each employee
- **Authorisation data base integrity** - A data base of public keys used by the mediator to authorise employees
- **Revocation parameters integrity and confidentiality** - The mediator server hold a data base which specifies which users are no longer allowed to create company signatures. This data base should only be modifiable by administrators
- **User preferences data base integrity and confidentiality** - The mediator server also holds a data base which specifies at what time particular employees are allowed to create signatures. This data base should be modifiable be the employees

- **Communication data integrity and confidentiality** - Communication packets sent between the mediator server and an employee during an execution of the signing protocol should be protected
- **Legitimacy of signatures** - It needs to be ensured that each signature that can be successfully verified was created be the mediator server and the correct employee (the one whose public key can be used to verify the signature) at a time when the employee's signing capabilities were not revoked
- **Confidentiality of the messages** - The messages that are being signed should not be leaked by the signing protocol
- **Cryptographic policies integrity** - The integrity of the cryptographic computation needs to be strongly protected
- **Availability of the mediator services** - The mediator services (i.e. the ability to create signatures and set up revocation parameters and user preferences) should be available at all times

3.2 Organisational Security Policies

P.AdminCount	**The number of admin needed to perform a change in the cryptographic policies** A specific number of admins should be required to make a change to the cryptographic policies to minimise the chance for unwanted of malicious modifications. It is recommended that this number should be at lest two, but the system should be fully functional if the number is set to just one
P.SignTime	**When can signatures be created** The organisation should specify at what times signatures can and cannot be created. For example, it might company policy that signatures are not to be created on Saturdays and Sundays. This policy might also target specific types of employees, for example, it might allow administrators to create signatures at times when that is not allowed for other employees
P.EmployeePrefs	**What kind of modification can employees make to their user preferences** The organisation should specify what kind of modifications can employees make to their entries in the user preferences data base. For example, the policy might (although it is not recommended) allow employees to give themselves signing permissions during a weekend, even though the company policy says otherwise, or it might (this is the recommended approach) only allow employees to further restricts their signing capabilities (i.e. set the preferences to not have signing permission when it is allowed by the company policy, for example, a specific range of hours every day)
P.Crypto	**Cryptographic policy of the company** The organisation should specify requirements for all cryptographic computations and authorisation processes. This might include setting minimum TLS level, specific or minimum key lengths as well as specifying the computational environment for the cryptographic derivations (for example, the group in which computations are done)

3.3 Threats

This section describes the threats for the TOE.

T.PrivDisclose	**Disclosure of the private keys of the mediator or an employee**
Assets:	Private keys used for creating signatures

Security goal:	The confidentiality of the assets
Adverse action:	The attacker either gains access to a storage device where partial privates keys are held (that can be either the mediator server or an employee owned device that stores their partial private keys) or learns partial or final private key through an attack on the signature protocol (e.g. through listening to a legitimate execution of the protocol)
Attacker:	The attacker either has access a device storing private keys or is able to listen in on or even intercept and modify the communication of the mediator server with a legitimate user (i.e. an employee)

T.Tampering	**Tampering of the signature protocol communication**
Assets:	Communication data integrity and confidentiality, legitimacy of signatures
Security goal:	The integrity of the assets
Adverse action:	The attacker intercepts and manipulates messages sent between the mediator and an employee to create force the parties into signing a different message
Attacker:	The attacker is able to intercept and modify the communication between the mediator and an employee

T.PubKeyModify	**The modification of the public key data base of the mediator server**
Assets:	Authorisation data base integrity, legitimacy of signatures, availability of the mediator services
Security goal:	The integrity of the assets
Adverse action:	The attacker gains access to the public key data base of the mediator server and either add new keys (to be authorised later on), removes keys (to block certain employees from being able to create signatures) or replaces a public key with a different one
Attacker:	The attacker is able to access/modify public key the data base

T.MsgLeak	**The content of the messages are leaked by the signing protocol**
Assets:	Confidentiality of the messages
Security goal:	The confidentiality of the assets
Adverse action:	The attacker listens in on the commutation between the mediator and an employee and, from the captures packets, determines what the signed message was
Attacker:	The attacker is able to capture the communication between the mediator and an employee

