Security Target

TrustedX

TrustedX Security Target

Safelayer Secure Communications, S.A.

Phone: +34 93 508 80 90
Fax: +34 93 508 80 91
Web: www.safelayer.com
Email: support@safelayer.com

CONTENTS

CHAPTER 1

1. Introduction

1.1. Security Target and TOE Reference

Document Identifier	0775BA94 v1.7
Title	Security Target - TrustedX
Issue Date	July, 2010
Release Identifier	3.0.10S1R1_T
Authors	Safelayer Secure Communications S.A.
CC Version	Common Criteria version 3.1 Revision 2
Evaluated TOE	TrustedX

1.2. TOE Overview

TrustedX is a Web services platform that, by providing authentication, authorization, electronic signature and data protection, resolves the security and trust problems that arise when business processes exchange documents and information.

Trust is not expressed identically nor does it present the same value and form in different ecosystems; not all data is encoded or exchanged uniformly (for instance, XML, CMS, PDF, ASN.1, etc.), security mechanisms prevail (for instance, login/password, digital certificates, etc.) and it is necessary to manage different trusted entities (for instance, government, corporate, etc.) in the exchange of B2B or B2C data.

ETI—Enterprise Trust Integration—refers to a model of integrating the processes of generating and interpreting the appropriate level of information trust, contemplating the federation of different security domains. The concept is therefore applied to the development of trust in information using standard and especially PKI-based security mechanisms, which is made possible by a trusted services platform such as TrustedX.

TrustedX enables the integration of the different trust elements and detaches consumers from the practical and conceptual complexity involved in implementing the different security mechanisms. Hence, it is an essential component that resolves the problems related to security and trust during the exchange of documents and data that:

- Makes security and trust services independent from business processes.
- Offers a complete and uniform set of security functions.
- Offers a common framework of interoperability with external security domains.
- Classifies and interprets the level of trust of information.

TrustedX includes a comprehensive set of trusted services based on public key infrastructures (PKI) that are standard and service-oriented for any type of consumer: end-users, applications or other services.

The TOE of this Security Target includes most of the TrustedX services:

1 **Authentication and authorization.** Exchanging authentication and authorization information between corporate applications and external security domains, enabling Web single sign-on (SSO) using the standards defined by OASIS.

2 **Key management.** Provides the functions for managing the keystores of the TrustedX entities, such as users and applications (key generation, certificate request, and certificate and key import).

3 **Digital certificate validation.** Recognition of multiple certification service providers, providing the information associated with the certificates in a uniform manner. Supports the standard certificate validation mechanisms and accepts the integration of any other customized mechanism.

4 **Electronic signature**. Supports most digital signature formats for documents, e-mail and web messaging, including multiple signatures, time-stamped signatures and advanced electronic signatures.

5 **Electronic signature custody.** Custody service for the electronic signatures of documents that maintains their validity for long periods of time using advanced signatures, thus implementing long-term electronic signatures for validating the signature once the digital certificates have expired.

6 **Data encryption**. Data protection—for electronic documents, e-mail and Web messaging—via encryption mechanisms.

7 **Auditing and accounting.** Service that manages log data generated by all platform service components and information on their use and/or consumption in a centralized, uniform and secure manner.

8 **Object and entity management.** Broker offering a uniform XML view of objects and entities managed by the platform, completely masking data-specific formats (XML, ASN.1, Text, etc.) and information sources (LDAP, SQL, Files, etc.) and allowing them to be used as Web services.

This complete set of services included in the TOE can be complemented with other services where they comply with Web services integration rules. TrustedX includes additional services that are not included in the TOE.

See 1.3.5 Environment Components for the list of environment components (non-TOE components).

1.3. TOE Description

As described above, TrustedX includes a comprehensive set of trusted services based on public key infrastructures (PKI). The TOE (Target of Evaluation) of this Security Target can be seen as a set of service components that implement authentication and authorization mechanisms, electronic signature and data encryption functions, and the required auxiliary protocols involved in the deployment of applications using public key infrastructure (PKI) services.

The following sections describe the service components of TrustedX that are included in the TOE of this Security Target.

The TOE platform is distributed in the following two formats:

- Hardware Appliance Edition: integrating the hardware with the product.
- Virtual Appliance Edition: providing a virtualization of the Hardware Appliance Edition.

The TrustedX product described in this section consists of the TOE for this Security Target.

The present Security Target considers all the threats, security objectives and functional requirements included in the [PKE_PP] Protection Profile. Additionally, a set of new Common Criteria elements (threats, objectives and requirements) have been included. The new security objectives have been derived, in terms of Common Criteria terminology, from the points included in Annex III "Requirements for secure signature-creation devices" and Annex IV "Recommendations for secure signature verification" of the "Directive 1999/93/EC of the European Parliament and of the Council" of 13 December 1999 on a Community framework for electronic signatures ([EUROPEAN_DIRECTIVE]).

1.3.1. TrustedX Architecture

The TOE consists of a set of Web service components that handles all the above-described functionality. The components are as follows (the following sections contain a detailed description of each one):

- **TrustedX Authentication & Authorization (TWS-AA)**. Authentication and authorization service that includes authentication mechanisms using login/password and certificate (TLS/SSL), both used in a direct standard manner, as well as additional mechanisms based on signatures with X.509 certificates. TrustedX can be easily extended with other authentication mechanisms, such as OTP (one-time passwords), biometrics, etc. These additional mechanisms are not included in the current TOE.
- **TrustedX Entity Profiler (TWS-EP)**. Information management service providing uniform object and/or entity profiles: users, applications, Web services, policies, certificates, logs/audits, etc., which results in a uniform and controlled method for accessing all configuration and audit data.

- **TrustedX Digital Signature (TWS-DS)**. Document digital signature service supporting the generation of different recognized "basic" signature formats (PKCS#7/CMS, PDF Signature, CAdES, XML-DSig/XAdES, S/MIME and WS-Security).

- **TrustedX Digital Non-Repudiation (TWS-DR)**. Advanced digital signature service adding reliable time and revocation information to previously-signed documents as a basis for long-term digital signatures. It supports the generation of different recognized "advanced" signature formats (AdES-EPES, AdES-T, AdES-C, AdES-XL and AdES-A), where AdES stands for advanced electronic signature and applies to CAdES (CMS AdES) and XAdES (XML AdES) signature formats.

- **TrustedX Digital Signature Verification (TWS-DSV)**. Digital signature verification service (including basic and advanced or long-term digital signatures), regardless of the supplier or the certificate and signature format verification mechanisms. It supports all the formats generated by the TWS-DS and TWS-DR components.

- **TrustedX Digital Signature Custody (TWS-DSC).** Custody service for the digital signatures of documents that maintains their validity for long periods of time, thus implementing long-term digital signatures.

- **TrustedX Digital Encryption (TWS-DE)**. Document encryption and decryption service in PKCS #7/CMS, S/MIME, XML-Enc and WS-Security formats.

- **TrustedX Key Management (TWS-KM)**. Provides the functions to securely manage the keystores of the TrustedX entities, such as users and applications (key generation, certificate request, certificate import and key import). These actions can be performed with an on-disk keystore or a keystore based on a HSM device.

The TrustedX platform provides a common management system that includes configuration, monitoring and access control for each service component. The system presents the following features:

- In order to maintain an open and customizable architecture, XML language is used for configuration, customization, monitoring, and audit and control data. This applies to any type of data stored or exchanged at control ports of online services. TWS-EP is the service component devoted to this function.

- Services are accessed through SOAP according to the WSDL specification of each service. Access is controlled using an authentication token that was previously requested from the TWS-AA service. Client–server interaction is performed via HTTP or HTTPS transport, thus enabling the channel to be secured with SSL/TLS with or without mutual authentication. For example, if login/password authentication is requested, it is recommended to use SSL/TLS.

Each TrustedX service component can interact with other corporate or external infrastructure elements, namely:

- **Trusted Third Parties (TTP)**, to which the TOE connects to validate the digital certificates (certification authorities (CA) or validation authorities (VA)) and to obtain time-stamps (time-stamp authorities (TSA)).

- The TOE can operate with an external **cryptographic device (HSM)**.

- **Database (SQL and FILE)**, where the TOE stores log data on the activity of the TrustedX platform's service components for auditing. This data is accessed

transparently by the TOE using the TWS-EP component and can be mapped to SQL or FILE physical repositories.

- **Document Management System (DMS/ECM)**, where the digital signature custody service component can store and manage the documents with signatures, and the encryption component can store encrypted documents. This data is accessed transparently by the TOE using the TWS-EP component and can be mapped to DMS/ECM (or any WebDAV compliant) or SQL physical repositories.

- **Directory**, from/to where the TOE can read and write data on the entities (individuals, applications or Web services) recognized by the platform. This information is accessed transparently by the TOE using the TWS-EP component and can be mapped to LDAP, SQL or FILE physical repositories.

The figure below illustrates the interaction between the mentioned infrastructure elements with the TOE. It also shows interactions with the corporate applications that use the TOE's services. There is also the option, especially for greater numbers of applications and/or if different authentication mechanisms are required, to have an authentication/authorization agent to centralize some or all of the authentication and authorization functions required by the applications.

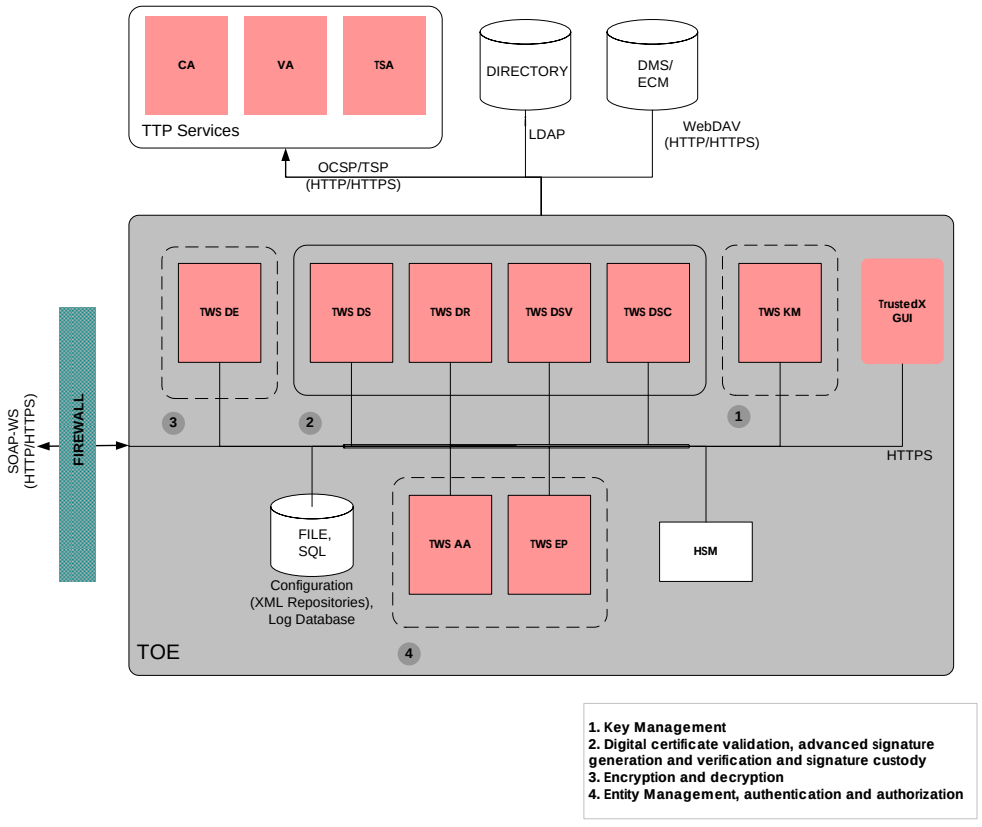

Figure 1-1. TOE architecture and interaction with external components

The figure above shows that the TOE interacts with an HSM component that provides cryptography services. In this case, the security services included in the [PKE_PP] protection profile (claimed in this Security Target) use this HSM[1] for cryptographic operations.

1.3.2. TrustedX Service Components

In this section, the different service components that make up the TrustedX platform are described.

1.3.2.1. TrustedX Authentication & Authorization (TWS-AA)

TWS-AA is a key element of the platform and is responsible for authenticating and authorizing the entities that access the system. The service provided by the TWS-AA component is based on SAML assertions, enabling single sign-on (SSO) and federated identity management of other domains (between users, Web services and applications).

The service is based on a Secure Token Service (STS) accessing other services (excluding authentication and authorization services) available in the platform. In the case of federated systems, secure tokens can also be used outside the platform stating the owner's privileges.

STS is based on X.509 and SAML (Security Assertion Markup Language) as defined by OASIS.

Using the authentication token means that:

- Single sign-on (SSO) is available in the entire TrustedX platform. This may also be used for external services in accordance with OASIS Web Services Security.
- For authentication, an end-to-end Web services security model is implemented without completely relegating security to the transport mechanism. This simplicity is essential for building an inter-domain or federated mechanism. It is also possible to delegate other security services such as integrity and confidentiality to the used transport protocol, generally SSL/TLS.

The system recognizes multiple authentication mechanisms and implements several of them: authentication support through username and password, SSL/TLS and digital signatures. Additional mechanisms can be included in the platform via integration with external agents.

Although presented here as one service component, this component behaves as if it were two separate services: authentication and authorization.

The system also includes powerful group management features that facilitate the easy set-up of consumer communities. The following is a list of grouping features and possibilities provided by the system:

- Separate management of users, applications and Web services.

[1] *In this evaluation, an approved FIPS 140-2 level3 HSM has been used.*

- Management of static groups of entities—groups defined by the number of entities in a group.
- Management of organizational groups where the definition is a condition on the value of the distinguished name's attributes of any entity that belongs to the group (organization, organizational unit, place, country, and domain component).
- Management of dynamic groups defined through an expression that sets the condition of group membership. This expression may be constructed as:
    - An X.509 template that defines the condition complied with by any entity that is a member of the group.
    - An XPath expression (query) that defines the condition complied with by the XML view of the data registered in the system for any member entity of the group.
- Management of groups of groups, which supports the implementation of a role-based access control system, in which each role is defined as a specific group of groups.

Moreover, group management supports the execution of an open RBAC (role-based access control) system, where each role is mapped to a group or a group of groups. Using the RBAC feature, the system incorporates default privileged user roles for administering the system.

The TOE **supports any authentication mechanisms**, and **implements some predefined mechanisms** (specifically, one-factor and two-factor authentication mechanisms). The following authentication mechanisms are implemented:

- Authentication of end entities via sending credentials in SOAP messages (following the [SOAPServicesSec] standard)
    - Username and password (over a SSL/TLS channel)
    - X.509 PKI-based signature of SOAP messages (XML-DSig WS-Security) (over a SSL/TLS channel)
- Authentication of end entities sending credentials at the transport level
    - Authentication based on a X.509 PKI-based certificate (over a SSL/TLS channel)
- Authentication of authentication agents sending credentials in a proprietary SOAP message
    - Authentication based on a HMAC cryptographic function (over a SSL/TLS channel)

All these authentication mechanisms are completely implemented in the TOE since they are executed with a proof-of–possession check, i.e., a check that the entity being authenticated has a secret, namely, a password or a PKI secret key component.

The system also supports additional authentication mechanisms that it recognizes to be defined, which, in turn, must be implemented in the corresponding **authentication**

agents. In this way, the TOE also provides support for other forms of PKI-based authentication mechanisms. These, however, are not completely implemented in the platform, and need external agents to implement the proof-of-possession check.

The TOE includes the implemented authentication mechanisms and the authentication protocol with the additional authentication mechanisms that can be added subsequently.

1.3.2.2. TrustedX Entity Profiler (TWS-EP)

TrustedX TWS-EP provides a uniform information model based on XML for all platform objects and entities, completely masking formats (XML, ASN.1, tables, etc.), information sources (SQL, LDAP, files, etc.), locations (Intranet, Extranet, WAN, etc.), etc. It, therefore, offers entity information registration, retrieval and modification functions, particularly regarding identity, configuration and audit data.

In the TrustedX platform, all data is viewed as an XML virtual superstructure in which an XPath expression is used to access any of the values. This mechanism greatly simplifies the construction and administration of the entire system since the same data schema is used regardless of data location, type of repository (databases, LDAP, files, etc.), data format, etc.

Entity identity information in TrustedX is based on the Liberty Alliance specifications on identity information services (ID-SIS Personal Profile Service Specification and ID-SIS Employee Profile Service Specification). Information profiles in TrustedX cater for all users—individuals and applications.

Given a specific identity and verified during authentication, an abstract service is created that can be trusted to:

- Obtain information on a specific user (person or application) that is registered in the system, such as an e-mail or postal address.
- Obtain information on the configuration data used by an application (for example, KeyOne Desktop) for a specific user.
- Obtain the customized log and audit information (for example, to generate custom reports).

TrustedX TWS-EP also offers a UDDI (Universal Description, Discovery and Integration) based query service on WS entities. More specifically, it provides the platform with a location or binding information service for Web services.

This service aims to emulate the Liberty XML schemas, expanding them where necessary, but under no circumstances does it aim to implement a complete identity service as described in the Liberty specifications. However, in the future, and imagining a hypothetical federated environment based on Liberty specifications, it will be possible for TrustedX to obtain information on repository information profiles according to Liberty specifications.

1.3.2.3. TrustedX Digital Signature (TWS-DS)

The TrustedX TWS-DS is a remote service for digitally signing data. The interface of this service follows the OASIS Digital Signature Service (DSS) specification. More specifically, a series of profiles for the most-commonly used scenarios has been defined to simplify client integration and interoperability.

This service component is neutral from the point of view of the signer (or entity that requires the signature) since any entity, once authenticated and authorized, can request the signature service by providing a key identifier or selector it wants to use. The platform stores the entity signature material in repositories, making it accessible in a uniform, controlled manner through TWS-EP. The objects that contain keys and certificates are either software or hardware self-protected.

The defined profiles are based on the type of signature to be performed and are as follows:

- PKCS #7 and CMS signatures

  This profile supports generating digital signatures as per RSA PKCS #7, IETF CMS and ETSI CAdES standards.

  Single and multiple signatures (sequential or parallel) are supported in enveloped or detached signature format.

- XML-DSig/XAdES signatures

  This profile comprises XML-DSig and XAdES format signatures defined by W3C and ETSI. XAdES elements used are basic signer policies and properties.

  Enveloped, enveloping and detached signatures may be produced, including signatures by reference at any node of an XML document.

- S/MIMEv2 and S/MIMEv3 signatures

  This profile supports generating secure e-mail messages as per the S/MIME formats defined by IETF.

- WS-Security signatures

  This profile supports generating secure SOAP messages as per the WS-Security formats defined by OASIS.

- PDF Signature

  This profile supports generating signed Adobe PDF documents as per IETF PDF Signature recommendations.

1.3.2.4. TrustedX Digital Non-Repudiation (TWS-DR)

When a digital signature is generated, the signer does not incorporate evidences in the document that grant the probative value of this signature. This digital evidence is usually picked up automatically during the verification process for each digital signature; however, it is also possible to introduce evidence data during the generation process. To perform later verifications of the signatures, the evidences are

archived as fundamental data that can later be extracted and used by third parties as probative elements.

Digital evidences include information on the moment when the signature is produced, all the certificates that make up the trust chain and reliable information on the status of the certificates at that time. In the TrustedX platform, it is the non-repudiation service (TWS-DR) that is responsible for incorporating such evidences in signed documents.

TrustedX TWS-DR is a service that completes signatures (performed, for example, by the TWS-DS) with non-repudiable information adding a time-stamp, validation chain certificates and/or certificate status information.

On the one hand, this means that the service checks whether the certificates used are recognized by the platform. On the other hand, the digital signature that is generated includes validity evidences to prevent its repudiation. The maintenance and custody of these evidences is performed by another service that is in charge of requesting their custody and update before the keys and cryptographic material become vulnerable.

TWS-DR adds the following evidences to a previously-generated signature:

- A time-stamp issued by a trusted third party, a TSA (time stamping authority) is included in the signature. The time-stamp ensures that both the document's original data and the status token of certificates were generated before a specific date. The time-stamp format follows the standard defined in IETF TSP.
- Revocation Information: A token ensuring that the signature certificate that is included is valid. This token is generated by a trusted third party, a VA (validation authority) or a CA (certification authority). It takes the form of an OCSP response or a CRL object respectively.

TWS-DR uses the services of trusted third parties (TTPs) through TSP (Time-Stamp Protocol) and OCSP (Online Certificate Status Protocol) or any other transport mechanism to access CRL objects (commonly HTTP).

The interface of this service follows the OASIS DSS (Digital Signature Service) specification and includes additional elements defined by ETSI XAdES. Two special profiles offer support for the non-repudiable signature:

- Non-Repudiation CMS Signature

  This profile supports the extension of digital signatures following the CMS standard specified by IETF and CAdES specified by ETSI. The non-repudiation evidences can be embedded in the CMS signature following IETF and ETSI recommendations.

  This profile extends the CMS or CAdES-BES basic profile of TWS-DS, forcing the use of time-stamps and revocation information and only accepting the CMS signature format (it is prohibited by PKCS#7).

- Non-Repudiation XML Signature

  This profile supports the extension of digital signatures following W3C's XML-DSig standard. The XML signature contains embedded non-repudiation evidences, firstly following the ETSI XAdES and later also W3C XAdES. This profile extends the

XML-DSig or XAdES-BES TWS-DS basic profile by including the use of time-stamps and revocation information using XAdES.

This service component can also be used for the renewal and update of the trusted elements (time-stamps and revocation information) to grant digital signatures long-term validity (long-term signatures).

Moreover, TWS-DR can be extended with a digital signature custody service through TWS-DSC (Data Signature Custody).

1.3.2.5. TrustedX Digital Signature Verification (TWS-DSV)

The digital signature verification service is responsible for the verification of digital signatures. It verifies the validity of all signatures formats generated by the digital signature service (TWS-DS) and those updated by the non-repudiation service (TWS-DR).

TWS-DSV uses the services of a trusted third party (TTP) via the OCSP (Online Certificate Status Protocol). It may also connect to the validation authority (for example, KeyOne VA) that is responsible for the online validation of the status of the certificates included in the signature, thereby providing direct access to the different types of revocation information sources, e.g., direct access to databases or to CRLs published online.

The TWS-DSV interface follows the OASIS DSS (Digital Signature Service) specification. The following profiles are supported:

- PKCS #7 and CMS Signatures

  This profile supports the verification of digital signatures that follow the RSA PKCS#7, IETF CMS and ETSI CAdES standards.

  This profile is used to verify the signatures included in a document (multiple signatures). If the signatures also include time-stamps, these are also verified, along with the other electronic evidences present.

- XML-DSig and XAdES Signatures

  This profile comprises XML format signatures defined in XML-DSig and XAdES by W3C and ETSI.

  This profile is used to verify all types of signatures (enveloping, embedded and detached) and any time-stamps or electronic evidences included in the document.

- S/MIME Signatures

  This profile supports the verification of secure e-mail messages as per the S/MIME formats defined by IETF.

- WS-Security signatures

  This profile supports the verification of secure SOAP messages as per the WS-Security formats defined by OASIS.

- PDF Signatures

This profile supports the verification of signed Adobe PDF documents according to IETF PDF Signature recommendations.

1.3.2.6. TrustedX Digital Encryption (TWS-DE)

TrustedX TWS-DE is a service component that supports the encryption and decryption of data as per IETF CMS, RSA PKCS #7 and S/MIME formats and the W3C XML-Enc XML encryption standard.

The component uses the TWS-EP services to obtain the encryption certificates of recipients.

The TWS-DE interface follows a proprietary specification partly based on the W3C XML-Enc and OASIS DSS standards. The following profiles are supported:

- PKCS #7 and CMS Encryption
- S/MIME Encryption
- WS-Security Encryption
- XML-Enc Encryption

1.3.2.7. TrustedX Data Signature Custody (TWS-DSC)

As described above, owing to the fact that algorithms, keys and other cryptographic data can become vulnerable over time, evidences are considered to be temporary and, consequently, their renovation is necessary. Therefore, to maintain the non-repudiation properties, the validity of electronic evidences must be reviewed periodically in an automated manner. In the TrustedX platform, the responsibility of applying a renewal policy to the electronic evidences falls upon the signature data custody service (TWS-DSC); the TWS-DSC service is responsible for maintaining digital signature properties during the time periods set out by corporate rules and/or any applicable legislative framework.

This service protects and maintains the validation data of digital signatures (long-term signatures) by periodically requesting the TWS-DR to update this cryptographic material. The TWS-DSC requests the renewal of electronic evidences prior to time-stamp expiry or the algorithms, keys or other cryptographic data used to build different signature formats becoming vulnerable. This renewal is obtained by temporarily time-stamping the signature and the evidence and adding certificate information and their status. This process is repeated when the protection that is used for temporarily time-stamping the evidence becomes vulnerable.

Documents and their signatures, managed by TWS-DSC, are stored in a document management system (DMS). DMSs consist of services for organizing electronic documents that manage content, enable access control to documents and their properties, route documents, and manage work-flow tasks. DMSs provide functions for storing, searching and obtaining information on the life-cycle and revisions of the documents.

Integrating a DMS in TrustedX has two advantages:

- It encapsulates the real location of the documents by virtualizing the repository, thereby providing optimum functionality with maximum simplicity.

- The level of management provided is superior due to the fact that when the digital signature is integrated in the DMS, the graphical administration console and all the DMS management tools can be used natively.

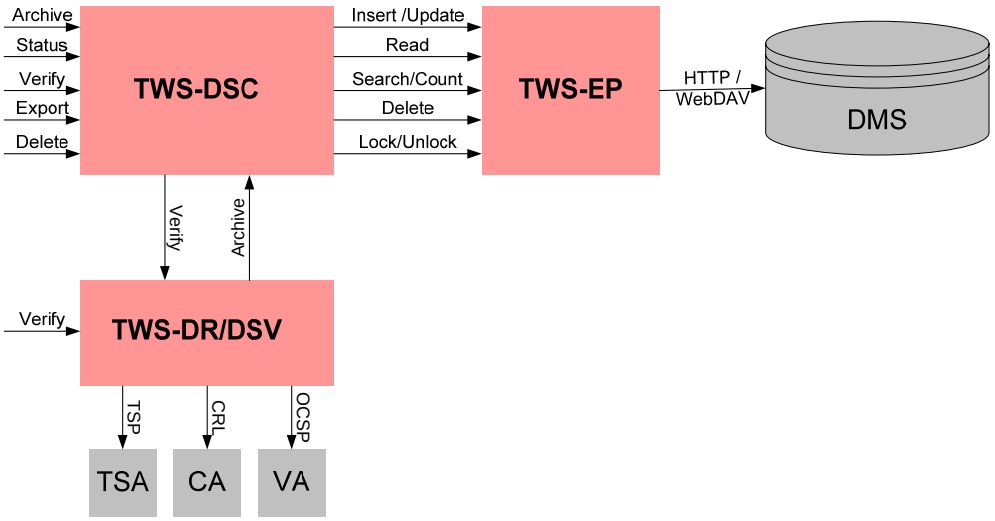

Figure 1-2. Interrelations between the TOE components and the DMS

Figure 1-2 depicts a simple architecture detailing the different interrelations between the TOE components (TWS-EP, TWS-DR and TWS-DSC), the trusted third parties (TSA, VA and CA) and the documental management system (DMS):

- The TWS-EP component (Entity Profiler) performs the XML broker functions on the different repositories (DMS/WebDAV, DB, etc.). In this case, it accesses a DMS repository via the HTTP/WebDAV interface to implement the insert/update, read and delete operations.

- The TWS-DR and TWS-DSV components provide the digital signature non-repudiation service and therefore implement the long-term signature formats (for advanced CAdES and XAdES formats) including the file format (ES-A).

As an added value, with the presence of the TWS-DSC custody component, it is possible to indicate to the TWS-DR component that it must require the archiving of a signature following the creation of a file signature.

Although not illustrated in the figure, note that all the TWS-DSC services (archive, verify, export, etc.) are subject to access control by the TWS-AA (Authentication and Authorization) component.

This service could also be used by an application such as Safelayer's KeyOne Desktop or by other services and applications to access the signature verification data to verify a specific digital signature.

1.3.2.8. TrustedX Key Management (TWS-KM)

With the key management service (also called KM) you can manage the keystores of the TrustedX users without having to use the graphic console to insert each user

certificate manually. This service is particularly suitable for organizations with a large number of TrustedX users.

The TWS-KM component is based on the W3C XKMS (XML Key Management Specification) [XKMS]. It implements a part of the KRS (Key Registration Service) functionality and protocol.

Through the use of XML, using the key management service, a user with sufficient privileges can perform the following actions on the keystores of TrustedX users and applications:

- Insert root certificates.
- Insert non-root certificates.
- Generate a key pair and an associated certification request.

The user can perform these actions with an on-disk keystore or a keystore based on a HSM device.

1.3.3. Administration and User Interface

The administration tasks for the trusted services platform can be performed by using one of the following methods or a combination of them:

- Using the administration console incorporated in the TrustedX platform. The supplied console supports performing the administration tasks for the whole platform—managing groups, trusted entities and policies and supervising events, etc.
- Using any specific application and via the platform's TWS-EP component. The fact that platform data is expressed in XML and that it is available as a service means that management applications can access data such as events and specific configurations. For example, the billing application used to perform the accounting of the service consumption of a specific entity.

1.3.3.1. Administration Model

The uniform information model of the TrustedX platform offers a conceptually simple administration procedure since it is as straightforward as reading and writing certain values of the virtual XML document.

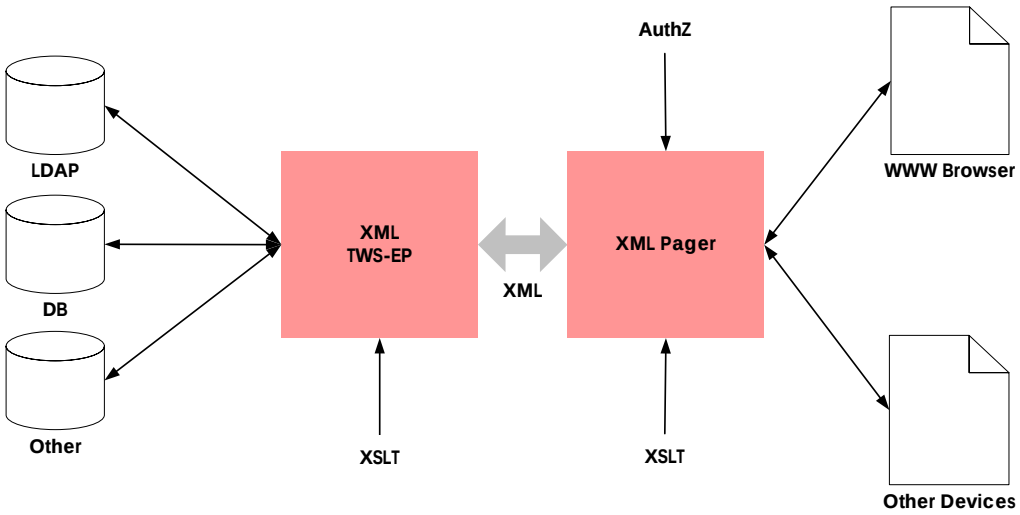

Figure 1-3. TOE GUI model

Figure 1-3 shows how the platform's information model is built. The TWS-EP component uses XML style sheets (XSLT) to virtually transform different information sources (LDAP, DB, etc.) into a single XML document. Once access authorization is granted, further XSLT transformations are applied to obtain pages directly represented in a Web browser or other devices.

This approach presents major advantages in terms of simplified system management in its ability to support different representation devices owing to its open and easily-extended nature. The exclusive use of XML, XPath and XSLT technologies supports the maximum degree of resource reuse and system extension by using the numerous XML products available on the market.

1.3.3.2. Administration Console

TrustedX administration console follows the above-mentioned model. The system has an advanced graphical user interface (GUI) for administering and accessing all the system information in a uniform and centralized manner via a Web browser.

The administration functions of the platform's console provide for:

- End-entity management: management of groups of privileged users, users, applications or services and the groups of end entities.
- Management of trusted entities: management of certification, validation and time-stamp authorities.
- Management of authentication and authorization policies: for defining a set of rules and actions that are applied depending on the type of authentication, authenticated entity and resource requested.
- Management of digital signature generation policies: for defining and making changes to the policies applied for generating digital signatures.
- Management of digital signature verification policies: for defining and managing the digital signature verification policies, including the digital certificate validation policies.

- System configuration management: for defining the configuration of the platform's own service components and the configuration of the databases, the directory, etc.
- Management of logs and audits: for browsing all the events generated by all the platform's service components.

1.3.4. TrustedX Security Policy

The TrustedX security policy is a set of configuration restrictions that must always be met. Thus, each time the TOE's services are started, the system verifies if the configuration satisfies the conditions enforced by the policy; where it does not, it aborts the starting and prevents service operation.

I.e., if, for any reason, the TOE configuration reaches a state that is incompatible with the security policy, the services are not allowed to start. Therefore, the system can never run under a configuration that violates the security policy. The security policy can be set (i.e., frozen). If it is, changes cannot be made to the TOE. From this moment on, the TOE configuration is permanently restricted by the security policy defined at the time of the freezing.

One of the security policies included in the TOE is the EAL4+ Security Policy. To guarantee the security conditions and the functional requirements included in this Security Target, it is necessary to fix this security policy.

1.3.4.1. Security Policy Parameters

The security policy comprises:

- Restrictions on the Configuration that Affects Entity Authentication
- Restrictions on Certificate Validation
- Restrictions on the Configuration that Affects the Keystores
- Restrictions on the Configuration Affecting Usages Required of Keys
- Restrictions on the Configuration of the Shell
- Miscellaneous Restrictions

Restrictions on the Configuration that Affects Entity Authentication

For a security policy that complies with EAL4+ restrictions, the following requirements are guaranteed:

- The allowed authentication agent list cannot be left empty.
- The allowed authentication mechanisms parameter cannot be left empty, and this parameter cannot contain the "anonymous" mechanism.
- The list of allowed SAML tokens can only contain the Signed Assertion type, or "any type of SAML token allowed" must be set.
- The maximum session length parameter cannot be left empty.

Restrictions on Certificate Validation

For a security policy that complies with EAL4+ restrictions, the following requirements are guaranteed:

- Inclusion of complete information on the signer certificate (for verifying a signature) or the validated certificate (for validating a certificate) in service responses.
- All certificate validation rules must always validate the complete certification chain of all the certificates.
- All certificate validation rules must define at least one certificate validation mechanism (OCSP or CRL).
- All signature generation and data encryption and decryption rules must always require prior validation of the certificate of the signer or data recipient.

Restrictions on the Configuration that Affects the Keystores

For a security policy that complies with EAL4+ restrictions, the following requirements are guaranteed:

- Only end entities can manage their keystores (i.e., security officers cannot access them).
- An HSM device must be used for user keystores.
- An HSM device is required for cryptographic operations in the system.
- Only HSM modules configured to comply with FIPS 140-2 level 3 requirements can be registered.

Restrictions on the Configuration Affecting Usages Required of Keys

For a security policy that complies with EAL4+ restrictions, a set of requirements on the key usages that the service policies must establish for the different certificates involved in each case has been defined.

Restrictions on the Configuration of the Shell

For a security policy that complies with EAL4+ restrictions, a root user cannot be enabled, and some shell commands are prohibited.

Miscellaneous Restrictions

For a security policy that complies with EAL4+ restrictions, the following requirements are guaranteed:

- The TOE rejects all certificates that contain a critical extension that it cannot process.
- The TOE includes extended audit information in the log records.

- The use of the NTP protocol is required to synchronize the TOE with the system clock.

1.3.5. Environment Components

The environment components selected for the evaluation of this product are the following:

- Operating System[2]: Red Hat Enterprise Linux Version 5.3 with the Tomcat 6.0.26 Web server and the JBoss 5.1.0 application server.
- Databases: Any DBMS supporting a well-defined JDBC interface with SSL/TLS support, for instance, Microsoft SQL Server or Oracle.
- Hardware Security Module: ncipher nShield F3 2000 for netHSM (FIPS 140-2 level 3 hardware cryptographic module).
- Document Management System: Any documental server (DMS/ECM) accessible by means of an access interface based on HTTP/WebDAV defined for TrustedX, for instance, Oracle Content Database.
- Directory: Any directory compliant with the RFC 4511 ("Lightweight Directory Access Protocol (LDAP): The Protocol") with SSL/TLS support, for instance, SunOne Directory Server.
- Optionally (only if the services of a time stamping authority are required): any TSA compliant with the TSP protocol (see [RFC3161]).
- For the Hardware Appliance Edition, it is necessary to use computers[3] with the following characteristics: A processor with x86-64 compatible architecture, two network interfaces (minimum) and a DVD reader.
- Clients making requests to the TrustedX Web services: Any that correctly validates the certificates for an SSL/TLS connection.

1.3.6. Annex III and Annex IV of the [EUROPEAN_DIRECTIVE]

This Security Target conforms to the security functional requirements included in the [PKE_PP] Protection Profile. Additionally, a set of new Common Criteria elements (threats, objectives and requirements) have been included. The new security objectives have been derived, in terms of Common Criteria terminology, from the points included in Annex III "Requirements for secure signature-creation devices" and Annex IV "Recommendations for secure signature verification" of the "Directive 1999/93/EC of the European Parliament and of the Council" of 13 December 1999 on a Community framework for electronic signatures ([EUROPEAN_DIRECTIVE]).

---

[2] *Component as delivered with the TrustedX distribution.*
[3] *All the components must appear in the hardware compatibility list of Red Hat Enterprise Linux 5.3 x86_64.*

The extended security functional requirements related to these new security objectives are included in the Table 5-3. Security Functional Requirements for the TOE derived from the [EUROPEAN_DIRECTIVE] of Security Requirements. The rationale for how the product complies with the functional requirements associated to these new security objectives can be found in Rationale for the security objectives derived from Annex III of the [EUROPEAN_DIRECTIVE] in TOE Summary Specification.

1.3.6.1. Annex III of the [EUROPEAN_DIRECTIVE]

Annex III of the [EUROPEAN_DIRECTIVE] includes several requirements on the generation of digital signatures. These requirements can be grouped and summarized as follows:

Secrecy of the user private keys, robustness of the cryptographic algorithms and security of the generated signatures (point 1.(a)[4] and point 1.(b)[5])

The TrustedX services work with approved FIPS 140-2 level 3 and PKCS11-compatible cryptographic modules as hardware cryptographic modules for performing cryptographic functions.

With regard to the protection of the user private keys, a **FIPS 140-2 level 3 approved HSM is required**. Security of the cryptography and therefore of the sensitive data to which this cryptography is applied is based on the robustness of FIPS 140-2 level 3 security requirements that these cryptographic modules fulfill.

The use of this FIPS 140-2 level 3 approved HSM supports protecting user keys via tamper-evident physical security mechanisms and preventing intruders from gaining access to critical security parameters held in the cryptographic module.

The TOE detects if the cryptographic module that protects the user private keys is configured as a FIPS 140-2 level 3 device. If it is not, TrustedX services are blocked until the required FIPS configuration is achieved.

Exclusive use of the private key by the user (point 1.(c)[6])

The TOE guarantees the exclusive use of the private key by the user. Basically, the TOE has several security mechanisms for protecting the sensitive information of users from illegitimate users.

One important mechanism from the security point of view that helps in the implementation of this requirement is the access control the TOE requires for using the private key.

---

[4] *From the Annex III of the [EUROPEAN_DIRECTIVE]: "The signature-creation-data used for signature generation can practically occur only once, and their secrecy is reasonably assured."*
[5] *From the Annex III of the [EUROPEAN_DIRECTIVE]: "The signature-creation-data used for signature generation cannot, with reasonable, be derived and the signature is protected against forgery using currently available technology."*
[6] *From the Annex III of the [EUROPEAN_DIRECTIVE]: "The signature-creation-data used for signature generation can be reliably protected by the legitimate signatory against the use of others."*

The TOE guarantees that the signature key cannot be used until the holder is identified and authenticated. The authentication mechanism is any authentication requirement that can be imposed on an electronic signature product.

The TOE supports several authentication mechanisms and comes with a set of them (specifically, single- and multiple-factor authentication mechanisms). The system also supports additional (not included in this TOE) authentication mechanisms that it recognizes to be defined, which, in turn, must be implemented in the corresponding authentication agents.

For each authentication mechanisms, the TOE assigns an authentication level (i.e., lower for single-factor authentication and higher for multiple-factor authentication). This allows defining the minimum authentication mechanism level required for the operations that require the use of a private key (e.g., cryptographic smart card containing PIN-protected RSA keys).

In the TOE, it is possible to define a specific authentication level for a particular TrustedX service (e.g., digital signature service or key management service) and to relate it to a specific operation of this service (e.g., digital signature operation of the TWS-DS service or key pair generation of the TWS-KM service).

In the above scenario, this multiple-factor authentication mechanism should be associated to a high or very high authentication level, and the selected authentication level should be associated to certain operations of the TWS-DS and TWS-KM services (such as the digital signature generation operation or the key pair generation operation).

Finally, the TOE can be defined to require authentication for each private key operation or to allow the use of the private key within a certain timeframe. This allows TrustedX to be used for bulk/batch signature purposes.

See Authentication Mechanisms for more information on the access control applied to the use of private keys.

The TOE detects if the cryptographic module is configured to be FIPS 140-2 level 3 compliant. Where it is, this guarantees a controlled access to the private key from the cryptographic device environment.

Integrity of the data to be signed (point 2[7])

Any communication from/to the TOE is protected by the use of security mechanisms, such as the SSL/TLS security protocols. The TOE provides technical mechanisms to assure this integrity in the path between the user requesting the signature service and the TOE.

1.3.6.2. Annex IV of the [EUROPEAN_DIRECTIVE]

Annex IV of the [EUROPEAN_DIRECTIVE] includes several requirements regarding digital signature verification. Basically, these requirements can be summarized as follows:

[7] *From the Annex III of the [EUROPEAN_DIRECTIVE]: "Secure signature-creation devices must not alter the data to be signed or prevent such data from being presented to the signatory prior to the signature process".*

- Integrity of the data to be validated, integrity of the data used for verifying a digital signature and integrity of the result of the signature validation.
- Robustness of the digital signature validation process.
- Logging of security-relevant events.

The integrity of the data exchanged between a user and the TOE's services, in a digital signature verification process, is protected by the use of security mechanisms such as the SSL/TLS security protocols.

Regarding the robustness of the cryptographic algorithms (in the digital signature validation processes), a FIPS 140-2 level 3 approved HSM is required. Security of these algorithms is based on the robustness of FIPS 140-2 level 3 security requirements that these cryptographic modules fulfill. The TOE is able to detect if the cryptographic module that protects the user private keys is configured as a FIPS 140-2 level 3 device; where it is not, the product is blocked.

The system logs all security-relevant operations performed by the services (e.g., starting and ending sessions, modifying the system configuration, using a resource). It also provides statistics on the system and on the use of the TrustedX services.

1.3.6.3. TrustedX: Server-Based Signature Services Technology

The TOE's technology features a server-based signature services architecture. This means that signature creation data is not stored in a signature creation device at the signer's location, but in a central hardware security module at the signature service provider's location. The use of this architecture is compatible with the functionality of creating advanced or qualified electronic signatures**.**

There are forums, studies and legislation that ratify this assertion, such as:

- Public Statement on Server Based Signature Services. Forum of European Supervisory Authorities for Electronic Signatures (FESA). October 17, 2005. [FESA].
- EU Electronic Signature Legislation Requirements for DSS. OASIS. December 22, 2008. [LEGISLATION_DSS_OASIS].

1.3.6.4. Compliance with the Legislation of EU Member States

The EU Directive has been transposed into the legislation of the different EU member states. Therefore, all the requirements for the signature devices included in the European Directive are part of the legal requirements of EU countries.

CHAPTER 2

2. Conformance Claims

The present Security Target conforms to the following assurance and functional requirements:

- Functional Requirements of the "Part 2: Security functional components" of the Common Criteria Standard. February 2009, Version 3.1, Revision 2.
- Functional Requirements of the Part 2 extended of the Common Criteria Standard. February 2009, Version 3.1, Revision 2.
- Assurance Requirements of the "Part 3: Security assurance components". February 2009, Version 3.1, Revision 2, for the **EAL4 Common Criteria certification level, augmented with ALC_FLR.2**.

The present Security Target conforms to the following **Protection Profile**:

- "**U.S. Government Basic Robustness PKE PP with the packages listed bellow at EAL4 with ALC_FLR.2 augmentation**", sponsored by United States Marine Corps (USMC), dated May 2007, version 2.8.

The TOE defined in this Security Target conforms to the following Packages included in the [PKE_PP] Protection Profile:

- Certification Path Validation (CPV) – Basic
- CPV – Basic Policy
- PKI Signature Generation
- PKI Signature Verification
- PKI Encryption using Key Transfer Algorithms
- PKI Decryption using Key Transfer Algorithms
- PKI Based Entity Authentication
- Online Certificate Status Protocol Client
- Certificate Revocation List (CRL) Validation
- Audit

- Continuous Authentication

CHAPTER 3

3. Security Problem Definition

This section includes the following:

- Secure usage assumptions
- Threats, and
- Organizational security policies

This information provides the basis for the Security Objectives specified in chapter 4 Security Objectives, and for the Security Functional Requirements for the TOE, Security Functional Requirements for the Environment and for the TOE Security Assurance Requirements specified in chapter 5 Security Requirements.

3.1. Secure Usage Assumptions

Table 3-1. Assumptions for the IT Environment lists the Secure Usage Assumptions for the IT environment.

Assumption Name	Description
A.Configuration	The TOE will be properly installed and configured.
A.Enhanced-Basic	The attack potential on the TOE is assumed to be "Enhanced-Basic".
A.NO_EVIL	Administrators are non-hostile, appropriately trained and follow all administrator guidance.
A.PHYSICAL	It is assumed that the environment provides the TOE with appropriate physical security, commensurate with the value of the IT assets protected by the TOE.

Table 3-1. Assumptions for the IT Environment

Note about the A.NO_EVIL assumption

The A.NO_EVIL assumption is defined as follow: "Administrators are non-hostile, appropriately trained and follow all administrator guidance."

The "administrator" user referred to by this assumption is a user who has one of the following user profiles:

- Administrator through the Command Shell Role.
- Is in the Console Administrator and First Console Administrator user groups (privileged groups).
- Is in the Security Officer and First Security Officer user groups (privileged groups).
- Administrators of the third-party components used.

See Annex III and Annex IV of the [EUROPEAN_DIRECTIVE] for more information on roles and user groups.

Note that to maintain the guarantees of EAL4+ certification, only the permissions included in the Table 8-1. Table of permissions of the privileged users for the user profiles considered in the A.NO_EVIL assumption are assumed.

3.2. Threats

This subsection defines the base threats to the TOE, included in Table 3-2. Base Threats to Security below. The asset under attack is the information transiting the TOE. In general, the threat agent includes, but is not limited to: 1) people with TOE access who are expected to possess "average" expertise, few resources, and moderate motivation, or 2) failure of the TOE.

Threat Name	Description
T.AUDIT_COMPROMISE	A user or process may view audit records, cause audit records to be lost or modified, or prevent future audit records from being recorded, thus masking a user's action.
T.CHANGE_TIME	An unauthorized user may change the TSF notion of time resulting in accepting old revocation information or expired certificates.
T.CRYPTO_COMPROMISE	A user or process may cause key, data or executable code associated with the cryptographic functionality to be inappropriately accessed (viewed, modified, or deleted), thus compromising the cryptographic mechanisms and the data protected by those mechanisms.

T.MASQUERADE	A user or process may masquerade as another entity in order to gain unauthorized access to data or TOE resources.
T.POOR_TEST	Lack of or insufficient tests to demonstrate that all TOE security functions operate correctly (including in a fielded TOE) may result in incorrect TOE behavior being undiscovered thereby causing potential security vulnerabilities.
T.RESIDUAL_DATA	A user or process may gain unauthorized access to data through reallocation of TOE resources from one user or process to another.
T.TSF_COMPROMISE	A user or process may cause, through an unsophisticated attack, TSF data, security attributes, or executable code to be inappropriately accessed (viewed, modified, or deleted).
T.UNATTENDED_SESSION	A user may gain unauthorized access to an unattended session.
T.UNAUTHORIZED_ACCESS	A user may gain access to user data for which they are not authorized according to the TOE security policy.
T.UNIDENTIFIED_ACTIONS	The administrator may not have the ability to notice potential security violations, thus limiting the administrator's ability to identify and take action against a possible security breach.

Table 3-2. Base Threats to Security

Table 3-3. Base Threats for the CPV – Basic Package include the security threats for the Certification Patch Validation – Basic Package.

Threat Name	**Description**
T.Certificate_Modi	An untrusted user may modify a certificate resulting in using a wrong public key.
T.DOS_CPV_Basic	The revocation information or access to revocation information could be made unavailable, resulting in loss of system availability.

T.Expired_Certificate	An expired (and possibly revoked) certificate as of TOI could be used for signature verification.
T.Untrusted_CA	An untrusted entity (Certification Authority (CA)) may issue certificates to bogus entities, permitting those entities to assume identity of other legitimate users.
T.No_Crypto	The user public key and related information may not be available to carry out the cryptographic function.
T.Path_Not_Found	A valid certification path is not found due to lack of system functionality.
T.Revoked_Certificate	A revoked certificate could be used as valid, resulting in security compromise.
T.User_CA	A user could act as a CA, issuing unauthorized certificates

Table 3-3. Base Threats for the CPV – Basic Package

Table 3-4. Threats for the CPV – Basic Policy Package include the security threats for the Certification Patch Validation – Basic Policy Package.

Threat Name	**Description**
T.Unknown_Policies	The user may not know the policies under which a certificate was issued.

Table 3-4. Threats for the CPV – Basic Policy Package

Table 3-5. Threats for the PKI Signature Generation Package include the security threats for the PKI Signature Generation Package.

Threat Name	**Description**
T.Clueless_PKI_Sig	The user may try only inappropriate certificates for signature verification because the signature does not include a hint.

Table 3-5. Threats for the PKI Signature Generation Package

Table 3-6. Threats for the PKI Signature Verification Package include the security threats for the PKI Signature Verification Package.

Threat Name	Description
T.Assumed_Identity_PKI_Ver	A user may assume the identity of another user in order to verify a PKI signature.
T.Clueless_PKI_Ver	The user may try only inappropriate certificates for signature verification because hints in the signature are ignored.

Table 3-6. Threats for the PKI Signature Verification Package

Table 3-7. Threats for the PKI Encryption using Key Transfer Algorithms Package include the security threats for the PKI Encryption using Key Transfer Algorithms Package.

Threat Name	Description
T.Assumed_Identity_WO_En	A user may assume the identity of another user in order to perform encryption using Key Transfer algorithms.
T.Clueless_WO_En	The user may try only inappropriate certificates for encryption using Key Transfer algorithms in absence of hint.

Table 3-7. Threats for the PKI Encryption using Key Transfer Algorithms Package

Table 3-8. Threats for the PKI Decryption using Key Transfer Algorithms Package include the security threats for the PKI Decryption using Key Transfer Algorithms Package.

Threat Name	Description
T.Garble_WO_De	The user may not apply the correct key transfer algorithm or private key, resulting in garbled data.

Table 3-8. Threats for the PKI Decryption using Key Transfer Algorithms Package

Table 3-9. Threats for the PKI Based Entity Authentication Package includes the security threats for the PKI Based Entity Authentication Package.

Threat Name	Description
T.Assumed_Identity_Auth	A user may assume the identity of another user to perform entity based authentication.
T.Replay_Entity	An unauthorized user may replay valid

	entity authentication data.

Table 3-9. Threats for the PKI Based Entity Authentication Package

Table 3-10. Threats for the Online Certificate Status Protocol Client Package include the security threats for the Online Certificate Status Protocol Client Package.

Threat Name	Description
T.DOS_OCSP	The OCSP response or access to the OCSP response could be made unavailable, resulting in loss of system availability.
T.Replay_OCSP_Info	The user may accept an OCSP response from well before TOI resulting in accepting a revoked certificate.
T.Wrong_OCSP_Info	The user may accept a revoked certificate or reject a valid certificate due to a wrong OCSP response.

Table 3-10. Threats for the Online Certificate Status Protocol Client Package

Table 3-11. Threats for the Certificate Revocation List (CRL) Validation Package include the security threats for the Certificate Revocation List (CRL) Validation Package.

Threat Name	Description
T.DOS_CRL	The CRL or access to CRL could be made unavailable, resulting in loss of system availability.
T.Replay_Revoc_Info_CRL	The user may accept a CRL issued well before TOI resulting in accepting a revoked certificate.
T.Wrong_Revoc_Info_CRL	The user may accept a revoked certificate or reject a valid certificate due to a wrong CRL

Table 3-11. Threats for the Certificate Revocation List (CRL) Validation Package

Table 3-12. Threats for the Audit Package include the security threats for the Audit Package.

Threat Name	Description

T.PKE_Accountability	The PKE related audit events cannot be linked to individual actions.

Table 3-12. Threats for the Audit Package

Table 3-13. Threats for the Continuous Authentication Package include the security threats for the Continuous Authentication Package.

Threat Name	**Description**
T.Hijack	An unauthorized user may hijack an authenticated session.

Table 3-13. Threats for the Continuous Authentication Package

Table 3-14. Threats related to objectives that have been derived from the Annex III and Annex IV of the [EUROPEAN_DIRECTIVE] includes the new (regarding to the threats previously included) security threats that are related to security objectives that have been derived from the Annex III and Annex IV of the [EUROPEAN_DIRECTIVE].

Threat Name	**Description**
T.DISCLOSURE_ OR_NOT_UNIQUE_PRIVATE_KEYS	A private key is improperly disclosed or it can be obtained again (it is not assured that the signature-creation data used for signature generation can practically occur only once).
T.ILEGITIMATE_USE_OF_PRIVATE_KEYS	An attacker can access to the private keys of users, so that the signatures can be generated without the legitimate signatory consent.
T.DATA_TO_BE_SIGNED	An attacker modifies the data to be signed while in the process of being signed.
T.DATA_USED_FOR_VERIFYING_THE_SIGNATURE	The data used to verify the signature do not correspond to the data presented to the verifier.
T.RESULT_OF_THE_SIGNATURE_VALIDATION	The result of the validation of a signature does not correspond to the result presented to the verifier.
T.SIGNATORY'S_IDENTITY	The signatory's identity does not correspond to the result

	presented to the verifier.
T.SECURITY_AUDIT_EVENTS	The security-relevant changes are not detected.
T.INVALID_CERTIFICATE	The authenticity and validity of the certificate required at the time of signature verification are not reliably verified.
T.SIGNATURE_NOT_RELIABLY_VERIFIED	The signatures are not reliably verified.
T.SIGNATURE_FALSIFIED	The signatures can be falsified and the private keys can be derived using currently available technology.
T.SIGNED_DATA_FALSIFIED	The signed data established by the verifier can be falsified.
T.PSEUDONYM_NOT_SUPPORTED	It is not assured, at the time of the signature verification, that a pseudonym could be indicated.

Table 3-14. Threats related to objectives that have been derived from the Annex III and Annex IV of the [EUROPEAN_DIRECTIVE]

3.3. Organizational Security Policies

Table 3-15. Organizational Security Policies lists the Organizational Security Policies.

Policy Name	Description
P.ACCESS_BANNER	The IT Environment shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the system.
P.ACCOUNTABILITY	The authorized users of the TOE shall be held accountable for their actions within the TOE.
P.CRYPTOGRAPHY	Only NIST FIPS validated cryptography (methods and implementations) are acceptable for key management (i.e.; generation, access, distribution, destruction, handling, and storage of keys) and cryptographic services (i.e.; encryption, decryption, signature, hashing, key exchange, and random

	number generation services).

Table 3-15. Organizational Security Policies

CHAPTER 4

4. Security Objectives

This chapter 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 organizational security policies and assumptions.

4.1. Security Objectives for the Environment

The security objectives for the Environment are defined in Table 4-1. Security Objectives for the Environment below.

There are four security objectives for the non-IT environment of the TOE: OE.Configuration, OE.NO_EVIL, OE.PHYSICAL, and OE.Enhanced-Basic. The remaining objectives are for the IT environment.