T.RevokeFail	**A signature gets created at a time when the signing capabilities of an employee were revoked**
Assets:	Legitimacy of signatures
Security goal:	The integrity of the assets
Adverse action:	The attacker signs a message via the way correct even tough the signing capabilities of the signing employee are revoked at the time of the protocol execution
Attacker:	The attacker has access to the signing API

T.RevokeModify	**The revocation parameters of the mediator server are modified by an unauthorised entity**
Assets:	Legitimacy of signatures, revocation parameters integrity and confidentiality, availability of the mediator services
Security goal:	The integrity of the assets

Adverse action:	The attacker gains access to the revocation parameters data base and changes the settings stored there to either block an employee's ability to create signatures or give themselves signing permissions (e.g. after the company revoked them permanently)
Attacker:	The attacker has access to the revocation parameters data base on the mediator

T.ParamModify	**The user preferences of the mediator server are modified by an unauthorised entity**
Assets:	Legitimacy of signatures, user preferences data base integrity and confidentiality, availability of the mediator services
Security goal:	The integrity of the assets
Adverse action:	The attacker gains access to the user preferences data base and changes the settings stored there to either block an employee's ability to create signatures or give themselves signing permissions (e.g. breaking the company policy for when signatures can be created)
Attacker:	The attacker has access to the revocation parameters data base on the mediator

T.CryptoFail	**The cryptographic module of the mediator malfunctions**
Assets:	Legitimacy of signatures, cryptographic policies integrity, availability of the mediator services
Security goal:	The integrity of the assets
Adverse action:	The cryptographic module might be damaged or malfunctioning in other ways which might lead to incorrect computations of signatures (i.e. signatures created correctly might not be verifiable due to corruption)
Attacker:	The cryptographic module of the mediator

T.AdmErr	**Accidentally made administrator's mistake**
Assets:	Communication data integrity and confidentiality, confidentiality of the messages
Security goal:	The availability of the assets
Adverse action:	An administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms
Attacker:	Accidental administrator error that may prevent access to services or allow unauthorised entry or damage to the system

T.Unauth	**Accidental or intentional unauthorised access**
Assets:	Legitimacy of signatures, confidentiality of the messages
Security goal:	The confidentiality of the assets
Adverse action:	A malicious or careless user may access to data for which they are not authorised according to the TOE security policy
Attacker:	Possibility of unauthorised modification of inappropriate files by a careless user. The attacker gains access with user rights.

T.MalUpdate	**Execution of a malicious update**
Assets:	Legitimacy of signatures, cryptographic policies integrity, availability of the mediator services
Security goal:	The integrity of the assets
Adverse action:	A malicious update is provided to the mediator server. After that update is performed, the server might change its behaviour, e.g. ignoring the revocation parameters. This might have devastating consequences to the security of the system

Attacker:	Someone able to either provide the server with an update file or able to inject malicious code to an existing update file

3.4 Assumptions

A.Room	**Secure room** The mediator server is assumed to be located in a secure room that protects it from being directly accessed to by unauthorised personnel
A.RespAdmins	**Responsible admins** The admins managing the mediator server are assumed to be responsible and to always act in accordance with the company policy

4 Security Objectives

4.1 Security objectives for the TOE

OT.SecProtocol	The protocol chosen for the signature creation should be a secure and approved zero knowledge protocol
OT.MsgHash	Only the hash of the message being signed (or of the message with additional data) should be sent over the communication channel
OT.ManInTheMiddle	The protocol chosen for the signature creation should be resistant to man in the middle attacks
OT.Update	Every update of the software running on the mediator server should be checked for integrity (for example via signatures) before being applied
OT.SecAuth	The authentication method used to authenticate and authorise employees should be trusted and secure
OT.PrivManage	There should be a strong and secure privilege management system in place
OT.RevokeCheck	The mediator should check the revocation parameters during every run of the signing protocol, this information should not be cached
OT.SecWeb	The web server should be implemented using latest tools and tested in terms of security
OT.Audit	The TOE shall provide functionality to generate audit records for security-relevant administrator actions