Objective Name	Description
OE.AUDIT_GENERATION	The IT Environment will provide the capability to detect and create records of security-relevant events associated with users.
OE.AUDIT_PROTECTION	The IT Environment will provide the capability to protect audit information.
OE.AUDIT_REVIEW	The IT Environment will provide the capability to selectively view audit information.
OE.Configuration	The TOE will be installed and configured properly for starting up the TOE in a secure state.
OE.CORRECT_TSF_OPERATION	The IT Environment will provide the capability to test the TSF to ensure the correct operation of the TSF at a customer's site.
OE.CRYPTOGRAPHY	The TOE shall use NIST FIPS 140-2 validated cryptographic services

	provided by the IT Environment.
OE.DISPLAY_BANNER	The IT Environment will display an advisory warning regarding use of the TOE.
OE.Enhanced-Basic	The TOE will be designed and implemented for a minimum attack potential of "Enhanced-Basic" as validated by the vulnerability analysis.
OE.MANAGE	The IT Environment will provide all the functions and facilities necessary to support the administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.
OE.MEDIATE	The IT Environment will protect user data in accordance with its security policy.
OE.NO_EVIL	Sites using the TOE will ensure that administrators are non-hostile, appropriately trained and follow all administrator guidance.
OE.PHYSICAL	The non-IT environment will provide an acceptable level of physical security so that the TOE cannot be tampered with or be subject to side channel attacks such as the various forms of power analysis and timing analysis.
OE.RESIDUAL_INFORMATION	The IT Environment will ensure that any information contained in a protected resource within its Scope of Control is not released when the resource is reallocated.
OE.SELF_PROTECTION	The IT Environment will maintain a domain for its own execution that protects it and its resources from external interference, tampering, or unauthorized disclosure.
OE.TIME_STAMPS	The IT Environment will provide reliable time stamps and the capability for the administrator to set the time used for these time stamps.
OE.TIME_TOE	The IT Environment will provide reliable time for the TOE use.
OE.TOE_ACCESS	The IT Environment will provide mechanisms that control a user's logical

	access to the TOE.
OE.TOE_PROTECTION	The IT Environment will protect the TOE and TOE resources from external interference, tampering, or unauthorized disclosure and modification.

Table 4-1. Security Objectives for the Environment

4.2. Security Objectives for the TOE

The security objectives for the TOE are defined in this section.

The following security objectives included in Table 4-2. Security Objectives for CPV – Basic Package are defined for the Certification Path Validation – Basic Package.

Objective Name	**Description**
O.Availability	The TSF shall continue to provide security services even if revocation information is not available.
O.Correct_Temporal	The TSF shall provide accurate temporal validation results.
O.Current_Certificate	The TSF shall only accept certificates that are not expired as of TOI.
O.Get_KeyInfo	The TSF shall provide the user public key and related information in order to carry out cryptographic functions.
O.Path_Find	The TSF shall be able to find a certification path from a trust anchor to the subscriber.
O.Trusted_Keys	The TSF shall use trusted public keys in certification path validation.
O.User	The TSF shall only accept certificates issued by a CA.
O.Verified_Certificate	The TSF shall only accept certificates with verifiable signatures.
O.Valid_Certificate	The TSF shall use certificates that are valid, i.e., not revoked.

Table 4-2. Security Objectives for CPV – Basic Package

Objectives O.Availability and O.Valid_Certificate mitigate threats T.DOS_CPV_Basic and T.Revoked_Certificate, respectively. But these objectives cannot completely counter the threats simultaneously. The FDP_DAU_CPV_(EXP).1.3 requirement has

been configured in order to the Security Officer group (Web Services Consumer role)of the TrustedX could determine if the lack of availability of revocation information must been overridden. In this sense, depending on the configuration made by the Security Officer regarding the FDP_DAU_CPV_(EXT).1.3 requirement, it will be mitigate the O.Valid_Certificate or the T.Revoked_Certificate threat. If the Security Officer configures that the revocation status must be checked, then the T.Revoked_Certificate threat will be mitigated; else the T.DOS_CPV_Basic will be mitigated.

This note also applies to the O.DIRECTIVE_ANNEX_IV_PART_d. This objective mitigates the T.invalid_certificate. In this case, if the Security Officer configures that the revocation status must be checked, then the T.DIRECTIVE_ANNEX_IV_PART_d threat will be mitigated; else the T.DOS_CPV_Basic will be mitigated.

The following security objectives included in Table 4-3. Security Objectives for CPV – Basic Policy Package are defined for the Certification Path Validation – Basic Policy Package.

Objective Name	Description
O.Provide_Policy_Info	The TSF shall provide certificate policies for which the certification path is valid.

Table 4-3. Security Objectives for CPV – Basic Policy Package

The following security objectives included in Table 4-4. Security Objectives for PKI Signature Generation Package are defined for the PKI Signature Generation Package.

Objective Name	Description
O.Give_Sig_Hints	The TSF shall provide hints for selecting correct certificates for signature verification.

Table 4-4. Security Objectives for PKI Signature Generation Package

The following security objectives included in Table 4-5. Security Objectives for PKI Signature Verification Package are defined for the PKI Signature Verification Package.

Objective Name	Description
O.Use_Sig_Hints	The TSF shall use hints for selecting correct certificates for signature verification.
O.Linkage_Sig_Ver	The TSF shall use the correct user public key for signature verification.

Table 4-5. Security Objectives for PKI Signature Verification Package

The following security objectives included in Table 4-6. Security Objectives for PKI Encryption using Key Transfer Algorithms Package are defined for the PKI Encryption using Key Transfer Algorithms Package.

Objective Name	Description
O.Hints_Enc_WO	The TSF shall provide hints for selecting correct certificates or keys for PKI Encryption using Key Transfer Algorithms.
O.Linkage_Enc_WO	The TSF shall use the correct user public key for key transfer.

Table 4-6. Security Objectives for PKI Encryption using Key Transfer Algorithms Package

The following security objectives included in Table 4-7. Security Objectives for PKI Decryption using Key Transfer Algorithms Package are defined for the PKI Decryption using Key Transfer Algorithms Package.

Objective Name	Description
O.Correct_KT	The TSF shall use appropriate private key and key transfer algorithm.

Table 4-7. Security Objectives for PKI Decryption using Key Transfer Algorithms Package

The following security objectives included in Table 4-8. Security Objectives for PKI Based Entity Authentication Package are defined for the PKI Based Entity Authentication Package.

Objective Name	Description
O.I&A	The TSF shall uniquely identify all entities, and shall authenticate the claimed identify before granting an entity access to the TOE facilities.
O.Limit_Actions_Auth	The TSF shall restrict the actions an entity may perform before the TSF verifies the identity of the entity.
O.Linkage	The TSF shall use the correct user public key for authentication.
O.Single_Use_I&A	The TSF shall use the I&A mechanism that requires unique authentication information for each I&A.

Table 4-8. Security Objectives for PKI Based Entity Authentication Package

The following security objectives included in Table 4-9. Security Objectives for Online Certificate Status Protocol Client Package are defined for the Online Certificate Status Protocol Client Package.

Objective Name	Description
O.Accurate_OCSP_Info	The TSF shall accept only accurate OCSP responses.
O.Auth_OCSP_Info	The TSF shall accept the revocation information from an authorized source for OCSP transactions.
O.Current_OCSP_Info	The TSF accept only OCSP responses current as of TOI.
O.User_Override_Time_OCSP	The TSF shall permit the user to override the time checks on the OCSP response.

Table 4-9. Security Objectives for Online Certificate Status Protocol Client Package

Objectives O.Current_OCSP_Info and O.User_Override_Time_OCSP mitigate threats T.Replay_OCSP_Info and T.DOS_OCSP, respectively. But these objectives cannot completely counter the threats simultaneously.

To fully mitigate the threat T.Replay_OCSP, the request nonce has been used in the security functional requirements FDP_DAU_OCS_(EXP).1.12.

The following security objectives included in Table 4-10. Security Objectives for the Certificate Revocation List (CRL) Validation Package are defined for the Certificate Revocation List (CRL) Validation Package.

Objective Name	Description
O.Accurate_Rev_Info	The TSF shall accept only accurate revocation information.
O.Auth_Rev_Info	The TSF shall accept the revocation information from an authorized source for CRL.
O.Current_Rev_Info	The TSF shall accept only CRL that are current as of TOI.
O.User_Override_Time_CRL	The TSF shall permit the user to override the time checks on the CRL.

Table 4-10. Security Objectives for the Certificate Revocation List (CRL) Validation Package

Objectives O.Current_Rev_Info and O.User_Override_Time_CRL mitigate threats T.Replay_Revoc_Info_CRL and T.DOS_CRL, respectively. But these objectives cannot completely counter the threats simultaneously.

The following security objectives included in Table 4-11. Security Objectives for the Audit Package are defined for Audit Package.

Objective Name	Description
O.PKE_Audit	The TSF shall audit security relevant PKE events.

Table 4-11. Security Objectives for the Audit Package

The following security objectives included in Table 4-12. Security Objectives for the Continuous Authentication Package are defined for Continuous Authentication Package.

Objective Name	Description
O.Continuous_I&A	The TSF shall continuously authenticate the entity.

Table 4-12. Security Objectives for the Continuous Authentication Package

The following security objectives included in Table 4-13. Security Objectives derived from Annex III “Requirements for secure signature-creation devices” of [EUROPEAN_DIRECTIVE] have been derived from Annex III “Requirements for secure signature-creation devices” of the “Directive 1999/93/EC of the European Parliament and of the Council” of 13 December 1999 on a Community framework for electronic signatures.

Objective Name	Description
O.DIRECTIVE_ANNEX_III_PART_1a	The TOE guarantees the operation of its services in an execution environment with a FIPS 140-2 level 3 security level for the following processes: assuring the secure custody and the uniqueness property of the private key (the signature-creation data used for signature generation can practically occur only once).
O.DIRECTIVE_ANNEX_III_PART_1b	The TOE guarantees the operation of its services in an execution environment with a FIPS 140-2 level 3 security level for the following processes: assuring that the private key cannot be derived using

	currently-available technology, and that the signatures cannot been falsified.
O.DIRECTIVE_ANNEX_III_PART_1c	The TOE must ensure that the signature-creation data used for signature generation can be reliably protected by the legitimate signer against being used by others.
O.DIRECTIVE_ANNEX_III_PART_2	The TOE must provide integrity to the data to be signed.

Table 4-13. Security Objectives derived from Annex III "Requirements for secure signature-creation devices" of [EUROPEAN_DIRECTIVE]

The following security objectives in Table 4-14. Security Objectives derived from Annex IV "Requirements for secure signature-creation devices" of [EUROPEAN_DIRECTIVE] have been derived from Annex IV "Requirements for secure signature-creation devices" of the "Directive 1999/93/EC of the European Parliament and of the Council" of 13 December 1999 on a Community framework for electronic signatures.

Objective Name	**Description**
O.DIRECTIVE_ANNEX_IV_PART_a	During the signature-verification process, the TOE must ensure with reasonable certainty that the data used for verifying the signature corresponds to the data displayed to the verifier.
O.DIRECTIVE_ANNEX_IV_PART_b	During the signature-verification process, the TOE must ensure with reasonable certainty that the signature is reliably verified and the result of that validation is correctly displayed.
O.DIRECTIVE_ANNEX_IV_PART_c	During the signature-verification process, the TOE must ensure with reasonable certainty that the verifier can, as necessary, reliably establish the contents of the signed data.
O.DIRECTIVE_ANNEX_IV_PART_d	During the signature-verification process, the TOE must ensure with reasonable certainty that the authenticity and validity of the certificate required at the time of signature verification are reliably verified.
O.DIRECTIVE_ANNEX_IV_PART_e	During the signature-verification process, the TOE must ensure with reasonable certainty that the signer's identity is correctly displayed.

O.DIRECTIVE_ANNEX_IV_PART_f	During the signature-verification process, the TOE must ensure with reasonable certainty that the use of a pseudonym is supported.
O.DIRECTIVE_ANNEX_IV_PART_g	During the signature-verification process, the TOE must ensure with reasonable certainty that any security-relevant changes can be detected.

Table 4-14. Security Objectives derived from Annex IV "Requirements for secure signature-creation devices" of [EUROPEAN_DIRECTIVE]

CHAPTER 5

5. Security Requirements

This section defines the TOE security functional requirements and assurance requirements. Requirements are drown from the [PKE_PP] Protection Profile and CC Parts 3 and have been written as required as Part 2 extended requirements.

5.1. Extended Components Definition

The extended components used are those that are defined in the [PKE_PP] protection profile (claimed in this Security Target). Furthermore these components are used methodologically as they are defined in this PP.

5.2. Security Functional Requirements

The TOE is part 2 extended. All functional requirements included in this Security Target are listed in Table 5-1. Part 2 or Part 2 Extended Requirements, below. Extended requirements are identified as “Part 2 extended.” And their name ends with "EXT" or a NIAP interpretation tag.

Requirement	Part 2 or extended
FDP_ACC.1	Part 2
FIA_UAU.1	Part 2
FIA_UAU.4	Part 2
FIA_UAU.6	Part 2
FIA_UID.1	Part 2
FMT_MSA.1	Part 2
FMT_SMF.1	Part 2
FMT_SMR.1	Part 2
FDP_CPD_(EXT).1	Part 2 Extended

FDP_DAU_CPV_(EXT).1	Part 2 Extended
FDP_DAU_CPV_(EXT).2	Part 2 Extended
FDP_DAU_CPI_(EXT).1	Part 2 Extended
FDP_DAU_CPI_(EXT).2	Part 2 Extended
FDP_DAU_CPO_(EXT).1	Part 2 Extended
FDP_DAU_CPO_(EXT).2	Part 2 Extended
FDP_DAU_CRL_(EXT).1	Part 2 Extended
FDP_DAU_ENC_(EXT).1	Part 2 Extended
FDP_DAU_OCS_(EXT).1	Part 2 Extended
FDP_DAU_SIG_(EXT).1	Part 2 Extended
FDP_ETC_ENC_(EXT).1	Part 2 Extended
FDP_ETC_SIG_(EXT).1	Part 2 Extended
FDP_ITC_ENC_(EXT).1	Part 2 Extended
FDP_ITC_SIG_(EXT).1	Part 2 Extended
FIA_UAU_SIG_(EXT).1	Part 2 Extended
FAU_GEN.1-NIAP-0407:2	Part 2 Extended
FAU_GEN.2-NIAP-0410:2	Part 2 Extended
FDP_ACF.1	Part 2
FMT_MSA.3	Part 2
FTP_TRP.1	Part 2
FPT_TEE.1	Part 2

Table 5-1. Part 2 or Part 2 Extended Requirements

5.2.1. Security Functional Requirements for the TOE

The security functional requirements for the TOE are listed in Table 5-2. Security Functional Requirements for the TOE and Table 5-3. Security Functional Requirements for the TOE derived from the [EUROPEAN_DIRECTIVE].

Package Name	Functional Requirement	Dependency Package
Certification Path	FDP_CPD_(EXT).1	None

Validation - Basic	FDP_DAU_CPI_(EXT).1	
	FDP_DAU_CPV_(EXT).1	
	FDP_DAU_CPV_(EXT).2	
	FDP_DAU_CPO_(EXT).1	
Certification Path Validation – Basic Policy	FDP_DAU_CPI_(EXT).2	Certification Path Validation - Basic
	FDP_DAU_CPO_(EXT).2	
PKI Signature Generation	FDP_ETC_SIG_(EXT).1	None
PKI Signature Verification	FDP_ITC_SIG_(EXT).1	Certification Path Validation - Basic
	FDP_DAU_SIG_(EXT).1	
PKI Encryption using Key Transfer Algorithms	FDP_ETC_ENC_(EXT).1	Certification Path Validation - Basic
	FDP_DAU_ENC_(EXT).1	
PKI Decryption using Key Transfer Algorithms	FDP_ITC_ENC_(EXT).1	None
PKI Based Entity Authentication	FIA_UAU.1 FIA_UAU.4 FIA_UAU_SIG_(EXT).1 FIA_UID.1	Certification Path Validation – Basic
Online Certificate Status Protocol Client	FDP_DAU_OCS_(EXT).1	None
Certificate Revocation List Validation	FDP_DAU_CRL_(EXT).1	None
Audit	FAU_GEN.1-NIAP-0407:2	None
	FAU_GEN.2-NIAP-0410:2	
Continuous Authentication	FIA_UAU.6	PKI Based Entity Authentication Certification Path Validation - Basic

Table 5-2. Security Functional Requirements for the TOE from the [PKE_PP] PP

Extended functionality	Functional Requirement
Annex III (“Requirements for secure	FDP_ACC.1

signature-creation devices") of [EUROPEAN_DIRECTIVE]	FDP_ACF.1
	FPT_TEE.1
	FTP_TRP.1
	FIA_UID.1 (already included in the PKI Based Entity Authentication Package)
	FIA_UAU.1 (already included in the PKI Based Entity Authentication Package)
	FMT_MSA.1 (TOE) (dependency from FDP_ACC.1)
	FMT_MSA.3 (dependency from FDP_ACC.1)
	FMT_SMR.1 (TOE) (dependency from FDP_ACC.1)
	FMT_SMF.1 (TOE) (dependency from FDP_ACC.1)
Annex IV ("Recommendations for secure signature verification") of [EUROPEAN_DIRECTIVE]	FPT_TEE.1
	FTP_TRP.1
	FDP_DAU_SIG_(EXT).1 (already included in the PKI Signature Verification Package)
	FDP_ITC_SIG_(EXT).1 (already included in the PKI Signature VerificationPackage)
	FDP_DAU_CPV_(EXT).1 (already included in the Certification Path Verification- Basic Package)
	FDP_DAU_CPI_(EXT).1 (already included in the Certification Path Verification- Basic Package)
	FDP_DAU_CPO_(EXT).1 (already included in the Certification Path Verification- Basic Package)
	FDP_CPD_(EXT).1 (already included in the Certification Path Verification- Basic Package)
	FDP_DAU_CPV_(EXT).2 (already included in the Certification Path Verification- Basic Package)
	FAU_GEN.1-NIAP_0407:2

Table 5-3. Security Functional Requirements for the TOE derived from the [EUROPEAN_DIRECTIVE]

5.2.1.1. Certification Path Validation – Basic Package

The functions in this package address the validation of the certification path. Certification path development is also a part of this package.

All processing defined is X.509 and PKIX compliant. The certification path validation in these standards is procedural, but in keeping with the spirit of functional specification, certification path validation requirements are specified using non-procedural techniques.

From certification path processing perspective, certificates can be of up to three types:

- Self-signed trust anchor certificate: The trust anchor can be in the form of a self-signed certificate. The trust anchor is used to obtain the Distinguished Name (DN), public key, algorithm identifier, and the public key parameters (if applicable). This package permits validation of trust anchor if it is in the form of self-signed certificate, including validating signature and verifying that the self-signed certificate validity period has not expired.
- Intermediate certificates: These are the certificates issued to the CAs. All certificates in a certification path are intermediate certificates, except the last one.
- End certificate: This is the last certificate in the certification path and is issued to the subscriber of interest. This is typically an end-entity (i.e., not a CA) certificate. However, this package permits that certificate to be a CA certificate also.

This package processes the following security related certificate extensions checks: no-check, keyUsage, extendedKeyUsage, and basicConstraints.

The TOE provides the capability to validate path as of a user-defined time called TOI which can be current time or earlier.

Class FDP – User Data Protection

FDP_CPD_(EXT).1 Certification path development

Hierarchical to: No other components.

FDP_CPD_(EXT).1.1 The TSF shall develop a certification path from a trust anchor provided by [selection: *Web Services Consumer*] to the subscriber using matching rules for the following subscriber certificate fields or extensions: [selection: *distinguished name, subject key identifier*, [assignment: *authority key identifier*]].

FDP_CPD_(EXT).1.2 The TSF shall develop the certification path using the following additional matching rule: [selection: *none*].

FDP_CPD_(EXT).1.3 The TSF shall develop the certification path using the following additional matching rule [selection of one by the ST author: *none*].

FDP_CPD_(EXT).1.4 The TSF shall bypass any matching rules except [selection: *distinguished name*] if additional certification paths are required.

Dependencies: None

Application Note: *In FDP_CPD_(EXT).1.2, the assignment nonRepudiation should be used if the path is being developed for signature verification; the assignment digitalSignature should be used if the path is being developed for entity authentication; the assignment keyEncipherment, should be used if the path is being developed for encryption certificate using a key transfer algorithm (e.g., RSA); the assignment keyAgreement should be used if the path is being developed for encryption certificate using a key calculation algorithm (e.g., DH, ECDH).EAL4+ Configuration of the Product does not require the presence of any specific bit of the extension Key Usage. Anyway, the Product can be configured in order to require the presence of specific bits of this extension, for certain uses of a certificate.*

In FDP_CPD_(EXT).1.3, the selection of the matching rule should be made depending on the PKE application requirement. anyExtendedKeyUsage is a match for any application.

In FDP_CPD_(EXT).1.1, specifically, the profile of user referred in the use of the term Web Service Consumer, it is a user with Web Service Consumer role and Security Office group.

FDP_DAU_CPI_(EXT).1 Certification path initialization - basic

Hierarchical to: No other components.

FDP_DAU_CPI_(EXT).1.1 The TSF shall use the trust anchor provided by [selection: *Web Services Consumer*].

FDP_DAU_CPI_(EXT).1.2 The TSF shall obtain the time of interest called "TOI' from a reliable source [selection: *local environment, NTP server*].

FDP_DAU_CPI_(EXT).1.3 The TSF shall perform the following checks on the trust anchor [selection: *Subject DN and Issuer DN match; Signature verifies using the subject public key and parameter (if applicable) from the trust anchor; notBefore field in the trust anchor <= TOI; notAfter field in the trust anchor => TOI*]

FDP_DAU_CPI_(EXT).1.4 The TSF shall derive from the trust anchor [selection: *subject DN, subject public key, subject public key algorithm object identifier, subject public key parameters*]

Dependencies: FCS_COP.1, FPT_STM.1

Application Note: In FDP_DAU_CPI_(EXT).1.1, specifically, the profile of user referred in the use of the term Web Service Consumer, it is a user with Web Service Consumer role and Security Office group.

FDP_DAU_CPV_(EXT).1 Certificate processing - basic

Hierarchical to: No other components.

FDP_DAU_CPV_(EXT).1.1 The TSF shall reject a certificate if any of the following checks fails:

a) Use parent-public-key, parent-public-key-algorithm-identifier, and parent-public-key-parameters to verify the signature on the certificate;

b) notBefore field in the certificate < = TOI;

c) notAfter field in the certificate > = TOI;

d) issuer field in the certificate = parent-DN; or

e) TSF is able to process all extensions marked critical

FDP_DAU_CPV_(EXT).1.2 The TSF shall bypass the revocation status check if the certificate contains no-check extension.

FDP_DAU_CPV_(EXT).1.3 The TSF shall bypass the revocation check if the revocation information is not available and [selection: [assignment: *Web Services Consumer*]] overrides revocation checking.

FDP_DAU_CPV_(EXT).1.4 The TSF shall reject a certificate if the revocation status using [selection: *CRL, OCSP*] demonstrates that the certificate is revoked.

FDP_DAU_CPV_(EXT).1.5 The TSF shall update the public key parameters state machine using the following rules:

a) Obtain the parameters from the subjectPublickeyInfo field of certificate if the parameters are present in the field; else

b) Retain the old parameters state if the subject public key algorithm of current certificate and parent public key algorithm of current certificate belong to the same family of algorithms, else

c) Set parameters = “null”.

Dependencies: FCS_COP.1, FPT_STM.1, [FDP_DAU_OCS_(EXT).1 or FDP_DAU_CRL_(EXT).1]

Application Note: While each certificate is expected to be checked using only one of the revocation mechanisms, each certificate in a certification path can be checked using different revocation mechanism. That is why the selection is one or more.

In FDP_DAU_CPV_(EXT).1.3, specifically, the profile of user referred in the use of the term Web Service Consumer, it is a user with Web Service Consumer role and Security Office group.

FDP_DAU_CPV_(EXT).2 Intermediate certificate processing -- basic

Hierarchical to: No other components.

FDP_DAU_CPV_(EXT).2.1 The TSF shall reject an intermediate certificate if any of the following additional checks fails:

a) basicConstraints field is present with cA = TRUE;

b) pathLenConstraint is not violated; or

c) if a critical keyUsage extension is present, keyCertSign bit is set

Dependencies: FDP_DAU_CPV_(EXT).1

FDP_DAU_CPO_(EXT).1 Certification path output -- basic

Hierarchical to: No other components.

FDP_DAU_CPO_(EXT).1.1 The TSF shall output certification path validation failure if any certificate in the certification path is rejected.

FDP_DAU_CPO_(EXT).1.2 The TSF shall output the following variables from the end certificate: subject DN, subject public key algorithm identifier, subject public key, critical keyUsage extension.

FDP_DAU_CPO_(EXT).1.3 The TSF shall output the following additional variables from the end certificate [selection: *certificate, subject alternative names, extendedKeyUsage*].

FDP_DAU_CPO_(EXT).1.4 The TSF shall output the subject public key parameters from the certification path parameter state machine.

Dependencies: FDP_DAU_CPV_(EXT).1

5.2.1.2. Certification Path Validation – Basic Policy Package

The security functional requirements in this package address certificate path processing with the processing of certificate Policies extension. This package is dependent upon the Certification Path Validation – Basic package.

Class FDP – User Data Protection

FDP_DAU_CPI_(EXT).2 Certification path initialisation – basic policy

Hierarchical to: No other components.

FDP_DAU_CPI_(EXT).2.1 The TSF shall use the initial-certificate-policies provided by [selection:[assignment: *Web Services Consumer*]].

Dependencies: FDP_DAU_CPI_(EXT).1

FDP_DAU_CPO_(EXT).2 Certification path output – basic policy

Hierarchical to: No other components.

FDP_DAU_CPO_(EXT).2.1 The TSF shall output the certificate policies using the following rule: intersection of certificatePolicies extensions in all the certificates in certification path and initial-certificate-policies.

Dependencies: FDP_DAU_CPO_(EXT).1

Application Note: In FDP_DAU_CPI_(EXT).1.2, specifically, the profile of user referred in the use of the term Web Service Consumer, it is a user with Web Service Consumer role and Security Office group.

5.2.1.3. PKI Signature Generation Package

The PKI Signature Generation package invokes a cryptographic module for digital signature generation. The package functionality includes generation of signature information that identifies the signer and is useful in efficient signature verification.

Class FDP – User Data Protection

FDP_ETC_SIG_(EXT).1 Export of PKI Signature

Hierarchical to: No other component

FDP_ETC_SIG_(EXT).1.1 The TSF shall invoke the cryptographic module with the user selected private key to generate digital signature.

FDP_ETC_SIG_(EXT).1.2 The TSF shall include the following information with the digital signature [selection: *hashing algorithm, signature algorithm, signer public key certificate*].

Dependencies: FCS_CRM_FPS_(EXT).1

5.2.1.4. PKI Signature Verification Package

The PKI Signature Verification package processes and verifies the signature information, and invokes a cryptographic module to verify digital signatures. This package is dependent upon the Certification Path Validation – Basic package. The signature verification package uses the Certification Path Validation package data as input.

Class FDP – User Data Protection

FDP_ITC_SIG_(EXT).1 Import of PKI Signature

Hierarchical to no other component

FDP_ITC_SIG_(EXT).1.1 The TSF shall use the following information from the signed data [selection: *hashing algorithm, signature algorithm, signer public key certificate, signer DN, signer subject key identifier*] during signature verification.

Dependencies: None

FDP_DAU_SIG_(EXT).1 Signature Blob Verification

Hierarchical to: No other components.

FDP_DAU_SIG_(EXT).1.1 The TSF shall invoke the cryptographic module with the following information from Certification Path Validation to verify digital signature on signed data: subject public key algorithm, subject public key, subject public key parameters.

FDP_DAU_SIG_(EXT).1.2 The TSF shall verify that the keyUsage extension output from the Certification Path Validation has the [selection: *nonRepudiation or digitalSignature*] bit set.

FDP_DAU_SIG_(EXT).1.3 The TSF shall apply the following additional checks [selection: *match the subject DN from the Certification Path Validation with that in the signed data*]

Dependencies: FCS_CRM_FPS_(EXT).1, FDP_DAU_CPO_(EXT).1

5.2.1.5. PKI Encryption using Key Transfer Algorithms Package

This package supports the performance of public key encryption using key transfer algorithms such as RSA. Certification path validation is used to ensure that the correct public key of the decrypting party is used. This package is dependent upon the Certification Path Validation – Basic package.

Class FDP – User Data Protection

FDP_ETC_ENC_(EXT).1 Export of PKI Encryption – Key Transfer Algorithms

Hierarchical to: No other component

FDP_ETC_ENC_(EXT).1.1 The TSF shall include the following information with the encrypted data [selection: *key encryption algorithm, data encryption algorithm, decryptor key identifier*].

FDP_ETC_ENC_(EXT).1.2 The TSF shall invoke the cryptographic module with the following information from Certification Path Validation to create encrypted data: subject public key algorithm, subject public key, subject public key parameters.

Dependencies: FCS_CRM_FPS_(EXT).1, FDP_DAU_CPO_(EXT).1

FDP_DAU_ENC_(EXT).1 PKI Encryption Verification – Key Transfer

Hierarchical to: No other components.

FDP_DAU_ENC_(EXT).1.1 The TSF shall verify that the keyUsage output from Certification Path Validation contains keyEncipherment bit set.

FDP_DAU_ENC_(EXT).1.2 The TSF shall apply the following additional checks [selection: *match the subject DN from the Certification Path Validation with that of the subject of interest*].

Dependencies: FDP_DAU_CPO_(EXT).1

Application Note: This component is used to verify that the correct public key is used during encryption.

5.2.1.6. PKI Decryption using Key Transfer Algorithms Package

This package supports the performance of public key decryption using key transfer algorithms such as RSA. Since only the decrypting party's private key is used, this package does not depend upon certificate path processing.

Class FDP – User Data Protection

FDP_ITC_ENC_(EXT).1 Import of PKI Encryption – Key Transfer Algorithms

Hierarchical to: No other components

FDP_ITC_ENC_(EXT).1.1 The TSF shall invoke the cryptographic module with the following information from the encrypted data [selection: *key encryption algorithm, data encryption algorithm, decryptor key identifier*] to perform decryption.

Dependencies: FCS_CRM_FPS_(EXT).1

5.2.1.7. PKI Based Entity Authentication Package

This package provides for the use of PKI as an entity authentication service. The identification and authentication (I&A) requirements in this package have a different purpose than I&A requirements for the IT Environment. The IT Environment requirements are always required and are used to manage and use the cryptographic keys, whereas this PKI Based Entity Authentication package is used when the PKE application (TOE) performs entity authentication (e.g., Secure Socket Layer (SSL), Transport Layer Security (TLS), etc.).

This package is dependent upon the Certification Path Validation – Basic package.

Class FIA – Identification and Authentication

FIA_UAU.1 Timing of authentication

Hierarchical to: No other components

FIA_UAU.1.1 The TSF shall allow [assignment: *carry out an identification successfully, introduce authentication data*] 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.

Dependencies: FIA_UID.1 Timing of identification

FIA_UAU.4 Single-use authentication mechanisms

Hierarchical to: No other components

FIA_UAU.4.1 The TSF shall prevent reuse of authentication data related to [selection: *TLS*].

Dependencies: No dependencies

FIA_UAU_SIG_(EXT).1 Entity Authentication

Hierarchical to: No other components.

FIA_UAU_SIG_(EXT).1.1 The TSF shall invoke the cryptographic module with the following information from Certification Path Validation to verify signature on response

from the entity to the challenge from the TSF: subject public key algorithm, subject public key, subject public key parameters.

FIA_UAU_SIG_(EXT).1.2 The TSF shall verify that the keyUsage output from Certification Path Validation contains digitalSignature bit set.

FIA_UAU_SIG_(EXT).1.3 The TSF shall apply the following additional checks [selection: *match the subject DN from the Certification Path Validation with the entity being authenticated*].

Dependencies: FCS_CRM_FPS_(EXT).1, FDP_DAU_CPO_(EXT).1

FIA_UID.1 Timing of identification

Hierarchical to: No other components

FIA_UID.1.1 The TSF shall allow [assignment: *enable the establishment a secure path), introduce the user identification data*] 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.

Dependencies: No dependencies

5.2.1.8. Online Certificate Status Protocol Client Package

This package allows for making Online Certificate Status Protocol (OCSP) requests and validating OCSP responses. This package permits the use of the OCSP Responder as a trust anchor, as the CA, or an end entity authorized to sign OCSP responses.

Class FDP – User Data Protection

FDP_DAU_OCS_(EXT).1 Basic OCSP Client

Hierarchical to: No other component

FDP_DAU_OCS_(EXT).1.1 The TSF shall formulate the OCSP requests in accordance with PKIX RFC 2560.

FDP_DAU_OCS_(EXT).1.2 The OCSP request shall contain the following extensions: [selection:[assignment: *any or some extension identified in PKIX RFC 2560*]].

FDP_DAU_OCS_(EXT).1.3 The TSF shall obtain the public key, algorithm, and public key parameters of the OCSP Responder from [selection: *OCSP responder certificate*].

FDP_DAU_OCS_(EXT).1.4 The TSF shall perform the following additional function [selection: *establish trust in OCSP responder certificate using [selection: certification path validation – basic*].

FDP_DAU_OCS_(EXT).1.5 The TSF shall invoke the cryptographic module to verify signature on the OCSP response using trusted public key, algorithm, and public key parameters of the OCSP responder.

FDP_DAU_OCS_(EXT).1.6 The TSF shall verify that if the OCSP responder certificate contains extendedKeyUsage extension, the extension contains the PKIX OID for ocsp-signing or the anyExtendedKeyUsage OID.

FDP_DAU_OCS_(EXT).1.7 The TSF shall match the responderID in the OCSP response with the corresponding information in the responder certificate

FDP_DAU_OCS_(EXT).1.8 The TSF shall match the certID in a request with certID in singleResponse.

FDP_DAU_OCS_(EXT).1.9 The TSF shall reject the OCSP response for an entry if all of the following are true:

a) time checks are not overridden;

b) [selection: *TOI > producedAt + x where x is provided by [selection: [assignment: Web Services Consumer]]*];

c) [selection: *TOI > thisUpdate for entry + x where x is provided by [selection: [assignment: Web Services Consumer]]*]; and

d) [selection: *TOI > nextUpdate for entry + x if nextUpdate is present and where x is provided by [selection: [assignment: Web Services Consumer]]*].

FDP_DAU_OCS_(EXT).1.10 The TSF shall permit [selection: *Web Services Consumer*] to override time checks.

FDP_DAU_OCS_(EXT).1.11 The TSF shall reject OCSP response if the response contains "critical" extension(s) that TSF does not process.

FDP_DAU_OCS_(EXT).1.12 The TSF shall perform the following additional checks [selection: *request nonce = response nonce*].

Dependencies: FCS_CRM_FPS_(EXT).1, FPT_STM.1

Application Note: In FDP_DAU_OCS_(EXT).1.9 and FDP_DAU_OCS_(EXT).1.10, specifically, the profile of user referred in the use of the term Web Service Consumer, it is a user with Web Service Consumer role and Security Office group.

5.2.1.9. Certificate Revocation List (CRL) Validation Package

This package is used for validating a CRL.

Class FDP – User Data Protection

FDP_DAU_CRL_(EXT).1 Basic CRL Checking

Hierarchical to no other component

FDP_DAU_CRL_(EXT).1.1 The TSF shall obtain the CRL from [selection: *local cache, repository, location pointed to by the CRL DP in public key certificate of interest*].

FDP_DAU_CRL_(EXT).1.2 The TSF shall obtain the trusted public key, algorithm, and public key parameters of the CRL issuer.

FDP_DAU_CRL_(EXT).1.3 The TSF shall invoke the cryptographic module to verify signature on the CRL using trusted public key, algorithm, and public key parameters of the CRL issuer.

FDP_DAU_CRL_(EXT).1.4 The TSF shall verify that if a critical keyUsage extension is present in CRL issuer certificate, cRLSign bit in the extension is set in the certificate.

FDP_DAU_CRL_(EXT).1.5 The TSF shall match the issuer field in the CRL with what it assumes to be the CRL issuer.

FDP_DAU_CRL_(EXT).1.6 The TSF shall reject the CRL if all of the following are true:

a) Time check are not overridden;

b) [selection: *TOI > thisUpdate + x where x is provided by [selection:[assignment: Web Services Consumer]*]]; and

c) [selection: *TOI > nextUpdate + x if nextUpdate is present and where x is provided by [selection: [assignment: Web Services Consumer]]*].

FDP_DAU_CRL_(EXT).1.7 The TSF shall permit [selection: *Web Services Consumer*] to override time checks.

FDP_DAU_CRL_(EXT).1.8 The TSF shall reject CRL if the CRL contains "critical" extension(s) that TSF does not process.

FDP_DAU_CRL_(EXT).1.9 The TSF shall perform the following additional checks [selection: *none*].

Dependencies: FCS_CRM_FPS_(EXT).1, FPT_STM.1

Application Note: The trusted public key, algorithm, and public key parameters of the CRL issuer should normally be the same as those used for verifying signature on the certificate being checked for revocation. If not, at least certificate path development – basic can be used to obtain the public key.

In FDP_DAU_CRL_(EXT).1.6 and FDP_DAU_CRL_(EXT).1.7, specifically, the profile of user referred in the use of the term Web Service Consumer, it is a user with Web Service Consumer role and Security Office group.

5.2.1.10. Audit Package

This package is used in order to generate and protect audit events relevant to the TOE. The dependencies for this package are satisfied by the IT Environment functional requirements.

Class FAU – Security Audit

FAU_GEN.1-NIAP-0407:2 Audit data generation – TOE

Hierarchical to: No other component

FAU_GEN.1.1-NIAP-0407;2 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) All auditable events listed in Table 5-4. TOE Auditable Events; and

c) [selection: *"any security-relevant changes can be detected"*].

FAU_GEN.1.2-NIAP-0410;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 (if applicable), 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 ST, information specified in column three of Table 5-4. TOE Auditable Events below.

Dependencies: FPT_STM.1 Reliable time stamps

Requirement	Auditable Events	Additional Audit Record Contents
FDP_CPD_(EXT).1	Success or failure to build path	For success, matching rules bypassed
FDP_DAU_CPI_(EXT).1	None	
FDP_DAU_CPV_(EXT).1	Success or failure of certificate processing Bypass of revocation status checking	For failure, reason(s) for failure
FDP_DAU_CPV_(EXT).2	Success or failure of certificate processing	For failure, reason(s) for failure
FDP_DAU_CPO_(EXT).1	None	
FDP_DAU_CPI_(EXT).2	None	
FDP_DAU_CPO_(EXT).2	None	
FDP_DAU_CPI_(EXT).3	None	
FDP_ETC_SIG_(EXT).1	Invocation of the function	
FDP_ITC_SIG_(EXT).1	None	
FDP_DAU_SIG_(EXT).1	Success or failure	In case of failure, reason for failure
FDP_ETC_ENC_(EXT).1	None	
FDP_DAU_ENC_(EXT).1	Success or failure	In case of failure, reason for failure
FDP_ITC_ENC_(EXT).1	Invocation of the function	

FIA_UAU.1	All use of authentication mechanism	
FIA_UAU.4	Attempt to reuse authentication data	
FIA_UAU_SIG_(EXT).1	Success or failure	In case of failure, reason for failure
FIA_UID.1	All use of identification mechanism	User identity
FDP_DAU_OCS_(EXT).1	Rejection of OCSP response Override time checks	Reason for rejection
FDP_DAU_CRL_(EXT).1	Rejection of CRL Override time checks	Reason for rejection
FAU_GEN.1-NIAP-0407:2	None	
FAU_GEN.2-NIAP-0410:2	None	
FIA_UAU.6	All re-authentication attempts	

Table 5-4. TOE Auditable Events

In the table above, if a component is included in this Security Target, then and only then the audit record event for that component will be generated.

FAU_GEN.2-NIAP-0410:2 User identity association – TOE

Hierarchical to: No other components.

FAU_GEN.2.1-NIAP-0410;2 For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

Dependencies:

FAU_GEN.1 Audit data generation

FIA_UID.1 Timing of identification

5.2.1.11. Continuous Authentication Package

This package provides for the use of the continuous authentication service of an entity. This package is dependent on the PKI Based Entity Authentication Package and the CPV – Basic package. This package is used for continuous authentication of an entity.

Class FIA – Identification and Authentication

FIA_UAU.6 Re-authenticating

Hierarchical to: No other components

FIA_UAU.6.1 The TSF shall re-authenticate the user under the conditions [selection: [assignment: *each transaction*]].

Dependencies: No dependencies

Application Note: It is acceptable to use the symmetric session cryptographic key established during the initial authentication in conjunction with integrity and authentication functions such as HMAC for re-authentication of commands, packets, transactions, etc.

5.2.1.12. Annex III and Annex IV of the [EUROPEAN_DIRECTIVE]

This section provides functional requirements derived from the Annex III ("Requirements for secure signature-creation devices") and Annex IV ("Recommendations for secure signature verification") of the [EUROPEAN_DIRECTIVE] Directive. Other functional requirements not included in this section are needed to satisfy the demands of this Annex III, but they are already included in some Package of the [PKE_PP] PP.

Class FDP – User Data Protection

FDP_ACC.1 Subset access control

Hierarchical to: No other components

FDP_ACC.1.1 The TSF shall enforce the [assignment: *TrustedX Access Control Policy*] on [assignment: *subjects: all users of the application; objects: any resource of the system; operations: any operation where the resource is involved*]

Dependencies:

FDP_ACF.1 Security attribute based access control

FDP_ACF.1 Security attribute based access control

Hierarchical to: No other components.

FDP_ACF.1.1 The TSF shall enforce the [assignment: *TrustedX Access Control Policy*] to objects based on the following: [assignment: *subjects and objects controlled: all users of the application (subjects), and any resource of the system (objects); security attributes: a) subject authentication information (credentials and authentication tokens), b) level of authentication (low, medium, high and very high), c) list of groups and groups of groups to whom the subject who tries the access belongs, and d) environment information (hour and day in which the access is tried, and IP address of the machine from which the access is tried)*.

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [assignment: *possession of necessary permissions on the part of the users that request access to a resource*].

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

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the [assignment: *none*].

Dependencies:

FDP_ACC.1 Subset access control

FMT_MSA.3 Static attribute initialization

FTP_TRP.1 Trusted path

Hierarchical to: No other components.

FTP_TRP.1.1 The TSF shall provide a communication path between itself and [selection: remote] users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from [selection: modification, disclosure].

FTP_TRP.1.2 The TSF shall permit [selection: remote users] to initiate communication via the trusted path.

FTP_TRP.1.3 The TSF shall require the use of the trusted path for [selection: any SOAP service initiated by the user].

Dependencies:

No dependencies

Class FMT – Security Management

FMT_MSA.1 Management of security attributes

Hierarchical to: No other components.

FMT_MSA.1.1 The TSF shall enforce the [assignment: *TrustedX Access Control Policy*] to restrict the ability to [selection: *modify*] the security attributes [assignment:*a) subject authentication information (credentials and authentication tokens), b) level of authentication (low, medium, high and very high), c) list of groups and groups of groups to whom the subject who tries the access belongs, and d) environment information (hour and day in which the access is tried, and IP address of the machine from which the access is tried)*] to [assignment: the *Web Services Consumer*].

Dependencies:

FDP_ACC.1 Subset access control

FMT_SMR.1 Security roles

FMT_SMF.1 Specification of Management Functions

Application Note: In FMT_MSA.1.1, specifically, the profile of user referred in the use of the term Web Service Consumer, it is a user with Web Service Consumer role and Security Office group.

FMT_MSA.3 Static attribute initialization

Hierarchical to: No other components.

FMT_MSA.3.1 The TSF shall enforce the [assignment: *TrustedX Access Control Policy*] to provide [selection, choose one of: *restrictive*] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the [assignment: *the Web Services Consumer*] to specify alternative initial values to override the default values when an object or information is created.

Dependencies:

FMT_MSA.1 Management of Security Attributes

FMT_SMR.1 Security roles

Application Note: In FMT_MSA.3.2, specifically, the profile of user referred in the use of the term Web Service Consumer, it is a user with Web Service Consumer role and Security Office group.

FMT_SMR.1 Security management roles

Hierarchical to: No other components.

FMT_SMR.1.1 The TSF shall maintain the roles [assignment: *Web Services Consumer, Administrator through the Command Shell*].

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

Dependencies:

FIA_UID.1 Timing of identification

FMT_SMF.1 Specification of Management Functions

Hierarchical to: No other components.

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions: [assignment: *administration of privileged users, administration of end entities, configuration parameters of an entity, administration of an entity's keystore, administration of an SSL keystore, group administration, administration of Templates for Dynamic Groups, administration of trusted entities, administration of CA groups, policy administration, service administration, system configuration, configuration of the HSM device parameters, administration managed by the command interpreter*].

Dependencies:

No dependencies.

Class FPT – Protection of the TSF

FPT_TEE.1 Testing of external entities

Hierarchical to: No other components.

FPT_TEE.1.1 The TSF shall run a suite of tests [selection: [assignment: each time the TrustedX services are started]] to check the fulfillment of [assignment: a level 3 technical configuration of the FIPS 140-2 cryptographic device].

FPT_TEE.1.2 If the test fails, the TSF shall [assignment: block all the SOAP accesses].

Dependencies:

No dependencies.

5.3. Security Assurance Requirements

The TOE Evaluation Assurance Level is EAL4 augmented by ALC_FLR.2.

EAL4 permits a Public Key Enabled application developer to gain added assurance from positive security engineering based on good commercial development practices, which, though rigorous, do not require substantial specialist knowledge, skills, and other resources. ALC_FLR.2 augmentation is done to ensure compliance with the Enhanced-Basic Robustness assurance requirements. The assurance components are listed in Table 5-5. EAL4 with Augmentation Assurance Requirements. These Security Assurance Requirements are drawn from the Common Criteria for Information Technology Security Evaluation, Part 3, Version 3.1, Revision 2, February 2009.

Assurance Class	Assurance Components	Assurance Components Description
Security Target Evaluation	ASE_CCL.1	Conformance Claims
	ASE_ECD.1	Extended Components Definition
	ASE_INT.1	Security Target Introduction
	ASE_OBJ.2	Security Objectives
	ASE_REQ.2	Security Requirements
	ASE_SPD.1	Security Problem Definition
	ASE_TSS.1	TOE summary specification
Development	ADV_ARC.1	Security architecture

		description
	ADV_FSP.4	Complete functional specification
	ADV_IMP.1	Implementation representation of the TSF
	ADV_TDS.3	Basic modular design
Guidance Documents	AGD_OPE.1	Operational user guidance
	AGD_PRE.1	Preparative procedures
Life Cycle Support	ALC_CMC.4	Production support, acceptance procedures and automation
	ALC_CMS.4	Problem tracking CM coverage
	ALC_DEL.1	Delivery procedures
	ALC_DVS.1	Identification of security measures
	ALC_FLR.2	Flaw reporting procedures
	ALC_LCD.1	Developer defined lift-cycle model
	ALC_TAT.1	Well-defined development tools
Tests	ATE_COV.2	Analysis of coverage
	ATE_DPT.2	Testing: security enforcing modules
	ATE_FUN.1	Functional testing
	ATE_IND.2	Independent testing - sample
Vulnerability Assessment	AVA_VAN.3	Focused vulnerability analysis

Table 5-5. EAL4 with Augmentation Assurance Requirements

5.3.1.1. Security Target Evaluation - ASE_ECD.1

Dependencies

No dependencies.

Developer action elements

ASE_ECD.1.1D The developer shall provide a statement of security requirements.

ASE_ECD.1.2D The developer shall provide an extended components definition.

Content and presentation elements

ASE_ECD.1.1C The statement of security requirements shall identify all extended security requirements.

ASE_ECD.1.2C The extended components definition shall define an extended component for each extended security requirement.

ASE_ECD.1.3C The extended components definition shall describe how each extended component is related to the existing CC components, families, and classes.

ASE_ECD.1.4C The extended components definition shall use the existing CC components, families, classes, and methodology as a model for presentation.

ASE_ECD.1.5C The extended components shall consist of measurable and objective elements such that conformance or nonconformance to these elements can be demonstrated.

5.3.1.2. Security Target Evaluation - ASE_ECD.1

Dependencies

No dependencies.

Developer action elements

ASE_ECD.1.1D The developer shall provide a statement of security requirements.

ASE_ECD.1.2D The developer shall provide an extended components definition.

Content and presentation elements

ASE_ECD.1.1C The statement of security requirements shall identify all extended security requirements.

ASE_ECD.1.2C The extended components definition shall define an extended component for each extended security requirement.

ASE_ECD.1.3C The extended components definition shall describe how each extended component is related to the existing CC components, families, and classes.

ASE_ECD.1.4C The extended components definition shall use the existing CC components, families, classes, and methodology as a model for presentation.

ASE_ECD.1.5C The extended components shall consist of measurable and objective elements such that conformance or nonconformance to these elements can be demonstrated.

5.3.1.3. Security Target Evaluation - ASE_INT.1

Dependencies

No dependencies.

Developer action elements

ASE_INT.1.1D The developer shall provide an ST introduction.

Content and presentation elements

ASE_INT.1.1C The ST introduction shall contain an ST reference, a TOE reference, a TOE overview and a TOE description.

ASE_INT.1.2C The ST reference shall uniquely identify the ST.

ASE_INT.1.3C The TOE reference shall identify the TOE.

ASE_INT.1.4C The TOE overview shall summarise the usage and major security features of the TOE.

ASE_INT.1.5C The TOE overview shall identify the TOE type.

ASE_INT.1.6C The TOE overview shall identify any non-TOE hardware/software/firmware required by the TOE.

ASE_INT.1.7C The TOE description shall describe the physical scope of the TOE.

ASE_INT.1.8C The TOE description shall describe the logical scope of the TOE.

5.3.1.4. Security Target Evaluation - ASE_OBJ.2

Dependencies

ASE_SPD.1 Security problem definition

Developer action elements

ASE_OBJ.2.1D The developer shall provide a statement of security objectives.

ASE_OBJ.2.2D The developer shall provide a security objectives rationale.

Content and presentation elements

ASE_OBJ.2.1C The statement of security objectives shall describe the security objectives for the TOE and the security objectives for the operational environment.

ASE_OBJ.2.2C The security objectives rationale shall trace each security objective for the TOE back to threats countered by that security objective and OSPs enforced by that security objective.

ASE_OBJ.2.3C The security objectives rationale shall trace each security objective for the operational environment back to threats countered by that security objective,

OSPs enforced by that security objective, and assumptions upheld by that security objective.

ASE_OBJ.2.4C The security objectives rationale shall demonstrate that the security objectives counter all threats.

ASE_OBJ.2.5C The security objectives rationale shall demonstrate that the security objectives enforce all OSPs.

ASE_OBJ.2.6C The security objectives rationale shall demonstrate that the security objectives for the operational environment uphold all assumptions.

5.3.1.5. Security Target Evaluation - ASE_REQ.2

Dependencies

ASE_OBJ.2 Security objectives

ASE_ECD.1 Extended components definition

Developer action elements

ASE_REQ.2.1D The developer shall provide a statement of security requirements.

ASE_REQ.2.2D The developer shall provide a security requirements rationale.

Content and presentation elements

ASE_REQ.2.1C The statement of security requirements shall describe the SFRs and the SARs.

ASE_REQ.2.2C All subjects, objects, operations, security attributes, external entities and other terms that are used in the SFRs and the SARs shall be defined.

ASE_REQ.2.3C The statement of security requirements shall identify all operations on the security requirements.

ASE_REQ.2.4C All operations shall be performed correctly.

ASE_REQ.2.5C Each dependency of the security requirements shall either be satisfied, or the security requirements rationale shall justify the dependency not being satisfied.

ASE_REQ.2.6C The security requirements rationale shall trace each SFR back to the security objectives for the TOE.

ASE_REQ.2.7C The security requirements rationale shall demonstrate that the SFRs meet all security objectives for the TOE.

ASE_REQ.2.8C The security requirements rationale shall explain why the SARs were chosen.

ASE_REQ.2.9C The statement of security requirements shall be internally consistent.

5.3.1.6. Security Target Evaluation - ASE_SPD.1

Dependencies

No dependencies.

Developer action elements

ASE_SPD.1.1D The developer shall provide a security problem definition.

Content and presentation elements

ASE_SPD.1.1C The security problem definition shall describe the threats.

ASE_SPD.1.2C All threats shall be described in terms of a threat agent, an asset, and an adverse action.

ASE_SPD.1.3C The security problem definition shall describe the OSPs.

ASE_SPD.1.4C The security problem definition shall describe the assumptions about the operational environment of the TOE.

5.3.1.7. Security Target Evaluation - ASE_TSS.1

Dependencies

ASE_INT.1 ST introduction

ASE_REQ.1 Stated security requirements

ADV_FSP.1 Basic functional specification

Developer action elements

ASE_TSS.1.1D The developer shall provide a TOE summary specification.

Content and presentation elements

ASE_TSS.1.1C The TOE summary specification shall describe how the TOE meets each SFR.

5.3.1.8. Development – ADV_ARC.1

Dependencies

ADV_FSP.1 Basic functional specification

ADV_TDS.1 Basic design

Developer action elements

ADV_ARC.1.1D The developer shall design and implement the TOE so that the security features of the TSF cannot be bypassed.

ADV_ARC.1.2D The developer shall design and implement the TSF so that it is able to protect itself from tampering by untrusted active entities.

ADV_ARC.1.3D The developer shall provide a security architecture description of the TSF.

Content and presentation elements

ADV_ARC.1.1C The security architecture description shall be at a level of detail commensurate with the description of the SFR-enforcing abstractions described in the TOE design document.

ADV_ARC.1.2C The security architecture description shall describe the security domains maintained by the TSF consistently with the SFRs.

ADV_ARC.1.3C The security architecture description shall describe how the TSF initialisation process is secure.

ADV_ARC.1.4C The security architecture description shall demonstrate that the TSF protects itself from tampering.

ADV_ARC.1.5C The security architecture description shall demonstrate that the TSF prevents bypass of the SFR-enforcing functionality.

5.3.1.9. Development – ADV_FSP.4

Dependencies

ADV_TDS.1 Basic design

Developer action elements

ADV_FSP.4.1D The developer shall provide a functional specification.

ADV_FSP.4.2D The developer shall provide a tracing from the functional specification to the SFRs.

Content and presentation elements

ADV_FSP.4.1C The functional specification shall completely represent the TSF.

ADV_FSP.4.2C The functional specification shall describe the purpose and method of use for all TSFI.

ADV_FSP.4.3C The functional specification shall identify and describe all parameters associated with each TSFI.

ADV_FSP.4.4C The functional specification shall describe all actions associated with each TSFI.

ADV_FSP.4.5C The functional specification shall describe all direct error messages that may result from an invocation of each TSFI.

ADV_FSP.4.6C The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

5.3.1.10. Development – ADV_IMP.1

Dependencies

ADV_TDS.3 Basic modular design

ALC_TAT.1 Well-defined development tools

Developer action elements

ADV_IMP.1.1D The developer shall make available the implementation representation for the entire TSF.

ADV_IMP.1.2D The developer shall provide a mapping between the TOE design description and the sample of the implementation representation.

Content and presentation elements

ADV_IMP.1.1C The implementation representation shall define the TSF to a level of detail such that the TSF can be generated without further design decisions.

ADV_IMP.1.2C The implementation representation shall be in the form used by the development personnel.

ADV_IMP.1.3C The mapping between the TOE design description and the sample of the implementation representation shall demonstrate their correspondence.

5.3.1.11. Development – ADV_TDS.3

Dependencies

ADV_FSP.4 Complete functional specification

Developer action elements

ADV_TDS.3.1D The developer shall provide the design of the TOE.

ADV_TDS.3.2D The developer shall provide a mapping from the TSFI of the functional specification to the lowest level of decomposition available in the TOE design.

Content and presentation elements

ADV_TDS.3.1C The design shall describe the structure of the TOE in terms of subsystems.

ADV_TDS.3.2C The design shall describe the TSF in terms of modules.

ADV_TDS.3.3C The design shall identify all subsystems of the TSF.

ADV_TDS.3.4C The design shall provide a description of each subsystem of the TSF.

ADV_TDS.3.5C The design shall provide a description of the interactions among all subsystems of the TSF.

ADV_TDS.3.6C The design shall provide a mapping from the subsystems of the TSF to the modules of the TSF.

ADV_TDS.3.7C The design shall describe each SFR-enforcing module in terms of its purpose and interaction with other modules.

ADV_TDS.3.8C The design shall describe each SFR-enforcing module in terms of its SFR-related interfaces, return values from those interfaces, interaction with and called interfaces to other modules.

ADV_TDS.3.9C The design shall describe each SFR-supporting or SFR-non-interfering module in terms of its purpose and interaction with other modules.

ADV_TDS.3.10C The mapping shall demonstrate that all behaviour described in the TOE design is mapped to the TSFIs that invoke it.

5.3.1.12. Guidance Documents – AGD_OPE.1

Dependencies

ADV_FSP.1 Basic functional specification

Developer action elements

AGD_OPE.1.1D The developer shall provide operational user guidance.