4.2 Security objectives for the Operational Environment

OE.PhySecure	The room in which the server is placed should only be accessible by authorised personnel
OE.SecStorage	The hardware that the server is running on should provide secure storage for partial secret keys, cryptographic configuration and the authentication data base
OE.AdminPolicy	The organisation should define an administration policy specifying how many administrators are needed to perform sensitive actions such as changes in the cryptographic policy. The company shall ensure that authorized administrators are non-hostile, appropriately trained and follow necessary informations for secure management.
OE.CryptoPolicy	The organisation should define a secure policy for handling cryptographic computations
OE.SignPolicy	The organisation should define a policy for when signatures can be created to maximise their legitimacy, for example, it is recommended that signature creation should be block during days when no employees are at work
OE.PrefPolicy	The company should define a policy specifying how much power do user preferences have. It is recommended that user preferences should only be allowed to further restrict the times when signature creation is allowed
OE.DataProtectionPolicy	The organisation will protect data from unauthorized modification by enforcing an access control policy defined by security supervisers.

4.3 Table of correspondence

Figure 1: Table of correspondence

	OT.SecProtocol	OT.MsgHash	OT.ManInTheMiddle	OT.Update	OT.SecAuth	OT.PrivManage	OT.RevokeCheck	OT.SecWeb	OT.Audit	OE.PhySecure	OE.SecStorage	OE.AdminPolicy	OE.CryptoPolicy	OE.SignPolicy	OE.PrefPolicy	OE.DataProtectionPolicy
T.PrivDisclose	X										X					
T.Tampering			X													
T.PubKeyModify					X	X			X							
T.MsgLeak		X						X								
T.RevokeFail							X									
T.RevokeModify					X	X		X	X							
T.ParamModify					X	X		X	X							
T.CryptoFail													X			
T.AdmErr												X				
T.Unauth					X			X	X							X
T.MalUpdate				X												
P.AdminCount												X				
P.SignTime														X		
P.EmployeePrefs															X	
P.Crypto													X			
A.Room	-	-	-	-	-	-				X						
A.RespAdmins	-	-	-	-	-	-						X				

4.4 Rationale

T.PrivDisclose The protection of secret key parameters should be provided by company according to security policy. Futhermore as it was stated in OE.SecStorage, that the hardware used by the server should secured.

T.Tampering The threat is controlled by securing the user's logical access to the TOE, by specifying the type and strength of the authentication mechanism. Also the protocol chosen according to OT.ManInTheMiddle should be resistant to those type of attacks.

T.PubKeyModify The ability to authorize and access and perform modifications is controlled by the type and strength of authentication in accordance with OT.SecAuth. The privilege management should be secoured as it was stated in OT.PrivManage.

T.MsgLeak Any access to communication between the employee and the mediator is secured in terms of the possibility of interception through the company's security protocols, in this case OT.MsgHash. This also include fully secured web server mentioned in OT.SecWeb.

T.RevokeFail This threat is mitigated by specific times of generating valid signatures and according to OT.RevokeCheck the revocation parameters are checked during every run of the protocol.

T.RevokeModify The possibility of generating valid signatures is limited by specific time points. In addition, the very ability to modify and grant permissions is limited by the signature strength and user type, as stated in OT.SecAuth and OT.PrivManage. In this case it is also ensured that the Web server is fully protected according to OT.SecWeb.