Content and presentation elements

AGD_OPE.1.1C The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.

AGD_OPE.1.2C The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.

AGD_OPE.1.3C The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.

AGD_OPE.1.4C The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_OPE.1.5C The operational user guidance shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AGD_OPE.1.6C The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfil the security objectives for the operational environment as described in the ST.

AGD_OPE.1.7C The operational user guidance shall be clear and reasonable.

5.3.1.13. Guidance Documents – AGD_PRE.1

Dependencies

No dependencies.

Developer action elements

AGD_PRE.1.1D The developer shall provide the TOE including its preparative procedures.

Content and presentation elements

AGD_PRE.1.1C The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.

AGD_PRE.1.2C The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.

5.3.1.14. Life Cycle Support – ALC_CMC.4

Dependencies

ALC_CMS.1 TOE CM coverage

ALC_DVS.1 Identification of security measures

ALC_LCD.1 Developer defined life-cycle model

Developer action elements

ALC_CMC.4.1D The developer shall provide the TOE and a reference for the TOE.

ALC_CMC.4.2D The developer shall provide the CM documentation.

ALC_CMC.4.3D The developer shall use a CM system.

Content and presentation elements

ALC_CMC.4.1C The TOE shall be labelled with its unique reference.

ALC_CMC.4.2C The CM documentation shall describe the method used to uniquely identify the configuration items.

ALC_CMC.4.3C The CM system shall uniquely identify all configuration items.

ALC_CMC.4.4C The CM system shall provide automated measures such that only authorised changes are made to the configuration items.

ALC_CMC.4.5C The CM system shall support the production of the TOE by automated means.

ALC_CMC.4.6C The CM documentation shall include a CM plan.

ALC_CMC.4.7C The CM plan shall describe how the CM system is used for the development of the TOE.

ALC_CMC.4.8C The CM plan shall describe the procedures used to accept modified or newly created configuration items as part of the TOE.

ALC_CMC.4.9C The evidence shall demonstrate that all configuration items are being maintained under the CM system.

ALC_CMC.4.10C The evidence shall demonstrate that the CM system is being operated in accordance with the CM plan.

5.3.1.15. Life Cycle Support – ALC_CMS.4

Dependencies

No dependencies.

Developer action elements

ALC_CMS.4.1D The developer shall provide a configuration list for the TOE.

Content and presentation elements

ALC_CMS.4.1C The configuration list shall include the following: the TOE itself; the evaluation evidence required by the SARs; the parts that comprise the TOE; the implementation representation; and security flaw reports and resolution status.

ALC_CMS.4.2C The configuration list shall uniquely identify the configuration items.

ALC_CMS.4.3C For each TSF relevant configuration item, the configuration list shall indicate the developer of the item.

5.3.1.16. Life Cycle Support – ALC_DEL.1

Dependencies

No dependencies.

Developer action elements

ALC_DEL.1.1D The developer shall document procedures for delivery of the TOE or parts of it to the consumer.

ALC_DEL.1.2D The developer shall use the delivery procedures.

Content and presentation elements

ALC_DEL.1.1C The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to the consumer.

5.3.1.17. Life Cycle Support – ALC_DVS.1

Dependencies

No dependencies.

Developer action elements

ALC_DVS.1.1D The developer shall produce development security documentation.

Content and presentation elements

ALC_DVS.1.1C The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.

5.3.1.18. Life Cycle Support – ALC_FLR.2

Dependencies

No dependencies.

Developer action elements

ALC_FLR.2.1D The developer shall document flaw remediation procedures addressed to TOE developers.

ALC_FLR.2.2D The developer shall establish a procedure for accepting and acting upon all reports of security flaws and requests for corrections to those flaws.

ALC_FLR.2.3D The developer shall provide flaw remediation guidance addressed to TOE users.

Content and presentation elements

ALC_FLR.2.1C The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.

ALC_FLR.2.2C The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.

ALC_FLR.2.3C The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

ALC_FLR.2.4C The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users.

ALC_FLR.2.5C The flaw remediation procedures shall describe a means by which the developer receives from TOE users reports and enquiries of suspected security flaws in the TOE.

ALC_FLR.2.6C The procedures for processing reported security flaws shall ensure that any reported flaws are remediated and the remediation procedures issued to TOE users.

ALC_FLR.2.7C The procedures for processing reported security flaws shall provide safeguards that any corrections to these security flaws do not introduce any new flaws.

ALC_FLR.2.8C The flaw remediation guidance shall describe a means by which TOE users report to the developer any suspected security flaws in the TOE.

5.3.1.19. Life Cycle Support – ALC_LCD.1

Dependencies

No dependencies.

Developer action elements

ALC_LCD.1.1D The developer shall establish a life-cycle model to be used in the development and maintenance of the TOE.

ALC_LCD.1.2D The developer shall provide life-cycle definition documentation.

Content and presentation elements

ALC_LCD.1.1C The life-cycle definition documentation shall describe the model used to develop and maintain the TOE.

ALC_LCD.1.2C The life-cycle model shall provide for the necessary control over the development and maintenance of the TOE.

5.3.1.20. Life Cycle Support – ALC_TAT.1

Dependencies

ADV_IMP.1 Implementation representation of the TSF

Developer action elements

ALC_TAT.1.1D The developer shall identify each development tool being used for the TOE.

ALC_TAT.1.2D The developer shall document the selected implementation-dependent options of each development tool.

Content and presentation elements

ALC_TAT.1.1C Each development tool used for implementation shall be well-defined.

ALC_TAT.1.2C The documentation of each development tool shall unambiguously define the meaning of all statements as well as all conventions and directives used in the implementation.

ALC_TAT.1.3C The documentation of each development tool shall unambiguously define the meaning of all implementation-dependent options.

5.3.1.21. Tests – ATE_COV.2

Dependencies

ADV_FSP.2 Security-enforcing functional specification

ATE_FUN.1 Functional testing

Developer action elements

ATE_COV.2.1D The developer shall provide an analysis of the test coverage.

Content and presentation elements

ATE_COV.2.1C The analysis of the test coverage shall demonstrate the correspondence between the tests in the test documentation and the TSFIs in the functional specification.

ATE_COV.2.2C The analysis of the test coverage shall demonstrate that all TSFIs in the functional specification have been tested.

5.3.1.22. Tests – ATE_DPT.2

Dependencies

ADV_ARC.1 Security architecture description

ADV_TDS.3 Basic modular design

ATE_FUN.1 Functional testing

Developer action elements

ATE_DPT.2.1D The developer shall provide the analysis of the depth of testing.

Content and presentation elements

ATE_DPT.2.1C The analysis of the depth of testing shall demonstrate the correspondence between the tests in the test documentation and the TSF subsystems and SFR-enforcing modules in the TOE design.

ATE_DPT.2.2C The analysis of the depth of testing shall demonstrate that all TSF subsystems in the TOE design have been tested.

ATE_DPT.2.3C The analysis of the depth of testing shall demonstrate that the SFR-enforcing modules in the TOE design have been tested.

5.3.1.23. Tests – ATE_FUN.1

Dependencies

ATE_COV.1 Evidence of coverage

Developer action elements

ATE_FUN.1.1D The developer shall test the TSF and document the results.

ATE_FUN.1.2D The developer shall provide test documentation.

Content and presentation elements

ATE_FUN.1.1C The test documentation shall consist of test plans, expected test results and actual test results.

ATE_FUN.1.2C The test plans shall identify the tests to be performed and describe the scenarios for performing each test. These scenarios shall include any ordering dependencies on the results of other tests.

ATE_FUN.1.3C The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE_FUN.1.4C The actual test results shall be consistent with the expected test results.

5.3.1.24. Tests – ATE_IND.2

Dependencies

ADV_FSP.2 Security-enforcing functional specification

AGD_OPE.1 Operational user guidance

AGD_PRE.1 Preparative procedures

ATE_COV.1 Evidence of coverage

ATE_FUN.1 Functional testing

Developer action elements

ATE_IND.2.1D The developer shall provide the TOE for testing.

Content and presentation elements

ATE_IND.2.1C The TOE shall be suitable for testing.

ATE_IND.2.2C The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

5.3.1.25. Vulnerability Assessment – AVA_VAN.3

Dependencies

ADV_ARC.1 Security architecture description

ADV_FSP.2 Security-enforcing functional specification

ADV_TDS.3 Basic modular design

ADV_IMP.1 Implementation representation of the TSF

AGD_OPE.1 Operational user guidance

AGD_PRE.1 Preparative procedures

Developer action elements

AVA_VAN.3.1D The developer shall provide the TOE for testing.

Content and presentation elements

AVA_VAN.3.1C The TOE shall be suitable for testing.

5.4. Security Requirements Rationale

5.4.1. Security Objectives Rationale

This section demonstrates how the Security Objectives trace back to the threats, OSPs and assumptions (security problem definition).

5.4.1.1. Environment Security Objectives Rationale

This section demonstrates how the Environment Security Objectives trace back to the general (base) threats, OSPs and assumptions.

Table 5-6. Mapping the Assumptions, OSPs and general (base) threats to Environment Objectives maps assumptions, OSPs and base threats to environment objectives, demonstrating that all assumptions, policies and base threats are mapped to at least one environment objective.

Assumption/Threat/OSP	Objectives
A.Configuration	OE.Configuration
A.Enhanced-Basic	OE.Enhanced-Basic
A.NO_EVIL	OE.NO_EVIL
A.PHYSICAL	OE.PHYSICAL
P.ACCESS_BANNER	OE.DISPLAY_BANNER
P.ACCOUNTABILITY	OE.AUDIT_GENERATION OE.TIME_STAMPS OE.TOE_ACCESS OE.TIME_TOE
P.CRYPTOGRAPHY	OE.CRYPTOGRAPHY
T.AUDIT_COMPROMISE	OE.AUDIT_PROTECTION OE.RESIDUAL_INFORMATION OE.SELF_PROTECTION OE.TOE_PROTECTION
T.CHANGE_TIME	OE.TIME_TOE
T.CRYPTO_COMPROMISE	OE.CRYPTOGRAPHY OE.PHYSICAL
T.MASQUERADE	OE.TOE_ACCESS
T.POOR_TEST	OE.CORRECT_TSF_OPERATION
T.RESIDUAL_DATA	OE.RESIDUAL_INFORMATION
T.TSF_COMPROMISE	OE.RESIDUAL_INFORMATION OE.SELF_PROTECTION OE.TOE_PROTECTION OE.MANAGE
T.UNATTENDED_SESSION	OE.TOE_ACCESS
T.UNAUTHORIZED_ACCESS	OE.MEDIATE
T.UNIDENTIFIED_ACTIONS	OE.AUDIT_REVIEW OE.AUDIT_GENERATION

	OE.TIME_STAMPS OE.TIME_TOE
T.DATA_TO_BE_SIGNED	OE.PHYSICAL
T.DATA_USED_FOR_VERIFYING_THE_SIGNATURE	OE.PHYSICAL
T.RESULT_OF_THE_SIGNATURE_VALIDATION	OE.PHYSICAL
T.SIGNATORY'S_IDENTITY	OE.PHYSICAL
T.SIGNED_DATA_FALSIFIED	OE.PHYSICAL

Table 5-6. Mapping the Assumptions, OSPs and general (base) threats to Environment Objectives

A.NO_EVIL states that administrators are non-hostile, appropriately trained and follow all administrator guidance. This assumption is mapped to:

- **OE.NO_EVIL**, which states that sites using the TOE will ensure that administrators are non-hostile, appropriately trained and follow all administrator guidance.

A.PHYSICAL states that environment provides the TOE with appropriate physical security, commensurate with the value of the IT assets protected by the TOE. This assumption is mapped to:

- **OE.PHYSICAL**, which states that the non-IT environment will provide an acceptable level of physical security so that the TOE cannot be tampered with or be subject to side channel attacks such as the various forms of power analysis and timing analysis.

A.Configuration states that the TOE will be properly installed and configured. This assumption is mapped to:

- **OE.Configuration**, which states that the TOE shall be installed and configured properly for starting up the TOE in a secure state.

A.Enhanced-Basic states that the attack potential on the TOE is assumed to be "Enhanced-Basic". A.Enhanced-Basic is mapped to:

- **OE.Enhanced-Basic**, which states that the TOE will be designed for a minimum attack potential of "Enhanced-Basic" as validated by the vulnerability analysis.

P.ACCESS_BANNER states that the IT Environment shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the system. This policy is mapped to:

- **OE.DISPLAY_BANNER** which states that the IT Environment will display an advisory warning regarding use of the TOE. OE.DISPLAY_BANNER satisfies this policy by ensuring that the TOE displays an administrator configurable banner that provides all interactive users with a warning about the unauthorized use of the TOE.

P.ACCOUNTABILITY states that the authorized users of the TOE shall be held accountable for their actions within the TOE. This policy is mapped to:

- **OE.AUDIT_GENERATION** which states that the IT Environment will provide the capability to detect and create records of security-relevant events associated

with users. **OE.AUDIT_GENERATION** addresses this policy by providing the administrator with the capability of configuring the audit mechanism to record the actions of a specific user, or review the audit trail based on the identity of the user. Additionally, the administrator's ID is recorded when any security relevant change is made (e.g., access rule modification, start-stop of the audit mechanism, establishment of a trusted channel, etc.).

- **OE.TIME_STAMPS** which states that the IT Environment will provide reliable time stamps and the capability for the administrator to set the time used for these time stamps. **OE.TIME_STAMPS** plays a role in supporting this policy by requiring the IT Environment to provide a reliable time stamp (configured locally by the Security Administrator or via an external NTP server). The audit mechanism is required to include the current date and time in each audit record. All audit records that include the user ID, will also include the date and time that the event occurred.

- **OE.TIME_TOE** which states that the IT Environment will provide reliable time for the TOE use. **OE.TIME_STAMPS** plays a role in supporting this policy by permitting the TOE to provide reliable time on audit records generated by the TOE.

- **OE.TOE_ACCESS** which states that the IT Environment will provide mechanisms that control a user's logical access to the TOE. **OE.TOE_ACCESS** supports this policy by requiring the IT Environment to identify and authenticate all authorized users prior to allowing any TOE access or any TOE mediated access on behalf of those users.

P.CRYPTOGRAPHY states that only NIST FIPS validated cryptography (methods and implementations) are acceptable for key management (i.e.; generation, access, distribution, destruction, handling, and storage of keys) and cryptographic services (i.e.; encryption, decryption, signature, hashing, key exchange, and random number generation services). This policy is mapped to:

- **OE.CRYPTOGRAPHY** which states The TOE shall use NIST FIPS 140-2 validated cryptographic services provided by the IT Environment. **OE.CRYPTOGRAPHY** satisfies this policy by requiring the IT Environment to implement NIST FIPS validated cryptographic services. These services will provide confidentiality and integrity services as required by the IT Environment and the TOE.

T.AUDIT_COMPROMISE states that a user or process may view audit records, cause audit records to be lost or modified, or prevent future audit records from being recorded, thus masking a user's action. This threat is mapped to:

- **OE.AUDIT_PROTECTION** which states that the IT Environment will provide the capability to protect audit information. **OE.AUDIT_PROTECT** contributes to mitigating this threat by controlling access to the audit trail. Only an administrator is allowed to read the audit trail, no one is allowed to modify audit records, the administrator is the only one allowed to delete the audit trail, and the IT Environment has the capability to prevent auditable actions from occurring if the audit trail is full.

- **OE.RESIDUAL_INFORMATION** which states that the IT Environment will ensure that any information contained in a protected resource within its Scope of Control is not released when the resource is reallocated. **OE.RESIDUAL_INFORMATION** prevents a user not authorized to read the audit trail from access to audit information that might otherwise be persistent in a resource (e.g., memory). By ensuring the IT Environment prevents residual information in a resource, audit

information will not become available to any user or process except those explicitly authorized for that data.

- **OE.SELF_PROTECTION** which states that the IT Environment will maintain a domain for its own execution that protects it and its resources from external interference, tampering, or unauthorized disclosure. **OE.SELF_PROTECTION** contributes to countering this threat by ensuring that the IT Environment can protect itself from users. If the IT Environment could not maintain and control its domain of execution, it could not be trusted to control access to the resources under its control, which includes the audit trail which are always invoked is also critical to the migration of this threat.

- **OE.TOE_PROTECTION** which states The IT Environment will protect the TOE and TOE resources from external interference, tampering, or unauthorized disclosure and modification. **OE.TOE_PROTECTION** contributes to countering this threat by ensuring that the IT Environment can protect TOE. If the TOE could not be protected, it could not be trusted to provide accurate audit information.

T.CHANGE_TIME states that an unauthorized user may change the TSF notion of time resulting in accepting old revocation information or expired certificates. This threat is mapped to:

- **OE.TIME_TOE** which states that the IT Environment will provide reliable time for the TOE use. **OE.TIME_TOE** protects against this threat by ensuring that the IT Environment does not permit users to change the time.

T.CRYPTO_COMPROMISE states that a user or process may cause key, data or executable code associated with the cryptographic functionality to be inappropriately accessed (viewed, modified, or deleted), thus compromising the cryptographic mechanisms and the data protected by those mechanisms. This threat is mapped to:

- **OE.CRYPTOGRAPHY** which states that the TOE shall use NIST FIPS 140-2 validated cryptographic services provided by the IT Environment. **OE.CRYPTOGRAPHY** protects against this threat by ensuring that the cryptography used is sound and has been validated.

- **OE.PHYSICAL** which states that the non-IT environment will provide an acceptable level of physical security so that the TOE cannot be tampered with or be subject to side channel attacks such as the various forms of power analysis and timing analysis. **OE.PHYSICAL** contributes to protection against this threat by providing physical protection from side channel attacks protects against the attempts to compromise the cryptographic mechanisms.

T.MASQUERADE states that a user or process may masquerade as another entity in order to gain unauthorized access to data or TOE resources. This threat is mapped to:

- **OE.TOE_ACCESS** which states that the IT Environment will provide mechanisms that control a user's logical access to the TOE. **OE.TOE_ACCESS** mitigates this threat by controlling the logical access to the TOE and its resources. By constraining how and when authorized users can access the TOE, and by mandating the type and strength of the authentication mechanism this objective helps mitigate the possibility of a user attempting to login and masquerade as an authorized user. In addition, this objective provides the administrator the means to control the number of failed login attempts a user can generate before an

account is locked out, further reducing the possibility of a user gaining unauthorized access to the TOE.

T.POOR_TEST states that lack of or insufficient tests to demonstrate that all TOE security functions operate correctly (including in a fielded TOE) may result in incorrect TOE behavior being undiscovered thereby causing potential security vulnerabilities. This threat is mapped to:

- **OE.CORRECT_TSF_OPERATION** which states that the IT Environment will provide the capability to test the TSF to ensure the correct operation of the TSF at a customer's site. **OE.CORRECT_TSF_OPERATION** ensures that once the TOE is installed at a customer's location, the capability exists that the integrity of the TSF (hardware and software) can be demonstrated, and thus providing end users the confidence that the TOE's security policies continue to be enforced.

T.RESIDUAL_DATA states that a user or process may gain unauthorized access to data through reallocation of TOE resources from one user or process to another. This threat is mapped to:

- **OE.RESIDUAL_INFORMATION** which states that the IT Environment will ensure that any information contained in a protected resource within its Scope of Control is not released when the resource is reallocated. **OE.RESIDUAL_INFORMATION** counters this threat by ensuring that TSF data and user data is not persistent when resources are released by one user/process and allocated to another user/process.

T.TSF_COMPROMISE states that a user or process may cause, through an unsophisticated attack, TSF data, security attributes, or executable code to be inappropriately accessed (viewed, modified, or deleted). This threat is mapped to:

- **OE.RESIDUAL_INFORMATION** which states that the IT Environment will ensure that any information contained in a protected resource within its Scope of Control is not released when the resource is reallocated. **OE.RESIDUAL_INFORMATION** is necessary to mitigate this threat, because even if the security mechanisms do not allow a user to explicitly view TSF data, if TSF data were to inappropriately reside in a resource that was made available to a user, that user would be able to inappropriately view the TSF data.

- **OE.SELF_PROTECTION** which states that the IT Environment will maintain a domain for its own execution that protects it and its resources from external interference, tampering, or unauthorized disclosure. **OE.SELF_PROTECTION** is necessary to mitigate this threat to provide the TOE a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure through its own interfaces. This feature in turn ensures that other processes cannot interfere with the IT Environment and defeat the IT Environment mechanisms.

- **OE.TOE_PROTECTION** which states that the IT Environment will protect the TOE and TOE resources from external interference, tampering, or unauthorized disclosure and modification. **OE.TOE_PROTECTION** is necessary to mitigate this threat by ensuring that the IT Environment will protect the TOE. This feature ensures that other processes cannot defeat the TOE protection mechanisms.

- **OE.MANAGE** which states that the IT Environment will provide all the functions and facilities necessary to support the administrators in their management of the

security of the TOE, and restrict these functions and facilities from unauthorized use. **OE.MANAGE** is necessary because an access control policy is not specified to control access to TSF data. This objective is used to dictate who is able to view and modify TSF data, as well as the behavior of TSF functions.

T.UNATTENDED_SESSION states that a user may gain unauthorized access to an unattended session. This threat is mapped to:

- **OE.TOE_ACCESS** which states that the IT Environment will provide mechanisms that control a user's logical access to the TOE. **OE.TOE_ACCESS** helps to mitigate this threat by including mechanisms that place controls on user's sessions. User and administrator's sessions are locked. Locking the session reduces the opportunity of someone gaining unauthorized access the session when the console is unattended.

T.UNAUTHORIZED_ACCESS states that a user may gain access to user data for which they are not authorized according to the TOE security policy. This threat is mapped to:

- **OE.MEDIATE** which states that the IT Environment will protect user data in accordance with its security policy. **OE.MEDIATE** ensures that all accesses to user data are subject to mediation, unless said data has been specifically identified as public data. The TOE requires successful authentication prior to gaining access to any controlled-access content. By implementing strong authentication to gain access to these services, an attacker's opportunity to successfully conduct a man-in-the-middle and/or password guessing attack is greatly reduced. Lastly, the IT Environment will ensure that all configured enforcement functions (authentication, access control rules, etc.) must be invoked prior to allowing a user to gain access to TOE or TOE mediated services. The IT Environment restricts the ability to modify the security attributes associated with access control rules, access to authenticated and unauthenticated services, etc to the Administrator. This feature ensures that no other user can modify the information flow policy to bypass the intended TOE security policy.

T.UNIDENTIFIED_ACTIONS states that the administrator may not have the ability to notice potential security violations, thus limiting the administrator's ability to identify and take action against a possible security breach. This threat is mapped to:

- **OE.AUDIT_REVIEW** which states that the IT Environment will provide the capability to selectively view audit information. **OE.AUDIT_REVIEW** helps to mitigate this threat by providing the Administrator with a required minimum set of configurable audit events that could indicate a potential security violation. By configuring these auditable events, the IT Environment and TOE monitors the occurrences of these events (e.g., set number of authentication failures, set number of information policy flow failures, self-test failures, etc.).

- **OE.AUDIT_GENERATION** which states that the IT Environment will provide the capability to detect and create records of security-relevant events associated with users. **OE.AUDIT_GENERATION** helps to mitigate this threat by recording actions for later review.

- **OE.TIME_STAMPS** which states that the IT Environment will provide reliable time stamps and the capability for the administrator to set the time used for these time stamps. **OE.TIME_STAMPS** helps to mitigate this threat by ensuring that audit records have correct timestamps.

- **OE.TIME_TOE** which states that the IT Environment will provide reliable time for the TOE use. **OE.TIME_STAMPS** plays a role in supporting this policy by permitting the TOE to provide reliable time on audit records generated by the TOE.

T.DATA_TO_BE_SIGNED states that an attacker modifies the data to be signed while in the process of being signed. Part of this threat is mapped to the OE.PHYSICAL objective; the other part is mapped to a TOE objective (see TOE Security Objectives Rationale section).

- **OE.PHYSICAL** which states that the non-IT environment will provide an acceptable level of physical security so that the TOE cannot be tampered with or be subject to side channel attacks such as the various forms of power analysis and timing analysis.

T.DATA_USED_FOR_VERIFYING_THE_SIGNATURE states that the data used to verify the signature do not correspond to the data presented to the verifier. Part of this threat is mapped to the OE.PHYSICAL objective; the other part is mapped to a TOE objective (see TOE Security Objectives Rationale section).

- **OE.PHYSICAL** which states that the non-IT environment will provide an acceptable level of physical security so that the TOE cannot be tampered with or be subject to side channel attacks such as the various forms of power analysis and timing analysis.

T.RESULT_OF_THE_SIGNATURE_VALIDATION states that the result of the validation of a signature does not correspond to the result presented to the verifier. Part of this threat is mapped to the OE.PHYSICAL objective; the other part is mapped to a TOE objective (see TOE Security Objectives Rationale section).

- **OE.PHYSICAL** which states that the non-IT environment will provide an acceptable level of physical security so that the TOE cannot be tampered with or be subject to side channel attacks such as the various forms of power analysis and timing analysis.

T.SIGNATORY'S_IDENTITY states that the signatory's identity does not correspond to the result presented to the verifier. Part of this threat is mapped to the OE.PHYSICAL objective; the other part is mapped to a TOE objective (see TOE Security Objectives Rationale section).

- **OE.PHYSICAL** which states that the non-IT environment will provide an acceptable level of physical security so that the TOE cannot be tampered with or be subject to side channel attacks such as the various forms of power analysis and timing analysis.

T.SIGNED_DATA_FALSIFIED states that the signed data established by the verifier can be falsified. Part of this threat is mapped to the OE.PHYSICAL objective; the other part is mapped to a TOE objective (see TOE Security Objectives Rationale section).

- **OE.PHYSICAL** which states that the non-IT environment will provide an acceptable level of physical security so that the TOE cannot be tampered with or be subject to side channel attacks such as the various forms of power analysis and timing analysis.

Table 5-7. Mapping the Environment Objectives to Assumptions, OSPs and general (base) threats maps environment objectives to assumptions, OSPs and base threats, demonstrating that all assumptions, policies and base threats are mapped to at least one environment objective.

Objectives	Assumption/Threat/OSP
OE.AUDIT_GENERATION	P.ACCOUNTABILITY T.UNIDENTIFIED_ACTIONS
OE.AUDIT_PROTECTION	T.AUDIT_COMPROMISE
OE.AUDIT_REVIEW	T.UNIDENTIFIED_ACTIONS
OE.Configuration	A.Configuration
OE.CORRECT_TSF_OPERATION	T.POOR_TEST
OE.CRYPTOGRAPHY	P.CRYPTOGRAPHY T.CRYPTO_COMPROMISE
OE.DISPLAY_BANNER	P.ACCESS_BANNER
OE.Enhanced-Basic	A.Enhanced-Basic
OE.MANAGE	T.TSF_COMPROMISE
OE.MEDIATE	T.UNAUTHORIZED_ACCESS
OE.NO_EVIL	A.NO_EVIL
OE.PHYSICAL	A.PHYSICAL T.CRYPTO_COMPROMISE T.DATA_TO_BE_SIGNED T.DATA_USED_FOR_VERIFYING_THE_SIGNATURE T.RESULT_OF_THE_SIGNATURE_VALIDATION T.SIGNATORY'S_IDENTITY T.SIGNED_DATA_FALSIFIED
OE.RESIDUAL_INFORMATION	T.AUDIT_COMPROMISE T.RESIDUAL_DATA T.TSF_COMPROMISE
OE.SELF_PROTECTION	T.AUDIT_COMPROMISE T.TSF_COMPROMISE

OE.TIME_STAMPS	P.ACCOUNTABILITY; T.UNIDENTIFIED_ACTIONS
OE.TIME_TOE	P.ACCOUNTABILITY T.CHANGE_TIME T.UNIDENTIFIED_ACTIONS
OE.TOE_ACCESS	P.ACCOUNTABILITY T.MASQUERADE T.UNATTENDED_SESSION
OE.TOE_PROTECTION	T.AUDIT_COMPROMISE T.TSF_COMPROMISE

Table 5-7. Mapping the Environment Objectives to Assumptions, OSPs and general (base) threats

5.4.1.2. TOE Security Objectives Rationale

This section demonstrates how the TOE Security Objectives trace back to the specific threats. The assumptions, OSPs and base threats have already been traced to objectives (environment objectives) in the previous section and therefore this section only manages the specific threats and TOE objectives.

CPV – Basic Package Security Objectives Rationale

The following tables demonstrate the mapping of threats to objectives and objectives to threats for the CPV – Basic package. Explanatory text is provided below the tables to support the mapping.

Threat	Objectives
T.Certificate_Modi	O.Verified_Certificate
T.DOS_CPV_Basic	O.Availability
T.Expired_Certificate	O.Correct_Temporal O.Current_Certificate
T.Untrusted_CA	O.Trusted_Keys
T.No_Crypto	O.Get_KeyInfo
T.Path_Not_Found	O.Path_Find
T.Revoked_Certificate	O.Valid_Certificate
T.User_CA	O.User

Table 5-8. Mapping of Threats to Objectives for CPV – Basic Package

T.Certificate_Modi states that an untrusted user may modify a certificate resulting in using a wrong public key. This threat is mapped to:

- **O.Verified_Certificate**, which states that the TSF shall only accept certificates with verifiable signatures.

T.DOS_CPV_Basic states that the revocation information or access to revocation information could be made unavailable, resulting in loss of system availability. This threat is mapped to:

- **O.Availability**, which states that the TSF shall continue to provide security services even if revocation information is not available.

T.Expired_Certificate states that an expired (and possibly revoked) certificate as of TOI could be used for signature verification. This threat is mapped to:

- **O.Correct_Temporal**, which states that the TSF shall provide accurate temporal validation results.

- **O.Current_Certificate**, which states that the TSF shall only accept certificates that are not expired as of TOI.

T.Untrusted_CA states that an untrusted entity (Certification Authority (CA)) may issue certificates to bogus entities, permitting those entities to assume identity of other legitimate users. This threat is mapped to:

- **O.Trusted_Keys**, which states that the TSF shall use trusted public keys in certification path validation.

T.No_Crypto states that the user public key and related information may not be available to carry out the cryptographic function. This threat is mapped to:

- **O.Get_KeyInfo**, which states that the TSF shall provide the user public key and related information in order to carry out cryptographic functions.

T.Path_Not_Found states that a valid certification path is not found due to lack of system functionality. This threat is mapped to:

- **O.Path_Find**, which states that the TSF shall be able to find a certification path from a trust anchor to the subscriber.

T.Revoked_Certificate states that a revoked certificate could be used as valid, resulting in security compromise. This threat is mapped to:

- O.Valid_Certificate, which states that the TSF shall use certificates that are valid, i.e., not revoked.

T.User_CA states that a user could act as a CA, issuing unauthorized certificates. This threat is mapped to:

- **O.User**, which states that the TSF shall only accept certificates issued by a CA.

Table 5-9. Mapping of Objectives to Threats for CPV – Basic Package maps objectives for the CPV – Basic Package to threats, demonstrating that every objective is mapped to a threat. The mapping is described in the text above and is not repeated following this Table.

Objective	Threats
O.Availability	T.DOS_CPV_Basic
O.Correct_Temporal	T.Expired_Certificate
O.Current_Certificate	T.Expired_Certificate
O.Get_KeyInfo	T.No_Crypto
O.Path_Find	T.Path_Not_Found
O.Trusted_Keys	T.Untrusted_CA
O.User	T.User_CA
O.Verified_Certificate	T.Certificate_Modi
O.Valid_Certificate	T.Revoked_Certificate

Table 5-9. Mapping of Objectives to Threats for CPV – Basic Package

CPV – Basic Policy Package Security Objectives Rationale

The mapping of threats to objectives for the CPV – Basic Policy package is shown in Table 5-10. Mapping of Threats to Objectives for CPV – Basic Policy Package. Text that further supports for the mapping is provided following this table.

Threat	Objectives
T.Unknown_Policies	O.Provide_Policy_Info

Table 5-10. Mapping of Threats to Objectives for CPV – Basic Policy Package

T.Unknown_Policies states that the user may not know the policies under which a certificate was issued. This threat is mapped to:

- **O.Provide_Policy_Info**, which states that the TSF shall provide certificate policies for which the certification path is valid.

Table 5-11. Mapping of Objectives to Threats for CPV – Basic Policy Package maps objectives for the CPV – Basic Package to threats, demonstrating that every objective is mapped to a threat. The mapping is described in the text above and is not repeated following this Table.

Objective	Threats
O.Provide_Policy_Info	T.Unknown_Policies

Table 5-11. Mapping of Objectives to Threats for CPV – Basic Policy Package

PKI Signature Generation Package Security Objectives Rationale

The mapping of threats to objectives for the PKI Signature Generation package is shown in Table 5-12. Mapping of Threats to Objectives for the PKI Signature Generation Package. Text that further supports for the mapping is provided following this table.

Threat	Objectives
T.Clueless_PKI_Sig	O.Give_Sig_Hints

Table 5-12. Mapping of Threats to Objectives for the PKI Signature Generation Package

T.Clueless_PKI_Sig states that the user may try only inappropriate certificates for PKI signature verification because the signature does not include a hint. This threat is addressed by:

- **O.Give_Sig_Hints**, which states that the TSF shall give hints for selecting correct certificates or keys for PKI signature.

Table 5-13. Mapping of Objectives to Threats for the PKI Signature Generation Package maps objectives for the PKI Signature Generation package to threats, demonstrating that every objective is mapped to a threat. The mapping is described in the text above and is not repeated following this table.

Objective	Threats
O.Give_Sig_Hints	T.Clueless_PKI_Sig

Table 5-13. Mapping of Objectives to Threats for the PKI Signature Generation Package

PKI Signature Verification Package Security Objectives Rationale

The mapping of threats to objectives for the PKI Signature Verification package is shown in Table 5-14. Mapping of Threats to Objectives for the PKI Signature Verification Package. Text that further supports for the mapping is provided following this table.

Threat	Objectives
T.Assumed_Identity_PKI_Ver	O.Linkage_Sig_Ver
T.Clueless_PKI_Ver	O.Use_Sig_Hints

Table 5-14. Mapping of Threats to Objectives for the PKI Signature Verification Package

T.Assumed_Identity_PKI_Ver states that a user may assume the identity of another user for PKI signature verification. This threat is addressed by:

- **O.Linkage_Sig_Ver**, which states that the TSF shall use the correct user public key for signature verification.

T.Clueless_PKI_Ver states that the user may try only inappropriate certificates for PKI signature verification by ignoring hints in the signature. This threat is addressed by:

- **O.Use_Sig_Hints**, which states that the TSF shall provide hints for selecting correct certificates or keys for signature verification.

Table 5-15. Mapping of Objectives to Threats for the PKI Signature Verification Package maps objectives The PKI Signature Verification package to threats, demonstrating that every objective is mapped to a threat. The mapping is described in the text above and is not repeated following this table.

Objective	Threats
O.Linkage_Sig_Ver	T.Assumed_Identity_PKI_Ver
O.Use_Sig_Hints	T.Clueless_PKI_Ver

Table 5-15. Mapping of Objectives to Threats for the PKI Signature Verification Package

PKI Encryption using Key Transfer Algorithms Package Security Objectives Rationale

The mapping of threats to objectives for all of PKI Encryption using Key Transfer Algorithms package is shown in Table 5-16. Mapping of Threats to Objectives for the PKI Encryption using Key Transfer Algorithms Package. Text that further supports for the mapping is provided following this table.

Threat	Objectives
T.Assumed_Identity_WO_En	O.Linkage_Enc_WO
T.Clueless_WO_En	O.Hints_Enc_WO

Table 5-16. Mapping of Threats to Objectives for the PKI Encryption using Key Transfer Algorithms Package

T.Assumed_Identity_WO_ En states that a user may assume the identity of another user in order to perform encryption using Key Transfer algorithms. This threat is addressed by:

- **O.Linkage_Enc_WO**, which states that the TSF shall use the correct user public key for key transfer.

T.Clueless_WO_En states that the user may try only inappropriate certificates in absence of hint for encryption using Key Transfer algorithms. This threat is addressed by:

- **O.Hints_Enc_WO**, which states that the TSF shall provide hints for selecting correct certificates or keys for PKI Encryption using Key Transfer algorithms.

Table 5-17. Mapping of Objectives to Threats for the PKI Encryption using Key Transfer Algorithms Package maps objectives for the PKI Encryption using Key Transfer

Algorithms package to threats, demonstrating that every objective is mapped to a threat. The mapping is described in the text above and is not repeated following this table.

Objective	Threats
O.Linkage_Enc_WO	T.Assumed_Identity_WO_En
O.Hints_Enc_WO	T.Clueless_WO_En

Table 5-17. Mapping of Objectives to Threats for the PKI Encryption using Key Transfer Algorithms Package

PKI Decryption using Key Transfer Algorithms Package Security Objectives Rationale

The mapping of threats to objectives for the PKI Decryption using Key Transfer Algorithms package is shown in Table 5-18. Mapping of Threats to Objectives for the PKI Decryption using Key Transfer Algorithms Package. Text that further supports for the mapping is provided following this table.

Threat	Objectives
T.Garble_WO_De	O.Correct_KT

Table 5-18. Mapping of Threats to Objectives for the PKI Decryption using Key Transfer Algorithms Package

T.Garble_WO_De states that the user may not apply the correct key transfer algorithm or private key, resulting in garbled data. This threat is addressed by:

- **O.Correct_KT**, which states that the TSF shall use appropriate private key and key transfer algorithm.

Table 5-19. Mapping of Objectives to Threats for the PKI Decryption using Key Transfer Algorithms Package maps objectives for the PKI Decryption using Key Transfer Algorithms package to threats, demonstrating that every objective is mapped to a threat. The mapping is described in the text above and is not repeated following this table.

Objective	Threats
O.Correct_KT	T.Garble_WO_De

Table 5-19. Mapping of Objectives to Threats for the PKI Decryption using Key Transfer Algorithms Package

PKI Based Entity Authentication Package Security Objective Rationale

The mapping of threats to objectives for the PKI Based Entity Authentication package is shown in Table 5-20. Mapping of Threats to Objectives for PKI Based Entity

Authentication Package. Text that further supports the mapping is provided following this table.

Threat	Objectives
T.Assumed_Identity_Auth	O.Linkage, O.I&A, O.Limit_Actions_Auth
T.Replay_Entity	O.Single_Use_I&A

Table 5-20. Mapping of Threats to Objectives for PKI Based Entity Authentication Package

T.Assumed_Identity_Auth states that a user may assume the identity of another user to perform entity based authentication. This threat is addressed by:

- **O.Linkage**, which states that the TSF shall use the correct user public for authentication.
- **O.I&A**, which states that the TSF shall uniquely identify all entities, and shall authenticate the claimed identify before granting an entity access to the TOE facilities.
- **O.Limit_Actions_Auth**, which states that the TSF shall restrict the actions an entity may perform before the TSF verifies the identity of the entity.

T.Replay_Entity states that an unauthorized user may replay valid authentication data. This threat is addressed by:

- **O.Single_Use_I&A**, which states that the TSF shall use the I&A mechanism that requires unique authentication information for each I&A.

Table 5-21. Mapping of Objectives to Threats for PKI Based Entity Authentication Package maps objectives for the PKI Based Entity Authentication Package to threats, demonstrating that every objective is mapped to a threat. The mapping is described in the text above and is not repeated following this table.

Objective	Threats
O.I&A	T.Assumed_Identity_Auth
O.Limit_Actions_Auth	T.Assumed_Identity_Auth
O.Linkage	T.Assumed_Identity_Auth
O.Single_Use_I&A	T.Replay_Entity

Table 5-21. Mapping of Objectives to Threats for PKI Based Entity Authentication Package

OCSP Package Security Objectives Rationale

The mapping of threats to objectives for the OCSP package is shown in Table 5-22. Mapping of Threats to Objectives for the OCSP Package. Text that further supports the mapping is provided following this table.

Threat	Objectives

T.DOS_OCSP	O.User_Override_Time_OCSP
T.Replay_OCSP_Info	O.Current_OCSP_Info
T.Wrong_OCSP_Info	O.Accurate_OCSP_Info, O.Auth_OCSP_Info

Table 5-22. Mapping of Threats to Objectives for the OCSP Package

T.DOS_OCSP states that the OCSP response or access to the OCSP response could be made unavailable, resulting in loss of system availability. This threat is mapped to:

- **O.User_Override_Time_OCSP**, which states that the TSF shall permit the user to override the time checks on the OCSP response.

T.Replay_OCSP_Info states that the user may accept revocation information from well before TOI resulting in accepting revoked certificate for OCSP transactions. This threat is mapped to:

- **O.Current_OCSP_Info**, which states that the TSF accept only OCSP responses current as of TOI.

T.Wrong_OCSP_Info states that the user may accept a revoked certificate or reject a valid certificate due to wrong revocation information. This threat is mapped to:

- **O.Accurate_OCSP_Info**, which states that the TSF shall accept only accurate OCSP responses.
- **O.Auth_OCSP_Info**, which states that the TSF shall accept the OCSP response from an authorized source.

Table 5-23. Mapping of Objectives to Threats for the OCSP Package maps objectives for the OCSP package to threats, demonstrating that every objective is mapped to a threat. The mapping is described in the text above and is not repeated following this table.

Objective	Threats
O.Accurate_OCSP_Info	T.Wrong_OCSP_Info
O.Auth_OCSP_Info	T.Wrong_OCSP_Info
O.Current_OCSP_Info	T.Replay_OCSP_Info
O.User_Override_Time_OCSP	T.DOS_OCSP

Table 5-23. Mapping of Objectives to Threats for the OCSP Package

CRL Verification Package Security Objectives Rationale

The mapping of threats to objectives for the CRL Verification package is shown in Table 5-24. Mapping of Threats to Objectives for CRL Verification Package. Text that further supports for the mapping is provided following this table.

Threat	Objectives
T.DOS_CRL	O.User_Override_Time_CRL
T.Replay_Revoc_Info_CRL	O.Current_Rev_Info
T.Wrong_Revoc_Info_CRL	O.Accurate_Rev_Info O.Auth_Rev_Info

Table 5-24. Mapping of Threats to Objectives for CRL Verification Package

T.DOS_CRL states that the CRL or access to the CRL could be made unavailable, resulting in loss of system availability. This threat is mapped to:

- **O.User_Override_Time_CRL**, which states that the TSF shall permit the user to override the time checks on the CRL.

T.Replay_Revoc_Info_CRL states that the user may accept a CRL issued well before TOI resulting in accepting currently revoked certificate. This threat is mapped to:

- **O.Current_Rev_Info**, which states that the TSF shall accept only CRL that are current as TOI.

T.Wrong_Revoc_Info_CRL states that the user may accept a revoked certificate or reject a valid certificate due to wrong revocation information. This threat is mapped to:

- **O.Accurate_Rev_Info**, which states that the TSF shall accept only accurate revocation information.
- **O.Auth_Rev_Info**, which states that the TSF shall accept the revocation information from an authorized source for CRL.

Table 5-25. Mapping of Objectives to Threats for the CRL Verification Package maps objectives for the CRL Verification package to threats, demonstrating that every objective is mapped to a threat. The mapping is described in the text above and is not repeated following this table.

Objective	Threats
O.Accurate_Rev_Info	T.Wrong_Revoc_Info_CRL
O.Auth_Rev_Info	T.Wrong_Revoc_Info_CRL
O.Current_Rev_Info	T.Replay_Revoc_Info_CRL
O.User_Override_Time_CRL	T.DOS_CRL

Table 5-25. Mapping of Objectives to Threats for the CRL Verification Package

Audit Package Security Objectives Rationale

The mapping of threats to objectives for the Audit package is shown in Table 5-26. Mapping of Objectives to Threats for Audit Package. Text that further supports for the mapping is provided following this table.

Threat	Objectives
O.PKE_Audit	T.PKE_Accountability

Table 5-26. Mapping of Objectives to Threats for Audit Package

T.PKE_Accountability states that the PKE related audit events cannot be linked to individual actions. This threat is mapped to:

- **O.PKE_Audit**, which states that the TSF shall audit security relevant PKE events. This coupled with the base audit functions provided by the IT Environment mitigate this threat.

Table 5-27. Mapping of Threats to Objectives for Audit Package maps objectives for the Audit package to threats, demonstrating that every objective is mapped to a threat. The mapping is described in the text above and is not repeated following this table.

Objective	Threats
T.PKE_Accountability	O.PKE_Audit

Table 5-27. Mapping of Threats to Objectives for Audit Package

Continuous Authentication Package Security Objective Rationale

The mapping of threats to objectives for the Continuous Authentication package is shown in Table 5-28. Mapping of Threats to Objectives for Continuous Authentication Package. Text that further supports the mapping is provided following this table.

Threat	Objectives
T.Hijack	O.Continuous_I&A

Table 5-28. Mapping of Threats to Objectives for Continuous Authentication Package

T.Hijack states that an unauthorized user may hijack an authenticated session. This threat is addressed by:

- **O.Continuous_I&A**, which states that the TSF shall continuously authenticate the entity.

Table 5-29. Mapping of Objectives to Threats for Continuous Authentication Package maps objectives for the Continuous Authentication Package to threats, demonstrating that every objective is mapped to a threat. The mapping is described in the text above and is not repeated following this table.

Objective	Threats
O.Continuous_I&A	T.Hijack

Table 5-29. Mapping of Objectives to Threats for Continuous Authentication Package

Annex III of the [EUROPEAN_DIRECTIVE] Security Objectives Rationale

The following tables list the mapping of threats and OSPs to objectives and objectives to threats and OSPs, for those security objectives derived from Annex III of the [EUROPEAN_DIRECTIVE]. Explanations for the mappings are provided below the tables.

Threat/OSP	Objectives
T.DISCLOSURE_OR_NOT_UNIQUE_PRIVATE _KEYS	O.DIRECTIVE_ANNEX_III_PART_1a
T.ILEGITIMATE_USE_OF_PRIVATE_KEYS	O.DIRECTIVE_ANNEX_III_PART_1c
T.DATA_TO_BE_SIGNED	O.DIRECTIVE_ANNEX_III_PART_2
T.SIGNATURE_FALSIFIED	O.DIRECTIVE_ANNEX_III_PART_1b

Table 5-30. Mapping of Threats and OSPs to Objectives derived from Annex III of the [EUROPEAN_DIRECTIVE]

T.SIGNATURE_FALSIFIED states that the signatures can be falsified and the private keys can be derived using currently-available technology.

- **O.DIRECTIVE_ANNEX_III_PART_1b**, which states that the TOE guarantees the operation of its services in an execution environment with a FIPS 140-2 level 3 security level for the following processes: assuring that the private key cannot be derived using currently-available technology and assuring that the signatures cannot been falsified.

  The first part of the threat (that signatures can be falsified) is mitigated by the second part of the objective (the TOE guarantees the operation of its services in an execution environment with a FIPS 140-2 level 3 security level for assuring that the signatures cannot been falsified).

  The second part of the threat (the private keys can be derived using the currently-available technology) is mitigated by the first part of the objective (the TOE guarantees the operation of its services in an execution environment with a FIPS 140-2 level 3 security level for assuring that the private key cannot be derived using currently-available technology).

T.DISCLOSURE_OR_NOT_UNIQUE_PRIVATE_KEYS states that a private key is improperly disclosed or it can be obtained again (it is not assured that the signature-creation data used for signature generation can, practically, occur only once). This threat is mapped to:

- **O.DIRECTIVE_ANNEX_III_PART_1a,** which states that the TOE guarantees the operation of its services in an execution environment with a FIPS 140-2 level 3 security level for the following processes: assuring the secure custody of the private key and the uniqueness property for the private key (the signature-creation data used for signature generation can, practically, occur only once).

The first part of the threat (the private key is improperly disclosed) is mitigated by the first part of the objective (the TOE guarantees the operation of its services in an execution environment with a FIPS 140-2 level 3 security level for secure custody of the private key).

The second part of the threat (the private key can be obtained again) is mitigated by the second part of the objective (the TOE guarantees the operation of its services in an execution environment with a FIPS 140-2 level 3 security level for the uniqueness property for the private key).

T.ILEGITIMATE_USE_OF_PRIVATE_KEYS states that an attacker can access the private keys of users to generate signatures without the legitimate signer consent. This threat is mapped to:

- **O.DIRECTIVE_ANNEX_III_PART_1c**, which states that the TOE must ensure that the signature-creation data used for signature generation can be reliably protected by the legitimate signer against being used by others.

The TOE assures this objective on the basis of the following protections:

- Use of access control policies and rules that implement these policies, which contribute to the protection of private keys.
- Use of authentication and identification mechanisms to guarantee the controlled access to the private keys.
- Use of mechanisms that impose strict control on the cryptographic devices for the storing of the private keys.
- Guaranteeing that the execution environment has a FIPS 140-2 level 3 security level for the secure custody of the private keys.

T.DATA_TO_BE_SIGNED states that an attacker modifies the data to be signed while it is in the process of being signed. This threat is mapped to:

- **O.DIRECTIVE_ANNEX_III_PART_2**, which states that the TOE must provide integrity to the data to be signed.

The TOE provides technical mechanisms to assure this integrity in the path between the user requesting the signature service and the TOE. In the TOE's context, the integrity is assured via the A.PHYSICAL assumption that assumes that the environment provides the TOE with appropriate physical security. Table 5-31. Mapping of Objectives derived from Annex III of the [EUROPEAN_DIRECTIVE] to Threats/OSPs showing that every objective is mapped to a threat/OSP. The mapping is described text above and is not repeated below this Table.

Objective	Threats/OSPs
O.DIRECTIVE_ANNEX_III_PART_1a	T.DISCLOSURE_OR_NOT_UNIQUE_PRIVATE _KEYS
O.DIRECTIVE_ANNEX_III_PART_1b	T.SIGNATURE_FALSIFIED
O.DIRECTIVE_ANNEX_III_PART_1c	T.ILEGITIMATE_USE_OF_PRIVATE_KEYS
O.DIRECTIVE_ANNEX_III_PART_2	T.DATA_TO_BE_SIGNED

Table 5-31. Mapping of Objectives derived from Annex III of the [EUROPEAN_DIRECTIVE] to Threats/OSPs

The T.DISCLOSURE_OR_NOT_UNIQUE_PRIVATE_KEYS, T.ILEGITIMATE_USE_OF_PRIVATE_KEYS and T.SIGNATURE_FALSIFIED threats are mitigated by the corresponding TOE objectives, which are included in this section and that guarantee the operation of the TOE's services in an execution environment with a FIPS 140-2 level 3 security level. Note that the TOE guarantees the execution of the services in a FIPS 140-2 Level 3 cryptographic module, which provides the mechanisms to mitigate the threats.

Annex IV of the [EUROPEAN_DIRECTIVE] Security Objectives Rationale

The following tables show the mapping of threats/OSPs to objectives and objectives to threats/OSPs for those security objectives derived from Annex IV of the [EUROPEAN_DIRECTIVE]. Explanations for the mappings are provided below the tables

Threat	Objectives
T.DATA_USED_FOR_VERIFYING_THE_SIGNATURE	O.DIRECTIVE_ANNEX_IV_PART_a
T.RESULT_OF_THE_SIGNATURE_VALIDATION	O.DIRECTIVE_ANNEX_IV_PART_b
T.SIGNATORY'S_IDENTITY	O.DIRECTIVE_ANNEX_IV_PART_e
T.SECURITY_AUDIT_EVENTS	O.DIRECTIVE_ANNEX_IV_PART_g
T.INVALID_CERTIFICATE	O.DIRECTIVE_ANNEX_IV_PART_d
T.SIGNATURE_NOT_RELIABLY_VERIFIED	O.DIRECTIVE_ANNEX_IV_PART_b
T.SIGNED_DATA_FALSIFIED	O.DIRECTIVE_ANNEX_IV_PART_c
T.PSEUDONYM_NOT_SUPPORTED	O.DIRECTIVE_ANNEX_IV_PART_f

Table 5-32. Mapping of Threats/OSPs to Objectives derived from Annex IV of the [EUROPEAN_DIRECTIVE]

T.INVALID_CERTIFICATE states that the authenticity and validity of the certificate required at the time of signature verification cannot be reliably verified. This threat is mapped to:

- **O.DIRECTIVE_ANNEX_IV_PART_d,** which states that the TOE must verify the authenticity and validity of the certificate required at the time of signature verification. The TOE assures this objective by implementing the following security properties:
  - Guaranteeing the operation of its services in an execution environment with a FIPS 140-2 level 3 security level. This assures that the validation of the signature included in the certificate is done in a suitable security environment.
  - Verifying all the information related to the certificate (building of certification paths, processing of the X.509 extensions and all aspects relating to the

[X.509] standard) and invoking cryptographic hardware for the cryptographic verification of the signatures related to the validation of the certificate.

T.DATA_USED_FOR_VERIFYING_THE_SIGNATURE states that the data used to verify the signature does not correspond to the data presented to the verifier. This threat is mapped to:

- **O.DIRECTIVE_ANNEX_IV_PART_a**, which states that during the signature-verification process, the TOE must ensure with reasonable certainty that the data used for verifying the signature corresponds to the data displayed to the verifier. The TOE mitigates this threat (through achieving the O.DIRECTIVE_ANNEX_IV_PART_a objective) by implementing an integrity service for the data exchanged between the user requesting digital signature generation and the TOE. In the TOE's context, the integrity is assured via the A.PHYSICAL assumption that assumes that the environment provides the TOE with appropriate physical security.

T.RESULT_OF_THE_SIGNATURE_VALIDATION states that the result of the validation of a signature does not correspond to the result presented to the verifier. This threat is mapped to:

- **O.DIRECTIVE_ANNEX_IV_PART_b**, which states that during the signature-verification process, the TOE must ensure with reasonable certainty that the result of that validation is correctly displayed (this objective also states that the signature is to be reliably verified). The TOE mitigates this threat (through achieving the O.DIRECTIVE_ANNEX_IV_PART_b objective) by implementing an integrity service for the data exchanged between the user demanding the generation of a digital signature and the TOE. In the TOE's context, the integrity is assured via the A.PHYSICAL assumption that assumes that the environment provides the TOE with appropriate physical security.

T.SIGNATORY'S_IDENTITY states that the signer's identity does not correspond to the result presented to the verifier. This threat is mapped to:

- **O.DIRECTIVE_ANNEX_IV_PART_e**, which states that during the signature-verification process, the TOE must ensure with reasonable certainty that the signer's identity is correctly displayed. The TOE mitigates this threat (through achieving the O.DIRECTIVE_ANNEX_IV_PART_e objective) by implementing an integrity service for the data exchanged between the TOE and the user requesting digital signature generation. In the TOE's context, the integrity is assured via the A.PHYSICAL assumption that assumes that the environment provides the TOE with appropriate physical security.

T.SECURITY_AUDIT_EVENTS states that the security-relevant changes are not detected. This threat is mapped to:

- **O.DIRECTIVE_ANNEX_IV_PART_g**, which states that during the signature-verification process, the TOE must ensure with reasonable certainty that any security-relevant changes can be detected. The TOE assures this objective by implementing mechanisms for login and detecting security events.

T.SIGNATURE_NOT_RELIABLY_VERIFIED states that the signatures can be not reliably verified. This threat is mapped to:

- **O.DIRECTIVE_ANNEX_IV_PART_b**, which states that during the signature-verification process, the TOE must ensure with reasonable certainty that the signature can be reliably verified (this objective also states that the result of this validation is to be displayed correctly). The TOE assures this objective by implementing the following security properties:

  - Guaranteeing the operation of its services in an execution environment with a FIPS 140-2 level 3 security level. This assures that signature validation is done in a suitable security environment.

  - Verifying all the information on the signature (building of certification paths, processing of the X.509 extensions and all aspects relating to the [X.509] standard) and invoking cryptographic hardware for the cryptographic verification of the signatures related to the validation of the certificate.

T.SIGNED_DATA_FALSIFIED states that the signed data established by the verifier can be falsified. This threat is mapped to:

- **O.DIRECTIVE_ANNEX_IV_PART_c**, which states that during the signature-verification process, the TOE must ensure with reasonable certainty that the signed data established by the verifier is not falsified. The TOE assures this objective by implementing an integrity service for the data exchanged between the verifier and the TOE. In the TOE's context, the integrity is assured via the A.PHYSICAL assumption that assumes that the environment provides the TOE with appropriate physical security.

T.PSEUDONYM_NOT_SUPPORTED states that it is not assured, at the time of the signature verification, that a pseudonym can be indicated. This threat is mapped to:

- **O.DIRECTIVE_ANNEX_IV_PART_f**, which states that during the signature-verification process, the TOE must ensure with reasonable certainty that the use of a pseudonym is supported. The TOE assures this objective by supporting the use of X.509 public key certificates in the signature verification process (the X.509 certificates support extensions and attributes in which it is possible to indicate all type of information, including pseudonyms).