T.ParamModify	Potential threat occurrence is controlled by authorization methods represented by OT.SecAuth. Additionally, the occurrence of a threat is mitigated by the granted permissions with secured privilege management mentioned in OT.PrivManage. Overall web server structure is secured according to OT.SecWeb.
T.CryptoFail	The occurrence of a cryptographic problem is controlled by OE.CryptoPolicy, implemented as the company's cryptographic policy.
T.AdmErr	OE.AdminPolicy helps to mitigate this threat by ensuring the TOE administrators have guidance that instructs them how to administer the TOE in a secure manner. Having this guidance helps to reduce the mistakes that an administrator might make that could cause the TOE to be configured in insecurely.
T.Unauth	The primary purpose of the TOE is to restrict access between subjects and objects. The ability of the TOE to enforce an access control policy against objects in the operational environment allows this purpose to be fulfilled. All subjects are checked according to OE.DataProtectionPolicy. The treat is also mitigated by secured authorization represented by OT.SecAuth and secured web server according to OT.SecWeb objective.
T.MalUpdate	Any possibility of modifying the server and introducing major changes requires more administrators. The potential threat is therefore mitigated by the possibility of authorizing its introduction. Also every update of the softwere is checked according to OT.Update.
P.AdminCount	According to this TOE all administrators are monitored and checked for authorization and secure administration of the TOE stated in OE.AdminPolicy. This also ensures that for vital changes, more than two authorized administrators are required. Administrator role shall be separate and distinct from other authorized users.
P.SignTime	This policy is essential for the company's security. The ability to control the time of generating important signatures and limit the time of a potential threat significantly reduces the possibility of its occurrence. Overall process is secured in accordance with OE.SignPolicy.
P.EmployeePrefs	General control of user preferences is essential as a control mechanism. The company policy defines in OE.PrefPolicy the specific permissions assigned to a given user type and the possibilities obtained through specific permission levels.
P.Crypto	This addresses the requirement to use cryptography, in accordance with OE.CryptoPolicy ; the assignment in the control should correspond with the type of crypto selected.
A.Room	A physical security issue related to the mediator server as stated in OE.PhySecure. The envisaged control room requires appropriate permissions to gain access to it.
A.RespAdmins	All administrator actions are ensured to be according to OE.AdminPolicy.

5 Security Requirements

This chapter defines the IT security requirements that shall be satisfied by the TOE or its environment. Common Criteria divides TOE security requirements into two categories:

- Security Functional Requirements (SFRs) (such as identification and authentication, security management, and cryptographic support) that the TOE and supporting evidence must meet to meet the TOE's security objectives.
- Security Assurance Requirements (SAR) that ensure that the TOE and its supporting IT environment meet security objectives (such as configuration management, guidance documents and vulnerability assessment).

5.1 Security Functional Requirements

5.1.1 Protection of the TSF (FPT)

5.1.1.1 Reliable time stamps (FPT_STM.1)

Dependencies: No dependencies

FPT_STM.1.1 The TSF shall be able to provide reliable time stamps for its own use.

5.1.2 Security Audit (FAU)

5.1.2.1 Audit data generation (FAU_GEN.1)

Dependencies: FPT_STM.1 Reliable time stamps

FAU_GEN.1.1 **Refinement:** The TSF shall be able to generate an audit record of the following auditable events:

a) Start-up and shutdown of the audit functions

b) Start-up and shutdown of the Web Server

c) All auditable events for the minimum level of audit **listed in Table 7**

d) Every modification of TOE configuration data

e) Software updates

f) [assignment: other specifically defined auditable events]

FAU_GEN.1.2 The TSF shall record within each audit record at least the following information:

a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [assignment: other audit relevant information]

Application Note: The "Additional Audit Record Contents" column is used to specify the data that should be included in the audit record if it reasonable in the context of the event that generates the record. If no other information (other than as mentioned in "a)" above) is required for the particular type of event being audited, then an assignment of 'None' is acceptable.

Table 7: Auditable Events

Security Functional Requirement	Auditable Events	Additional Audit Record Contents
FPT_STM.1	None	None
FAU_GEN.1	None	None
FAU_GEN.2	None	None
FCS_CKM.1/RSA	Key generation	Success and failure of the activity
FCS_CKM.2	None	None
FCS_CKM.3	None	None
FCS_CKM.4	Key destruction	Success and failure of the activity
FCS_COP.1/HMAC	None	None
FIA_UID.1	Unsuccessful use of the user identification mechanism	The identity of the subject performing the operation and the type of operation being performed
FIA_UAU.1	Unsuccessful use of the user authentication mechanism	The identity of the subject performing the operation and the type of operation being performed
FIA_AFL.1	Unsuccessful authentication attempts occur	The identity of the subject performing the operation (if can be identify e.g. login), device localisation, time and date, any other possibly useful information
FIA_ATD.1	None	None
FMT_SMF.1	Use of the management functions	Identity of the administrator performing these functions
FMT_SMR.1	Modifications to the group of users that are part of a role	Identity of authorized administrator modifying the role definition
FMT_MOF.1	Disable or enable of the functions in the TSF	All modifications in the behaviour of the functions in the TSF. Identity of the authorized administrator performing these actions
FTA_SSL.3	Initiation of a session termination	Identity of the session being terminated and the reason for the termination
FTA_MCS.1	Exceed of maximum number of session	The identity of the user and device localisation