Table 5-33. Mapping of Objectives derived from Annex IV of the [EUROPEAN_DIRECTIVE] to Threats/OSPs, in which every objective is mapped to a threat. The mapping is described above and is not repeated below this Table.

Objective	Threats/OSPs
O.DIRECTIVE_ANNEX_IV_PART_a	T.DATA_USED_FOR_VERIFYING_THE_SIGNATURE
O.DIRECTIVE_ANNEX_IV_PART_b	T.RESULT_OF_THE_SIGNATURE_VALIDATION T.SIGNATURE_NOT_RELIABLY_VERIFIED
O.DIRECTIVE_ANNEX_IV_PART_c	T.SIGNED_DATA_FALSIFIED
O.DIRECTIVE_ANNEX_IV_PART_d	T.INVALID_CERTIFICATE
O.DIRECTIVE_ANNEX_IV_PART_e	T.SIGNATORY'S_IDENTITY
O.DIRECTIVE_ANNEX_IV_PART_f	T.PSEUDONYM_NOT_SUPPORTED

O.DIRECTIVE_ANNEX_IV_PART_g	T.SECURITY_AUDIT_EVENTS

Table 5-33. Mapping of Objectives derived from Annex IV of the [EUROPEAN_DIRECTIVE] to Threats/OSPs

The T.INVALID_CERTIFICATE and T.SIGNATURE_NOT_RELIABLY_VERIFIED threats are mitigated by the corresponding TOE objectives that are included in this section and that guarantee the operation of the TOE's services in an execution environment with a FIPS 140-2 level 3 security level. Note that the TOE guarantees the execution of the services in a FIPS 140-2 Level 3 cryptographic module, which provides the mechanisms to mitigate the threats.

5.4.2. Security Requirements Rationale

In this section, the objectives are mapped to the functional requirements and rationale is provided for the selected assurance level and its components and augmentation.

5.4.2.1. Functional Security Requirements Rationale

The mapping of all security objectives to functional requirements (components) or to assumptions is provided in Table 5-34. Security Objective to Functional Component Mapping. Rationale for the IT Environment functional requirements mapping and for each package are described in separate subsections.

Objective	Functional Components
Mapping for CPV – Basic Package	
O.Availability	FDP_DAU_CPV_(EXT).1
O.Correct_Temporal	FDP_DAU_CPI_(EXT).1
O.Current_Certificate	FDP_DAU_CPV_(EXT).1
O.Get_KeyInfo	FDP_DAU_CPO_(EXT).1
O.Path_Find	FDP_CPD_(EXT).1
O.Trusted_Keys	FDP_DAU_CPI_(EXT).1
O.User	FDP_DAU_CPV_(EXT).2
O.Verified_Certificate	FDP_DAU_CPV_(EXT).1
O.Valid_Certificate	FDP_DAU_CPV_(EXT).1
Mapping for CPV – Basic Policy Package	
O.Provide_Policy_Info	FDP_DAU_CPI_(EXT).2, FDP_DAU_CPO_(EXT).2

Mapping for PKI Signature Generation Package	
O.Give_Sig_Hints	FDP_ETC_SIG_(EXT).1
Mapping for PKI Signature Verification Package	
O.Use_Sig_Hints	FDP_ITC_SIG_(EXT).1
O.Linkage_Sig_Ver	FDP_DAU_SIG_(EXT).1
Mapping for PKI Encryption using Key Transfer Algorithms Package	
O.Hints_Enc_WO	FDP_ETC_ENC_(EXT).1
O.Linkage_Enc_WO	FDP_ETC_ENC_(EXT).1 FDP_DAU_ENC_(EXT).1
Mapping for PKI Decryption using Key Transfer Algorithms Package	
O.Correct_KT	FDP_ITC_ENC_(EXT).1
Mapping for PKI Based Entity Authentication Package	
O.I&A	FIA_UAU.1 FIA_UID.1
O.Limit_Actions_Auth	FIA_UAU.1 FIA_UID.1
O.Linkage	FIA_UAU_SIG_(EXT).1
O.Single_Use_I&A	FIA_UAU.4
Mapping for Online Certificate Status Protocol Client Package	
O.Accurate_OCSP_Info	FDP_DAU_OCS_(EXT).1
O.Auth_OCSP_Info	FDP_DAU_OCS_(EXT).1
O.Current_OCSP_Info	FDP_DAU_OCS_(EXT).1
O.User_Override_Time_OCSP	FDP_DAU_OCS_(EXT).1
Mapping for Certificate Revocation List (CRL) Validation Package	
O.Accurate_Rev_Info	FDP_DAU_CRL_(EXT).1
O.Auth_Rev_Info	FDP_DAU_CRL_(EXT).1
O.Current_Rev_Info	FDP_DAU_CRL_(EXT).1
O.User_Override_Time_CRL	FDP_DAU_CRL_(EXT).1
Mapping for Audit Package	

O.PKE_Audit	FAU_GEN.1-NIAP-0407:2 FAU_GEN.2-NIAP-0410:2
Mapping for Continuous Authentication Package	
O.Continuous_I&A	FIA_UAU.6
Mapping for Objectives derived from the Annex III of the [EUROPEAN_DIRECTIVE]	
O.DIRECTIVE_ANNEX_III_PART_1a	FPT_TEE.1
O.DIRECTIVE_ANNEX_III_PART_1b	FPT_TEE.1
O.DIRECTIVE_ANNEX_III_PART_1c	FPT_TEE.1 FDP_ACC.1 FDP_ACF.1 FIA_UAU.1 FIA_UID.1 FMT_MSA.1 FMT_MSA.3 FMT_SMR.1 FMT_SMF.1
O.DIRECTIVE_ANNEX_III_PART_2	FTP_TRP.1
Mapping for Objectives derived from the Annex IV of the [EUROPEAN_DIRECTIVE]	
O.DIRECTIVE_ANNEX_IV_PART_a	FTP_TRP.1
O.DIRECTIVE_ANNEX_IV_PART_b	FPT_TEE.1 FTP_TRP.1 FDP_DAU_SIG_(EXT).1 FDP_ITC_SIG_(EXT).1
O.DIRECTIVE_ANNEX_IV_PART_c	FTP_TRP.1
O.DIRECTIVE_ANNEX_IV_PART_d	FDP_DAU_CPV_(EXT).1 FDP_DAU_CPI_(EXT).1 FDP_DAU_CPO_(EXT).1 FDP_CPD_(EXT).1 FDP_DAU_CPV_(EXT).2 FPT_TEE.1

O.DIRECTIVE_ANNEX_IV_PART_e	FTP_TRP.1
O.DIRECTIVE_ANNEX_IV_PART_f	FDP_ITC_SIG_(EXT).1
O.DIRECTIVE_ANNEX_IV_PART_g	FAU_GEN.1-NIAP-0407:2

Table 5-34. Security Objective to Functional Component Mapping

Certification Path Validation – Basic Package Rationale

O.Availability states that the TSF shall continue to provide security services even if revocation information is not available. This objective is met by:

- FDP_DAU_CPV_(EXT).1, Certificate processing – basic, which requires that the TSF bypass the revocation check if the revocation information is not available.

O.Correct_Temporal states that the TSF shall provide accurate temporal validation results. This objective is met by:

- FDP_DAU_CPI_(EXT).1, Certification path initialisation – basic, which requires that the TSF obtain the time of interest called "TOI" from a reliable source.

O.Current_Certificate states that the TSF shall only accept certificates that are not expired as of TOI. This objective is met by:

- FDP_DAU_CPV_(EXT).1, which requires that the TSF accept a certificate only if the specified checks succeed, including that the certificate is not expired as of TOI.

O.Get_KeyInfo states that the TSF shall provide the user public key and related information in order to carry out cryptographic functions. This objective is met by:

- FDP_DAU_CPO_(EXT).1, Certification path output – basic, which requires that the TSF output the subject public key from the certification path and other information specified by the ST author.

O.Path_Find states that the TSF shall be able to find a certification path from a trust anchor to the subscriber. This objective is met by:

- FDP_CPD_(EXT).1, Certification path development, which requires that the TSF shall develop a certification path from a trust anchor to the subscriber.

O.Trusted_Keys states that the TSF shall use trusted public keys in certification path validation. This objective is met by:

- FDP_DAU_CPI_(EXT).1, Certification path initialisation -- basic, which requires that the TSF use trusted public keys in the certification path validation.

O.User states that the TSF shall only accept certificates issued by a CA. This objective is met by:

- FDP_DAU_CPV_(EXT).2, Intermediate certificate processing – basic, which requires that the TSF accept an intermediate certificate only when the certificate is issued by a CA.

O.Verified_Certificate states that the TSF shall only accept certificates with verifiable signatures. This objective is met by:

- FDP_DAU_CPV_(EXT).1, Certificate processing – basic, which requires that the TSF accept certificates only with verifiable signatures.

O.Valid_Certificate states that the TSF shall use certificates that are valid, i.e., not revoked. This objective is met by:

- FDP_DAU_CPV_(EXT).1, Certificate processing – basic, which requires that that the TSF shall use only those certificates that are valid, i.e., revocation status demonstrates that the certificate is not revoked.

Certification Path Validation – Basic Policy Package Rationale

O.Provide_Policy_Info states that the TSF shall provide certificate policies for which the certification path is valid. This objective is met by:

- FDP_DAU_CPI_(EXT).2, Certification path initialisation – basic policy, which requires that the TSF shall use the initial-certificate-policies provided by user roles specified by the ST author.
- FDP_DAU_CPO_(EXT).2, Certification path output – basic policy, which requires that The TSF shall output the certificate policies using the following rule: intersection of certificatePolicies extensions in all the certificates in certification path and initial-certificate-policies.

PKI Signature Generation Package Rationale

O.Give_Sig_Hints states that the TSF shall provide hints for selecting correct certificates for PKI signature verification. This objective is met by:

- FDP_ETC_SIG_(EXT).1 Export of PKI Signature, which requires that the TSF use the user selected private to key perform digital signature and that the TSF include additional information with the digital signature to facilitate signature verification.

PKI Signature Verification Package Rationale

O.Use_Sig_Hints states that the TSF shall use hints for selecting correct certificates for signature verification. This objective is met by:

- FDP_ITC_SIG_(EXT).1, Import of PKI Signature, which requires that the TSF use the following information from the signed data: hashing algorithm, signature algorithm, signer public key certificate, signer DN, signer subject alternative name, signer subject key identifier, or other data during signature verification.

O.Linkage_Sig_Ver states that the TSF shall use the correct user public key for signature verification. This objective is met by:

- FDP_DAU_SIG_(EXT).1, Signature Blob Verification, which requires that the TSF invoke a cryptographic module with the following information from Certification Path Validation to verify digital signature on signed data: subject public key algorithm, subject public key, subject public key parameters.

PKI Encryption using Key Transfer Algorithms Package Rationale

O.Hints_Enc_WO states that the TSF shall provide hints for selecting correct certificates or keys for PKI Encryption using Key Transfer algorithms. This objective is met by:

- FDP_ETC_ENC_(EXT).1, Export of PKI Encryption – Key Transfer Algorithms, which requires that the TSF include the information with the encrypted data, such as the public key, as selected or assigned by the ST author and that the TSF invoke a cryptographic module with the following information from Certification Path Validation to create encrypted data: subject public key algorithm, subject public key, subject public key parameters.

O.Linkage_Enc_WO states that the TSF shall use the correct user public key for key transfer.

- FDP_ETC_ENC_(EXT).1, Export of PKI Encryption – Key Transfer Algorithms, which requires that the TSF invoke a cryptographic module with the following information from Certification Path Validation to create encrypted data: subject public key algorithm, subject public key, subject public key parameters.
- FDP_DAU_ENC_(EXT).1, PKI Encryption Verification – Key Transfer, which requires that the TSF apply verification checks for key transfer as selected by the ST author.

PKI Decryption using Key Transfer Algorithms Package Rationale

O.Correct_KT states that the TSF shall use appropriate private key and key transfer algorithm:

- FDP_ITC_ENC_(EXT).1, Import of PKI Encryption – Key Transfer Algorithms, which requires that the TSF invoke a cryptographic module with the information from the encrypted data as selected by the ST author to provide a means to identify an appropriate private key and key transfer algorithm.

PKI Based Entity Authentication Package Rationale

The PKI Based Entity Authentication package may or may not be included in an ST, depending on the functionality of the application.

O.I&A states that the TSF shall uniquely identify all entities, and shall authenticate the claimed identity before granting an entity access to the TOE facilities. This objective is met by:

- FIA_UAU.1;1, Timing of authentication, which requires that the TSF allow the a list of TSF mediated actions, specified by the ST author, to be performed on behalf of the user before the user is authenticated and that TSF shall require each user to

be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user. This requirement ensures that all users are authenticated.

- FIA_UID.1;1, Timing of identification, which requires that the TSF allow the a list of TSF mediated actions, specified by the ST author, to be performed on behalf of the user before the user is identified and that TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user. This requirement ensures that all users are identified.

O.Limit_Actions_Auth states that the TSF shall restrict the actions an entity may perform before the TSF verifies the identity of the entity. This objective is met by:

- FIA_UAU.1;1, Timing of authentication, which requires that the TSF allow the a list of TSF mediated actions, specified by the ST author, to be performed on behalf of the user before the user is authenticated and that TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user. This requirement ensures that all users are authenticated.

- FIA_UID.1;1, Timing of identification, which requires that the TSF allow the a list of TSF mediated actions, specified by the ST author, to be performed on behalf of the user before the user is identified and that TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user. This requirement ensures that all users are identified.

O.Linkage states that the TSF shall use the correct user public key for authentication. This objective is met by:

- FIA_UAU_SIG_(EXT).1, Entity authentication, which requires that the TSF invoke a cryptographic module with the following information from Certification Path Validation to verify the signature on signed data: subject public key algorithm, subject public key, subject public key parameters, and that the TSF perform additional checks as specified by the ST author.

O.Single_Use_I&A states that the TSF shall use the I&A mechanism that requires unique authentication information for each I&A. This objective is met by:

- FIA_UAU.4, Single-use authentication mechanisms, which requires that the TSF prevent reuse of authentication data.

Online Certificate Status Protocol Package Rationale

O.Accurate_OCSP_Info states that the TSF shall accept only accurate OCSP responses. This objective is met by:

- FDP_DAU_OCS_(EXT).1, Basic OCSP Client, which requires that only accurate revocation information be accepted from the OCSP responder.

O.Auth_OCSP_Info states that the TSF shall accept the OCSP responses from an authorized source. This objective is met by:

- FDP_DAU_OCS_(EXT).1, Basic OCSP Client, which requires that the OCSP responder be verified as an authorized source.

O.Current_OCSP_Info states that the TSF may accept only OCSP responses current as of TOI. This objective is met by:

- FDP_DAU_OCS_(EXT).1, Basic OCSP Client, which requires that only reasonably current as of TOI revocation information may be accepted through a series of policy and parameter checks.

O.User_Override_Time_OCSP states that the TSF shall permit the user to override the time checks on the OCSP response. This objective is met by:

- FDP_DAU_OCS_(EXT).1, Basic OCSP Client, which requires that a role or roles specified by the ST author be able to override the time checks on the OCSP response.

Certificate Revocation List (CRL) Validation Package Rationale

O.Accurate_Rev_Info states that the TSF shall accept only accurate revocation information. This objective is met by:

- FDP_DAU_CRL_(EXT).1, Basic CRL checking, which requires that the TSF accept accurate revocation information. Accuracy is determined through a series of verification and policy requirements within this extended stated requirement.

O.Auth_Rev_Info states that the TSF shall accept the revocation information from an authorized source for CRL. This objective is met by:

- FDP_DAU_CRL_(EXT).1, Basic CRL checking, which requires that the TSF accept revocation information from an authorized source as selected or assigned by the ST author.

O.Current_Rev_Info states that the TSF shall accept only CRL current as of TOI. This objective is met by:

- FDP_DAU_CRL_(EXT).1, Basic CRL checking, which requires that the TSF accept only reasonably current as of TOI revocation information through a series of policy requirements defined in FDP_DAU_CRL_(EXT).1.

O.User_Override_Time_CRL states that the TSF shall permit the user to override the time checks on the CRL. This objective is met by:

- FDP_DAU_CRL_(EXT).1, Basic CRL checking, which requires that the TSF accept the CRL as current if a role assigned by the ST author overrides time checks.

Audit Package Rationale

O.PKE_Audit states that the TSF shall audit security relevant PKE events. This objective is met by:

- FAU_GEN.1-NIAP-0407 defines the set of events that the TOE must be capable of recording. This requirement ensures that the Administrator has the ability to audit events that take place in the TOE. This requirement also defines the information that must be contained in the audit record for each auditable event. This

requirement also places a requirement on the level of detail that is recorded on any additional security functional requirements an ST author adds to this PP.

- FAU_GEN.2-NIAP-0410 ensures that the audit records associate a user identity with the auditable event.

Continuous Authentication Package Rationale

O.Continuous_I&A states that the TSF shall continuously authenticate the entity. This objective is met by:

- FIA_UAU.6, Re-authenticating entity, which requires that the TSF re-authenticate an entity under the conditions specified by the ST author.

Rationale for the security objectives derived from Annex III of the [EUROPEAN_DIRECTIVE]

O.DIRECTIVE_ANNEX_III_PART_1a states that the TOE guarantees the operation of its services in an execution environment with a FIPS 140-2 level 3 security level for the following processes: assuring the secure custody of the private key and the uniqueness property for the private key (the signature-creation data used for signature generation can, practically, occur only once). This objective is satisfied by the following requirements:

- **FPT_TEE.1** provides a mechanism with which the TOE forces a high level of security in the use of the cryptographic devices. The TOE detects if the cryptographic module is FIPS 140-2 level 3 configured. If it is, this guarantees strict security in the custody of the private keys and in the cryptographic processes that assure the uniqueness property of the private key.

O.DIRECTIVE_ANNEX_III_PART_1b states that the TOE guarantees the operation of its services in an execution environment with a FIPS 140-2 level 3 security level for the following processes: assuring that the private key cannot be derived using currently-available technology and that the signatures cannot been falsified. This objective is satisfied by the following requirements:

- **FPT_TEE.1** provides a mechanism with which the TOE forces a high level of security in the use of the cryptographic devices. The TrustedX TOE detects if the cryptographic module is FIPS 140-2 level 3 configured. If it is, this guarantees strict security in the custody of the private keys and in the cryptographic processes of creating private keys and generating digital signatures.

O.DIRECTIVE_ANNEX_III_PART_1c states that the TOE must ensure that the signature-creation data used for signature generation can be reliably protected by the legitimate signer against being used by others. This objective is satisfied by the following requirements:

- **FPT_TEE.1** provides a mechanism with which the TOE forces a high level of security in the use of the cryptographic devices. The TrustedX TOE detects if the cryptographic module is FIPS 140-2 level 3 configured. If it is, this guarantees controlled access to the private key from the cryptographic device environment.

- The **FDP_ACC.1** and **FDP_ACF.1** requirements force the use of access control policies and rules that implement these policies, which contributes to the protection of private keys. The FDP_ACC.1 requirements define that an access control policy is enforced on a list of subjects acting on the behalf of users attempting to gain access to a list of named objects. The operations between subject and object covered are those in which resources are involved. These requirements guarantee that only authorized users are granted access to the operations where a resource is involved. The FDP_ACF.1 requirements define rules that can implement the access control policy mentioned above.

- The **FIA_UID.1** and **FIA_UAU.1** requirements force the use of authentication and identification mechanisms to guarantee controlled access to the private keys. The FIA_UID.1 requires that a user be identified to the TOE to be able to do anything. The FIA_UAU.1 requires that a user be authenticated before the TOE to be able to do anything.

The **FMT_MSA.1**, **FMT_MSA.3**, **FMT_SMR.1** and **FMT_SMF.1** requirements force the control access policy and therefore contribute to the protection of the private keys. The FMT_MSA.1 requirement allows authorized users (roles) to manage the specified security attributes. The FMT_MSA.3 requirement ensures that the default values of security attributes are appropriately permissive or restrictive in nature. The FMT_SMR.1 requirement supports specifying the roles with respect to security that the TSF recognizes. The FMT_SMF.1 requirement requires the provision by the TSF of specific management functions.

O.DIRECTIVE_ANNEX_III_PART_2 states that the TOE must provide integrity to the data to be signed. This objective is satisfied by the following requirement:

- **FTP_TRP.1** Trusted path requires that a trusted path between the TSF and a user be provided for a set of events defined by the ST. The TOE provides technical mechanisms to assure this integrity in the path between the user requesting the signature service and the TOE. In the TOE's context, the integrity is assured by via the OE.PHYSICAL environment objective that assumes that the environment provides the TOE with appropriate physical security.

Rationale for the security objectives derived from Annex IV of the [EUROPEAN_DIRECTIVE]

O.DIRECTIVE_ANNEX_IV_PART_a states that during the signature-verification process, the TOE must ensure with reasonable certainty that the data used for verifying the signature corresponds to the data displayed to the verifier. This objective is satisfied by the following requirements:

- **FTP_TRP.1** Trusted path requires that a trusted path between the TSF and a user be provided for a set of events defined by the ST. The TOE provides technical mechanisms to assure the integrity service in the path between the TOE and the user requesting the signature verification service. In the TOE's context, the integrity is assured via the OE.PHYSICAL environment objective that assumes that the environment provides the TOE with appropriate physical security.

O.DIRECTIVE_ANNEX_IV_PART_b states that during the signature-verification process, the TOE must ensure with reasonable certainty that the signature is reliably verified

and the result of that validation is correctly displayed. This objective is satisfied by the following requirements:

- **FPT_TEE.1** provides a mechanism with which the TOE forces a high level of security in the use of the cryptographic devices. The TrustedX TOE detects if the cryptographic module is FIPS 140-2 level 3 configured. Where it is, this guarantees strict security in the cryptographic processes for signature validation.

- **FDP_DAU_SIG_(EXT).1**, Signature blob verification, which requires that the TSF invokes a cryptographic module with the following information on the certification path validation to a verify digital signature on signed data: subject public key algorithm, subject public key, subject public key parameters. **FDP_ITC_SIG_(EXT).1**, Import of PKI Signature, which requires that the TSF use the following information on the signed data: hashing algorithm, signature algorithm, signer public key certificate, signer DN, signer subject alternative name, signer subject key identifier, or other data during signature verification. These requirements force the verification of all the information on the certificate (building of certification paths, processing of the X.509 extensions, and all aspects related to the [X.509] standard), and invoking cryptographic hardware for the cryptographic verification of the signatures related to the validation of the certificate.

- **FTP_TRP.1** Trusted path requires that a trusted path between the TSF and a user be provided for a set of events defined by the ST. The TOE provides technical mechanisms to assure the integrity service in the path between the TOE and the user requesting the signature verification service. In the TOE's context, the integrity is assured via the OE.PHYSICAL environment objective that assumes that the environment provides the TOE with appropriate physical security.

O.DIRECTIVE_ANNEX_IV_PART_c states that during the signature-verification process, the TOE must ensure with reasonable certainty that the verifier can, as necessary, reliably establish the contents of the signed data. This objective is satisfied by the following requirements:

- **FTP_TRP.1** Trusted path requires that a trusted path between the TSF and a user be provided for a set of events defined by the ST. The TOE provides technical mechanisms to assure the integrity service in the path between the TOE and the user requesting the signature verification service. In the TOE's context, the integrity is assured via the OE.PHYSICAL environment objective that assumes that the environment provides the TOE with appropriate physical security.

O.DIRECTIVE_ANNEX_IV_PART_d states that during the signature-verification process, the TrustedX TOE must ensure with reasonable certainty that the authenticity and validity of the certificate required at the time of signature verification are reliably verified. This objective is satisfied by the following requirements:

- **FDP_DAU_CPV_(EXT).1**, Certificate processing (basic), which requires that the TSF use only valid certificates, i.e., the revocation status demonstrates that the certificate is not revoked. **FDP_DAU_CPI_(EXT).2**, Certification path initialization (basic policy), which requires that the TSF use the initial-certificate policies provided by user roles. **FDP_DAU_CPO_(EXT).2**, Certification path output (basic policy), which requires that the TSF return certificate policies using the following rule: the intersection of certificatePolicies extensions in all the certificates in certification path and initial-certificate-policies. **FDP_CPD_(EXT).1**, Certification path development, which requires that the TSF develop a certification path from

a trust anchor to the subscriber. **FDP_DAU_CPV_(EXT).2**, Intermediate certificate processing (basic), which requires that the TSF accept an intermediate certificate only when the certificate is issued by a CA.

These requirements force the verification of all the information on the certificate (building of certification paths, processing of the X.509 extensions and all aspects related to the [X.509] standard) and the invoking of cryptographic hardware for the cryptographic verification of the signatures related to the validation of the certificate.

- **FPT_TEE.1** provides a mechanism with which the TOE forces a high level of security in the use of the cryptographic devices. The TrustedX TOE detects if the cryptographic module is FIPS 140-2 level 3 configured. Where it is, this guarantees that signature verification is done in line with FIPS 140-2 level 3 security conditions.

O.DIRECTIVE_ANNEX_IV_PART_e states that during the signature-verification process, the TOE must ensure with reasonable certainty that the signatory's identity is correctly displayed. This objective is satisfied by the following requirements:

- **FTP_TRP.1** Trusted path requires that a trusted path between the TSF and a user be provided for a set of events defined by the ST. The TOE provides technical mechanisms to assure the integrity service in the path between the TOE and the user requesting the signature verification service. In the TOE's context, the integrity is assured via the OE.PHYSICAL environment objective that assumes that the environment provides the TOE with appropriate physical security.

O.DIRECTIVE_ANNEX_IV_PART_f states that during the signature-verification process, the TOE must ensure with reasonable certainty that the use of pseudonyms is supported. This objective is satisfied by the following requirement:

- **FDP_ITC_SIG_(EXT).1**, Import of PKI Signature, which requires that the TSF use the following information from the signed data: hashing algorithm, signature algorithm, signer public key certificate, signer DN, signer subject alternative name, signer subject key identifier, or other data during signature verification. The TOE assures this objective by supporting the use of X.509 public key certificates in the signatures verification process (the X.509 certificates support extensions and attributes in which it is possible to indicate all type of information, including pseudonyms).

O.DIRECTIVE_ANNEX_IV_PART_g states that during the signature-verification process, the TOE must ensure with reasonable certainty that any security-relevant changes can be detected. This objective is satisfied by the following requirement:

- **FAU_GEN.1-NIAP-0407:2** defines the set of events that the TOE must be capable of logging. This requirement means that the administrator can audit events that take place in the TOE. This requirement also defines the information that must be contained in the audit record for each auditable event. This requirement also places a requirement on the level of detail that is recorded. The TOE accomplishes this objective by implementing mechanisms for logging and detecting security events.

5.4.2.2. Assurance Requirements Rationale

EAL4 provides assurance by an analysis of security functions, using a functional and complete interface specification, guidance documentation, the high-level and low-level design of the TOE, and a subset of the implementation, to understand the security behavior. Assurance is additionally gained through an informal model of the TOE security policy. EAL4 is augmented with ALC_FLR.2 to track and correct the reported and found security flaws in the product and also to provide flaw reporting procedures to the product users.