5.1.2.2 User identity association (FAU_GEN.2)

Dependencies: FAU_GEN.1 Audit data generation
FIA_UID.1 Timing of identification

FAU_GEN.2.1 The TSF shall be able to associate each auditable event with the identity of the user that caused the event.

5.1.3 Cryptographic Support (FCS)

5.1.3.1 Cryptographic key generation (FCS_CKM.1)

FCS_CKM.1.1 The OS shall generate asymmetric cryptographic keys in accordance with a specified cryptographic key generation algorithm.

FCS_CKM.1/RSA Cryptographic key generation (RSA keys)

Dependencies: FCS_CKM.2 Cryptographic key distribution
FCS_COP.1 Cryptographic operation
FCS_CKM.4 Cryptographic key destruction

FCS_CKM.1.1/RSA The TSF shall generate cryptographic RSA keys in accordance with a specified cryptographic key generation algorithm RSA key generator and specified cryptographic key sizes 1024 and 2048 bits that meet the following: [RFC 3447] and [IEEE1363].

5.1.3.2 Cryptographic key distribution (FCS_CKM.2)

Dependencies: FCS_CKM.1 Cryptographic key generation
FCS_CKM.4 Cryptographic key destruction

FCS_CKM.2.1 The OS shall implement functionality to perform cryptographic key establishment in accordance with a specified cryptographic key establishment method:

- RSA-based key establishment schemes that meets the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 8017, "Public-Key Cryptography Standards (PKCS) 1: RSA Cryptography Specifications Version 2.2,
- Elliptic curve-based key establishment schemes that meets the following: NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography",
- Finite field-based key establishment schemes that meets the following: NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography".

5.1.3.3 Cryptographic key access (FCS_CKM.3)

Dependencies: FCS_CKM.1 Cryptographic key generation
FCS_CKM.4 Cryptographic key destruction

FCS_CKM.3.1 The TSF shall perform [assignment: type of cryptographic key access using HSMs] in accordance with a specified cryptographic key access method [assignment: type of cryptographic key access using HSMs] that meets the following [assignment: list of standards].

5.1.3.4 Cryptographic key destruction (FCS_CKM.4)

Dependencies: FCS_CKM.1 Cryptographic key generation

FCS_CKM.4.1 The OS shall destroy cryptographic keys in accordance with the specified cryptographic key destruction methods:

- For volatile memory, the destruction shall be executed by a single overwrite consisting of zeroes.
- For non-volatile memory that consists of the invocation of an interface provided by the underlying platform that logically addresses the storage location of the key and performs a single overwrite consisting of zeroes.

5.1.3.5 Cryptographic operation (FCS_COP.1)

FCS_COP.1.1 The OS shall perform encryption/decryption services for data in accordance with a specified cryptographic algorithm.

FCS_COP.1/HMAC/SHA1

Dependencies: FCS_CKM.1 Cryptographic key generation
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1/HMAC The TSF shall perform HMAC value generation and verification in accordance with a specified cryptographic algorithm HMAC with SHA-1 and cryptographic key sizes 160 bits that meet the following: [FIPS 198-1 29].

FCS_COP.1/HMAC/SHA256

Dependencies: FCS_CKM.1 Cryptographic key generation
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1/HMAC The TSF shall perform HMAC value generation and verification in accordance with a specified cryptographic algorithm HMAC with SHA-256 and cryptographic key sizes 160 bits that meet the following: [FIPS 198-1 29].

5.1.4 Identification and authentication (FIA)

5.1.4.1 Timing of identification (FIA_UID.1)

Dependencies: No dependencies

FIA_UID.1.1 The TSF shall allow [assignment: list of TSF-mediated actions] on behalf of the user to be performed before the user is identified.

FIA_UID.1.2 The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user

Application Note: An example of such an TSF-mediated action might include the request for help on the login procedure, etc.

5.1.4.2 Timing of authentication (FIA_UAU.1)

Dependencies: FIA_UID.1 Timing of identification

FIA_UAU.1.1 The TSF shall allow [assignment: list of TSF mediated actions] on behalf of the user to be performed before the user is authenticated.

FIA_UAU.1.2 The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

5.1.4.3 Authentication failure handling (FIA_AFL.1)

Dependencies: FIA_UAU.1 Timing of authentication

FIA_AFL.1.1 The TSF shall detect when [selection: [assignment: positive integer number], an administrator configurable positive integer within [assignment: range of acceptable values]] unsuccessful authentication attempts occur related to [assignment: list of authentication events].

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall [assignment: list of actions].

Application Note: The TOE shall at least implement a temporary/permanent account lockout when the required number of failed authentication attempts has been reached.

5.1.4.4 User attribute definition (FIA_ATD.1)

Dependencies: No dependencies

FIA_AFL.1.1 The TSF shall maintain the following list of security attributes belonging to individual users: [assignment: list of security attributes].

Application Note: In the context of company signatures, the list of security attributes should contain at least the following parameters: working hours and days (in which it will be possible to use the signature), days of leave, illness.

5.1.5 Security management (FMT)

5.1.5.1 Specification of management functions (FMT_SMF.1)

Dependencies: No dependencies

FMT_SMF.1.1 The TSF shall be capable of performing the following security management functions: update, backup, recovery, user administration, defines some secure attributes and other security management functions.

FMT_SMR.1.2 The TSF shall be able to associate users with roles.

Application Note: Identifies the management functions that are available to the authorized administrator.

5.1.5.2 Security roles (FMT_SMR.1)

Dependencies: FIA_UID.1 Timing of identification

FMT_SMR.1.1 The TSF shall maintain the roles [assignment: the authorised identified roles].

FMT_SMR.1.2 The TSF shall be able to associate users with roles.

5.1.5.3 Management of security functions behavior (FMT_MOF.1)

Dependencies: FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_MOF.1.1 **Refinement:** The TSF shall restrict the ability to **disable** or **enable** the functions [assignment: list of functions] to [assignment: the authorised identified roles].

Application Note: In the case of company signatures, a perfect example is the exclusion (by authorized persons) of the possibility of executing signatures during the holiday season.

5.1.6 TOE Access (FTA)

5.1.6.1 TSF-initiated session termination (FTA_SSL.3)

Dependencies: No dependencies

FTA_SSL.3.1 The TSF shall terminate an interactive session after a [assignment: time interval of user inactivity].

Application Note: To ensure adequate safety, the user's inactivity period should be short, but long enough to ensure trouble-free operation. The recommended value is 15 minutes.

5.1.6.2 Basic limitation on multiple concurrent sessions (FTA_MCS.1)

Dependencies: FIA_UID.1 Timing of identification

FTA_MCS.1.1 The TSF shall restrict the maximum number of concurrent sessions that belong to the same user.

FTA_MCS.1.2 **Refinement:** The TSF shall enforce, by default, a limit of **one** session per user.

Application Note: The previous session should be closed.

Table 8: Coverage of Security Objectives for the TOE by SFRs

	OT.SecProtocol	OT.MsgHash	OT.ManInTheMiddle	OT.Update	OT.SecAuth	OT.PrivManage	OT.RevokeCheck	OT.SecWeb	OT.Audit
FPT_STM.1									X
FAU_GEN.1									X
FAU_GEN.2									X
FCS_CKM.1/RSA	X	X	X						
FCS_CKM.2	X								
FCS_CKM.3	X								
FCS_CKM.4	X								
FCS_COP.1/HMAC		X							
FIA_UID.1								X	
FIA_UAU.1					X			X	
FIA_AFL.1					X			X	
FIA_ATD.1							X		
FMT_SMF.1				X					
FMT_SMR.1								X	
FMT_MOF.1						X	X		
FTA_SSL.3								X	
FTA_MCS.1								X	