5.4.3. Dependency Rationale

Requirement	Dependencies
CPV – Basic Package	
FDP_CPD_(EXT).1	None
FDP_DAU_CPI_(EXT).1	Note 2
FDP_DAU_CPV_(EXT).1	[FDP_DAU_OCS_(EXT).1 or FDP_DAU_CRL_(EXT).1] Note 3
FDP_DAU_CPV_(EXT).2	FDP_DAU_CPV_(EXT).1
FDP_DAU_CPO_(EXT).1	FDP_DAU_CPV_(EXT).1
Basic Policy Package	
FDP_DAU_CPI_(EXT).2	FDP_DAU_CPI_(EXT).1 (See Note 1)
FDP_DAU_CPO_(EXT).2	FDP_DAU_CPO_(EXT).1 (See Note 1)
PKI Signature Generation Package	
FDP_ETC_SIG_(EXT).1	Note 2
PKI Signature Verification Package	
FDP_ITC_SIG_(EXT).1	None
FDP_DAU_SIG_(EXT).1	FDP_DAU_CPO_(EXT).1 (See Note 1) Note 3
PKI Encryption using Key Transfer Algorithms Package	
FDP_ETC_ENC_(EXT).1	FDP_DAU_CPO_(EXT).1 (See Note 1) Note 3
FDP_DAU_ENC_(EXT).1	FDP_DAU_CPO_(EXT).1 (See Note 1)

PKI Decryption using Key Transfer Algorithms Package	
FDP_ITC_ENC_(EXT).1	Note 2
PKI Based Entity Authentication Package	
FIA_UAU.1	FIA_UID.1
FIA_UAU.4	None
FIA_UAU_SIG_(EXT).1	FDP_DAU_CPO_(EXT).1 (see Note 1) Note 3
FIA_UID.1	None
Online Certificate Status Protocol Client Package	
FDP_DAU_OCS_(EXT).1	Note 2
Certificate Revocation List (CRL) Validation Package	
FDP_DAU_CRL_(EXT).1	Note 2
Audit Package	
FAU_GEN.1-NIAP-0407:2	Note 2
FAU_GEN.2-NIAP-0410:2	Note 2
Continuous Authentication Package	
FIA_UAU.6	None
Annex III of the [EUROPEAN_DIRECTIVE]	
TOE Requirements	
FDP_ACC.1	FDP_ACF.1
FDP_ACF.1	FDP_ACC.1 FMT_MSA.3
FIA_UAU.1	FIA_UID.1
FIA_UID.1	None
FMT_MSA.3	FMT_MSA.1 FMT_SMR.1
FMT_SMR.1	FIA_UID.1
FMT_MSA.1	FDP_ACC.1 FMT_SMR.1

	FMT_SMF.1
FMT_SMF.1	None
FMT_SMF.1	FDP_ACC.1
FMT_SMR.1	FDP_ACC.1
FTP_TRP.1	None
FPT_TEE.1	None
Annex IV of the [EUROPEAN_DIRECTIVE]	
FDP_DAU_SIG_(EXT).1	FDP_DAU_CPO_(EXT).1 (See Note 1) Note 3
FDP_ITC_SIG_(EXT).1	None
FDP_DAU_CPV_(EXT).1	[FDP_DAU_OCS_(EXT).1 FDP_DAU_CRL_(EXT).1] Note 3
FDP_DAU_CPI_(EXT).1	Note 2
FDP_DAU_CPO_(EXT).1	FDP_DAU_CPV_(EXT).1
FDP_CPD_(EXT).1	None
FDP_DAU_CPV_(EXT).2	FDP_DAU_CPV_(EXT).1
FAU_GEN.1-NIAP-0407:2	Note 2
FPT_TEE.1	None

Table 5-35. Functional Requirements Dependencies

Note 1: The dependency is satisfied by including the CPV – Basic Package

Note 2: The dependencies related to this requirement are satisfied by the environment (see Clarifications regarding Note 2 and Note 3 section).

Note 3: And other dependencies that are satisfied by the environment (see Clarifications regarding Note 2 and Note 3 section).

Clarifications regarding Note 2 and Note 3

The **FDP_DAU_CPI_(EXT).1** requirements depend on environment requirements of cryptographic operations (requiring a cryptographic operation to be performed in accordance with a specified algorithm and with a cryptographic key of specified sizes), and on the environment requirements of reliable time stamps (requiring that the TSF provide reliable time stamps for TSF functions).

The **FDP_DAU_CPV_(EXT).1** requirements depend on environment requirements of cryptographic operations (requiring a cryptographic operation to be performed in accordance with a specified algorithm and with a cryptographic key of specified sizes), and on the environment requirements of reliable time stamps (requiring that the TSF provide reliable time stamps for TSF functions).

The **FDP_ETC_SIG_(EXT).1** requirements depend on environment requirements requiring FIPS compliant cryptographic modules.

The **FDP_DAU_SIG_(EXT).1** requirements depend on environment requirements requiring FIPS compliant cryptographic modules.

The **FDP_ETC_ENC_(EXT).1** requirements depend on environment requirements requiring FIPS compliant cryptographic modules.

The **FDP_ITC_ENC_(EXT).1** requirements depend on environment requirements requiring FIPS compliant cryptographic modules.

The **FIA_UAU_SIG_(EXT).1** requirements depend on environment requirements requiring FIPS compliant cryptographic modules.

The **FDP_DAU_OCS_(EXT).1** requirements depend on environment requirements requiring FIPS compliant cryptographic modules, and on environment requirements of reliable time stamps (requiring that the TSF provide reliable time stamps for TSF functions).

The **FDP_DAU_CRL_(EXT).1** requirements depend on environment requirements requiring FIPS compliant cryptographic modules, and on environment requirements of reliable time stamps (requiring that the TSF provide reliable time stamps for TSF functions).

The **FAU_GEN.1-NIAP-0407:2** requirements depend on environment requirements of reliable time stamps (requiring that the TSF provide reliable time stamps for TSF functions).

The **FAU_GEN.2-NIAP-0410:2** requirements depend on environment requirements requiring timing of identification (allowing users to perform certain actions before being identified by the TSF), and on environment requirements of audit data generation (that define the level of auditable events, and specify the list of data that shall be recorded in each record).

CHAPTER 6

6. Protection Profile Conformance Claim Rationale

This section justifies the conformity of this Security Target with the [PKE_PP] PP claimed. The following sections provide the evidence of the conformance with the Protection Profile.

6.1. Conformance with the TOE type

The TOE described in this Security Target is conformance with the Protection Profile TOE type included in the chapter 2 “TOE Description” of the [PKE_PP] document.

The TOE type described in [PKE_PP] PP consists of a public key application that is enabled of (page 19 of the [PKE_PP] document):

1 Securely manage private keys and trust anchors.

2 Manage public key certificates.

3 Use one or more of the security services supported by a PKI by accepting and processing an X.509 public key certificate.

4 Obtain relevant certificates and revocation data.

5 Check each certificate for validity, using procedures described in the X.509 standard, prior to reliance, including checking for revocation.

6 Have access to accurate and reliable time in order to verify the dates on certificates, revocation data, and application data.

7 Correctly interoperate with an appropriate cryptographic token.

8 Collect, store and maintain the data required to support digital signature verification in the future.

9 Be able to automatically select from multiple private decryption keys if it performs public key based decryption.

This Security Target defines the TOE type of the following way (section 1.2 TOE Overview: “The TrustedX TOE is a Web services platform which, by providing authentication, authorization, electronic signature and data protection, resolves the security and trust problems that arise when business processes exchange documents

and information.". The points mentioned above are included as security functionality inside the TrustedX web services platform. All this functionality is described in section 1.3.2 TrustedX Service Components of this Security Target.

6.2. Conformance with the PP requirements

6.2.1. Conformance with the PP functional requirements for the TOE

This Security Target is conformance with the Protection Profile functional requirements for the TOE.

The following table provides the evidence of this conformance for the packages[8] included in this ST.

PP Requirement Name	ST Requirement Name
FDP_CPD_(EXT).1	FDP_CPD_(EXT).1
FDP_DAU_CPI_(EXT).1	FDP_DAU_CPI_(EXT).1
FDP_DAU_CPV_(EXT).1	FDP_DAU_CPV_(EXT).1
FDP_DAU_CPV_(EXT).2	FDP_DAU_CPV_(EXT).2
FDP_DAU_CPO_(EXT).1	FDP_DAU_CPO_(EXT).1
FDP_DAU_CPI_(EXT).2	FDP_DAU_CPI_(EXT).2
FDP_DAU_CPO_(EXT).2	FDP_DAU_CPO_(EXT).2
FDP_ETC_SIG_(EXT).1	FDP_ETC_SIG_(EXT).1
FDP_ITC_SIG_(EXT).1	FDP_ITC_SIG_(EXT).1
FDP_DAU_SIG_(EXT).1	FDP_DAU_SIG_(EXT).1
FDP_ETC_ENC_(EXT).1	FDP_ETC_ENC_(EXT).1
FDP_DAU_ENC_(EXT).1	FDP_DAU_ENC_(EXT).1
FDP_ITC_ENC_(EXT).1	FDP_ITC_ENC_(EXT).1
FIA_UAU.1	FIA_UAU.1

[8] *Basic Package, Basic Policy Package, PKI Signature Generation Package, PKI Signature Verification Package, PKI Encryption using Key Transfer Algorithms Package, PKI Decryption using Key Transfer Algorithms Package, PKI Based Entity Authentication Package, Online Certificate Status Protocol Client Package, Certificate Revocation List Validation Package, Audit Package and Continuous Authentication Package.*

FIA_UAU.4	FIA_UAU.4
FIA_UAU_SIG_(EXT).1	FIA_UAU_SIG_(EXT).1
FIA_UID.1	FIA_UID.1
FDP_DAU_OCS_(EXT).1	FDP_DAU_OCS_(EXT).1
FDP_DAU_CRL_(EXT).1	FDP_DAU_CRL_(EXT).1
FAU_GEN.1-NIAP-0407:2	FAU_GEN.1-NIAP-0407:2
FAU_GEN.2-NIAP-0410:2	FAU_GEN.2-NIAP-0410:2
FIA_UAU.6	FIA_UAU.6

Table 6-36. Conformance with the PP functional requirements for the TOE

Additionally the functional requirements included in Table 5-3. Security Functional Requirements for the TOE derived from the [EUROPEAN_DIRECTIVE] have been included in this Security Target in order to support new security objectives that complement the [PKE_PP] PP.

6.2.2. Conformance with the PP assurance requirements

This Security Target is conformance with the Protection Profile assurance requirements.

The following table provides the evidence of this conformance.

PP Component Name	ST Component Name
ASE_CCL.1	ASE_CCL.1
ASE_ECD.1	ASE_ECD.1
ASE_INT.1	ASE_INT.1
ASE_OBJ.2	ASE_OBJ.2
ASE_REQ.2	ASE_REQ.2
ASE_SPD.1	ASE_SPD.1
ASE_TSS.1	ASE_TSS.1
ADV_ARC.1	ADV_ARC.1
ADV_FSP.4	ADV_FSP.4
ADV_IMP.1	ADV_IMP.1
ADV_TDS.3	ADV_TDS.3

AGD_OPE.1	AGD_OPE.1
AGD_PRE.1	AGD_PRE.1
ALC_CMC.4	ALC_CMC.4
ALC_CMS.4	ALC_CMS.4
ALC_DEL.1	ALC_DEL.1
ALC_DVS.1	ALC_DVS.1
ALC_FLR.2	ALC_FLR.2
ALC_LCD.1	ALC_LCD.1
ALC_TAT.1	ALC_TAT.1
ATE_COV.2	ATE_COV.2
ATE_DPT.2	ATE_DPT.2
ATE_FUN.1	ATE_FUN.1
ATE_IND.2	ATE_IND.2
AVA_VAN.3	AVA_VAN.3

Table 6-37. Conformance with the PP assurance requirements

6.3. Conformance with the PP assumptions

This Security Target is conformance with the Protection Profile security assumptions for the IT environment.

The following table provides the evidence of this conformance.

PP Assumption Name	ST Assumption Name	Description
A.Configuration	A.Configuration	The TOE will be properly installed and configured.
A.Basic	A.Enhanced-Basic	The attack potential on the TOE is assumed to be ”Enhanced-Basic”. The TOE is more resistant that the level of attack required by the PP.
A.NO_EVIL	A.NO_EVIL	Administrators are non-hostile, appropriately

		trained and follow all administrator guidance.
A.PHYSICAL	A.PHYSICAL	It is assumed that the environment provides the TOE with appropriate physical security, commensurate with the value of the IT assets protected by the TOE.

Table 6-38. Conformance with the PP assumptions

6.4. Conformance with the PP organizational security policies

This Security Target is conformance with the Protection Profile organizational security policies.

The following table provides the evidence of this conformance.

PP Policy Name	ST Policy Name	Description
P.ACCESS_BANNER	P.ACCESS_BANNER	The IT Environment shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the system.
P.ACCOUNTABILITY	P.ACCOUNTABILITY	The authorized users of the TOE shall be held accountable for their actions within the TOE.
P.CRYPTOGRAPHY	P.CRYPTOGRAPHY	Only NIST FIPS validated cryptography (methods and implementations) are acceptable for key management (i.e.; generation, access, distribution, destruction, handling, and storage of keys) and cryptographic services (i.e.; encryption, decryption, signature, hashing, key exchange, and random number

		generation services).

Table 6-39. Conformance with the PP policies

6.5. Conformance with the PP threats

This Security Target is conformance with the Protection Profile threats for the TOE and environment.

The following table provides the evidence of this conformance for the **Base threats**.

PP Threat Name	ST Threat Name	Threat Description
T.AUDIT_COMPROMISE	T.AUDIT_COMPROMISE	A user or process may view audit records, cause audit records to be lost or modified, or prevent future audit records from being recorded, thus masking a user's action.
T.CHANGE_TIME	T.CHANGE_TIME	An unauthorized user may change the TSF notion of time resulting in accepting old revocation information or expired certificates.
T.CRYPTO_COMPROMISE	T.CRYPTO_COMPROMISE	A user or process may cause key, data or executable code associated with the cryptographic functionality to be inappropriately accessed (viewed, modified, or deleted), thus compromising the cryptographic mechanisms and the data protected by those mechanisms.
T.MASQUERADE	T.MASQUERADE	A user or process may masquerade as another entity in order to gain unauthorized access to data or TOE resources.

T.POOR_TEST	T.POOR_TEST	Lack of or insufficient tests to demonstrate that all TOE security functions operate correctly (including in a fielded TOE) may result in incorrect TOE behavior being undiscovered thereby causing potential security vulnerabilities.
T.RESIDUAL_DATA	T.RESIDUAL_DATA	A user or process may gain unauthorized access to data through reallocation of TOE resources from one user or process to another.
T.TSF_COMPROMISE	T.TSF_COMPROMISE	A user or process may cause, through an unsophisticated attack, TSF data, security attributes, or executable code to be inappropriately accessed (viewed, modified, or deleted).
T.UNATTENDED_SESSION	T.UNATTENDED_SESSION	A user may gain unauthorized access to an unattended session.
T.UNAUTHORIZED_ACCESS	T.UNAUTHORIZED_ACCESS	A user may gain access to user data for which they are not authorized according to the TOE security policy.
T.UNIDENTIFIED_ACTIONS	T.UNIDENTIFIED_ACTIONS	The administrator may not have the ability to notice potential security violations, thus limiting the administrator's ability to identify and take action against a possible security breach.

Table 6-40. Conformance with PP threats to TOE security for the Base threats

The following table provides the evidence of this conformance for the **Basic Package**.

PP Threat Name	ST Threat Name	Threat Description
T.Certificate_Modi	T.Certificate_Modi	An untrusted user may modify a certificate resulting in using a wrong public key.
T.DOS_CPV_Basic	T.DOS_CPV_Basic	The revocation information or access to revocation information could be made unavailable, resulting in loss of system availability.
T.Expired_Certificate	T.Expired_Certificate	An expired (and possibly revoked) certificate as of TOI could be used for signature verification.
T.Untrusted_CA	T.Untrusted_CA	An untrusted entity (Certification Authority (CA)) may issue certificates to bogus entities, permitting those entities to assume identity of other legitimate users.
T.No_Crypto	T.No_Crypto	The user public key and related information may not be available to carry out the cryptographic function.
T.Path_Not_Found	T.Path_Not_Found	A valid certification path is not found due to lack of system functionality.
T.Revoked_Certificate	T.Revoked_Certificate	A revoked certificate could be used as valid, resulting in security compromise.
T.User_CA	T.User_CA	A user could act as a CA, issuing unauthorized certificates.

Table 6-41. Conformance with PP threats to TOE security for the Basic Package

The following table provides the evidence of this conformance for the **Basic Policy Package**.

PP Threat Name	ST Threat Name	Threat Description
T.Unknown_Policies	T.Unknown_Policies	The user may not know the policies under which a certificate was issued.

Table 6-42. Conformance with PP threats to TOE security for the Basic Policy Package

The following table provides the evidence of this conformance for the **PKI Signature Generation Package**.

PP Threat Name	ST Threat Name	Threat Description
T.Clueless_PKI_Sig	T.Clueless_PKI_Sig	The user may try only inappropriate certificates for signature verification because the signature does not include a hint.

Table 6-43. Conformance with PP threats to TOE security for the PKI Signature Generation Package

The following table provides the evidence of this conformance for the **PKI Signature Verification Package**.

PP Threat Name	ST Threat Name	Threat Description
T.Assumed_Identity_PKI_Ver	T.Assumed_Identity_PKI_Ver	A user may assume the identity of another user in order to verify a PKI signature.
T.Clueless_PKI_Ver	T.Clueless_PKI_Ver	The user may try only inappropriate certificates for signature verification because hints in the signature are ignored.

Table 6-44. Conformance with PP threats to TOE security for the PKI Signature Verification Package

The following table provides the evidence of this conformance for the **PKI Encryption using Key Transfer Algorithms Package**.

PP Threat Name	ST Threat Name	Threat Description
T.Assumed_Identity_WO_En	T.Assumed_Identity_WO_En	A user may assume the identity of another user in order to perform encryption using Key Transfer algorithms.
T.Clueless_WO_En	T.Clueless_WO_En	The user may try only inappropriate certificates for encryption using Key Transfer algorithms in absence of hint.

Table 6-45. Conformance with PP threats to TOE security for the PKI Encryption using Key Transfer Algorithms Package

The following table provides the evidence of this conformance for the **PKI Decryption using Key Transfer Algorithms Package**.

PP Threat Name	ST Threat Name	Threat Description
T.Garble_WO_De	T.Garble_WO_De	The user may not apply the correct key transfer algorithm or private key, resulting in garbled data.

Table 6-46. Conformance with PP threats to TOE security for the PKI Decryption using Key Transfer Algorithms Package

The following table provides the evidence of this conformance for the **PKI Based Entity Authentication Package**.

PP Threat Name	ST Threat Name	Threat Description
T.Assumed_Identity_Auth	T.Assumed_Identity_Auth	A user may assume the identity of another user to perform entity based authentication.
T.Replay_Entity	T.Replay_Entity	An unauthorized user may replay valid entity authentication data.

Table 6-47. Conformance with PP threats to TOE security for the PKI Based Entity Authentication Package

The following table provides the evidence of this conformance for the **Online Certificate Status Protocol Client Package**.

PP Threat Name	ST Threat Name	Threat Description
T.DOS_OCSP	T.DOS_OCSP	The OCSP response or access to the OCSP response could be made unavailable, resulting in loss of system availability.
T.Replay_OCSP_Info	T.Replay_OCSP_Info	The user may accept an OCSP response from well before TOI resulting in accepting a revoked certificate.
T.Wrong_OCSP_Info	T.Wrong_OCSP_Info	The user may accept a revoked certificate or reject a valid certificate due to a wrong OCSP response.

Table 6-48. Conformance with PP threats to TOE security for the Online Certificate Status Protocol Client Package

The following table provides the evidence of this conformance for the **Certificate Revocation List (CRL) Package**.

PP Threat Name	ST Threat Name	Threat Description
T.DOS_CRL	T.DOS_CRL	The CRL or access to CRL could be made unavailable, resulting in loss of system availability.
T.Replay_Revoc_Info_CRL	T.Replay_Revoc_Info_CRL	The user may accept a CRL issued well before TOI resulting in accepting a revoked certificate.
T.Wrong_Revoc_Info_CRL	T.Wrong_Revoc_Info_CRL	The user may accept a revoked certificate or reject a valid certificate due to a wrong CRL.

Table 6-49. Conformance with PP threats to TOE security for the Certificate Revocation List (CRL) Package

The following table provides the evidence of this conformance for the **Audit Package**.

PP Threat Name	ST Threat Name	Threat Description
T.PKE_Accountability	T.PKE_Accountability	The PKE related audit events cannot be linked to individual actions.

Table 6-50. Conformance with PP threats to TOE security for the Audit Package

The following table provides the evidence of this conformance for the **Continuous Authentication Package**.

PP Threat Name	ST Threat Name	Threat Description
T.Hijack	T.Hijack	An unauthorized user may hijack an authenticated session.

Table 6-51. Conformance with PP threats to TOE security for the Continuous Authentication Package

Additionally the following threats have been defined: T.DISCLOSURE_ OR_NOT_UNIQUE_PRIVATE_KEYS, T.ILEGITIMATE_USE_OF_PRIVATE_KEYS, T.DATA_TO_BE_SIGNED, T.INVALID_CERTIFICATE, T.SIGNATURE_NOT_RELIABLY_VERIFIED, T.SIGNATURE_FALSIFIED, T.DATA_USED_FOR_VERIFYING_THE_SIGNATURE, T.RESULT_OF_THE_SIGNATURE_VALIDATION, T.SIGNATORY'S_IDENTITY, T.SIGNED_DATA_FALSIFIED, T.PSEUDONYM_NOT_SUPPORTED and T.SECURITY_AUDIT_EVENTS. This Security Target also defines news security objectives that the TOE achieves in order to mitigate these threats.

6.6. Conformance with the PP objectives

6.6.1. Conformance with PP Objectives for IT Environment

This Security Target is conformant with PP objectives for IT environment.

The following table provides the evidence of this conformance.

PP Objective Name	ST Objective Name	Objective Description
OE.AUDIT_GENERATION	OE.AUDIT_GENERATION	The IT Environment

		will provide the capability to detect and create records of security-relevant events associated with users.
OE.AUDIT_PROTECTION	OE.AUDIT_PROTECTION	The IT Environment will provide the capability to protect audit information.
OE.AUDIT_REVIEW	OE.AUDIT_REVIEW	The IT Environment will provide the capability to selectively view audit information.
OE.Configuration	OE.Configuration	The TOE will be installed and configured properly for starting up the TOE in a secure state.
OE.CORRECT_TSF_OPERATION	OE.CORRECT_TSF_OPERATION	The IT Environment will provide the capability to test the TSF to ensure the correct operation of the TSF at a customer's site.
OE.CRYPTOGRAPHY	OE.CRYPTOGRAPHY	The TOE shall use NIST FIPS 140-2 validated cryptographic services provided by the IT Environment.
OE.DISPLAY_BANNER	OE.DISPLAY_BANNER	The IT Environment will display an advisory warning regarding use of the TOE.
OE.Basic	OE.Enhanced-Basic	The TOE will be designed and implemented for a minimum attack potential of "Enhanced-Basic" as validated by the vulnerability analysis. The TOE is more

		resistant that the level of attack required by the PP.
OE.MANAGE	OE.MANAGE	The IT Environment will provide all the functions and facilities necessary to support the administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.
OE.MEDIATE	OE.MEDIATE	The IT Environment will provide all the functions and facilities necessary to support the administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.
OE.NO_EVIL	OE.NO_EVIL	Sites using the TOE will ensure that administrators are non-hostile, appropriately trained and follow all administrator guidance.
OE.PHYSICAL	OE.PHYSICAL	The non-IT environment will provide an acceptable level of physical security so that the TOE cannot be tampered with or be subject to side channel attacks such as the various forms of power analysis and timing analysis.

OE.RESIDUAL_INFORMATION	OE.RESIDUAL_INFORMATION	The IT Environment will ensure that any information contained in a protected resource within its Scope of Control is not released when the resource is reallocated.
OE.SELF_PROTECTION	OE.SELF_PROTECTION	The IT Environment will maintain a domain for its own execution that protects it and its resources from external interference, tampering, or unauthorized disclosure.
OE.TIME_STAMPS	OE.TIME_STAMPS	The IT Environment will provide reliable time stamps and the capability for the administrator to set the time used for these time stamps.
OE.TIME_TOE	OE.TIME_TOE	The IT Environment will provide reliable time for the TOE use.
OE.TOE_ACCESS	OE.TOE_ACCESS	The IT Environment will provide mechanisms that control a user's logical access to the TOE.
OE.TOE_PROTECTION	OE.TOE_PROTECTION	The IT Environment will protect the TOE and TOE resources from external interference, tampering, or unauthorized disclosure and modification.

Table 6-52. Conformance with PP objectives for IT Environment

6.6.2. Conformance with PP Objectives for TOE

This Security Target is conformant with PP objectives for TOE.

The following table provides the evidence of this conformance for the **Basic Package**.

PP Objective Name	ST Objective Name	Objective Description
O.Availability	O.Availability	The TSF shall continue to provide security services even if revocation information is not available.
O.Correct_Temporal	O.Correct_Temporal	The TSF shall provide accurate temporal validation results.
O.Current_Certificate	O.Current_Certificate	The TSF shall only accept certificates that are not expired as of TOI.
O.Get_KeyInfo	O.Get_KeyInfo	The TSF shall provide the user public key and related information in order to carry out cryptographic functions.
O.Path_Find	O.Path_Find	The TSF shall be able to find a certification path from a trust anchor to the subscriber.
O.Trusted_Keys	O.Trusted_Keys	The TSF shall use trusted public keys in certification path validation.
O.User	O.User	The TSF shall only accept certificates issued by a CA.
O.Verified_Certificate	O.Verified_Certificate	The TSF shall only accept certificates

		with verifiable signatures.
O.Valid_Certificate	O.Valid_Certificate	The TSF shall use certificates that are valid, i.e., not revoked.

Table 6-53. Conformance with PP objectives for the TOE (Basic Package)

The following table provides the evidence of this conformance for the **Basic Policy Package**.

PP Objective Name	ST Objective Name	Objective Description
O.Provide_Policy_Info	O.Provide_Policy_Info	The TSF shall provide certificate policies for which the certification path is valid.

Table 6-54. Conformance with PP objectives for the TOE (Basic Policy Package)

The following table provides the evidence of this conformance for the **PKI Signature Generation Package**.

PP Objective Name	ST Objective Name	Objective Description
O.Give_Sig_Hints	O.Give_Sig_Hints	The TSF shall provide hints for selecting correct certificates for signature verification.

Table 6-55. Conformance with PP objectives for the TOE (PKI Signature Generation Package)

The following table provides the evidence of this conformance for the **PKI Signature Verification Package**.

PP Objective Name	ST Objective Name	Objective Description
O.Use_Sig_Hints	O.Use_Sig_Hints	The TSF shall use hints for selecting correct certificates for signature verification.

O.Linkage_Sig_Ver	O.Linkage_Sig_Ver	The TSF shall use the correct user public key for signature verification.

Table 6-56. Conformance with PP objectives for the TOE (PKI Signature Verification Package)

The following table provides the evidence of this conformance for the **PKI Encryption using Key Transfer Algorithms Package**.

PP Objective Name	ST Objective Name	Objective Description
O.Hints_Enc_WO	O.Hints_Enc_WO	The TSF shall provide hints for selecting correct certificates or keys for PKI Encryption using Key Transfer Algorithms.
O.Linkage_Enc_WO	O.Linkage_Enc_WO	The TSF shall use the correct user public key for key transfer.

Table 6-57. Conformance with PP objectives for the TOE (PKI Encryption using Key Transfer Algorithms Package)

The following table provides the evidence of this conformance for the **PKI Decryption using Key Transfer Algorithms Package**.

PP Objective Name	ST Objective Name	Objective Description
O.Correct_KT	O.Correct_KT	The TSF shall use appropriate private key and key transfer algorithm.

Table 6-58. Conformance with PP objectives for the TOE (PKI Decryption using Key Transfer Algorithms Package)

The following table provides the evidence of this conformance for the **PKI Based Entity Authentication Package**.

PP Objective Name	ST Objective Name	Objective Description
O.I&A	O.I&A	The TSF shall uniquely identify all entities,

		and shall authenticate the claimed identify before granting an entity access to the TOE facilities.
O.Limit_Actions_Auth	O.Limit_Actions_Auth	The TSF shall restrict the actions an entity may perform before the TSF verifies the identity of the entity.
O.Linkage	O.Linkage	The TSF shall use the correct user public key for authentication.
O.Single_Use_I&A	O.Single_Use_I&A	The TSF shall use the I&A mechanism that requires unique authentication information for each I&A.

Table 6-59. Conformance with PP objectives for the TOE (PKI Based Entity Authentication Package)

The following table provides the evidence of this conformance for the **Online Certificate Status Protocol Client Package**.

PP Objective Name	ST Objective Name	Objective Description
O.Accurate_OCSP_Info	O.Accurate_OCSP_Info	The TSF shall accept only accurate OCSP responses.
O.Auth_OCSP_Info	O.Auth_OCSP_Info	The TSF shall accept the revocation information from an authorized source for OCSP transactions.
O.Current_OCSP_Info	O.Current_OCSP_Info	The TSF accept only OCSP responses current as of TOI.
O.User_Override_Time_OCSP	O.User_Override_Time_OCSP	The TSF shall permit the user to override the time checks on the OCSP response.

Table 6-60. Conformance with PP objectives for the TOE (Online Certificate Status Protocol Client Package)

The following table provides the evidence of this conformance for the **Certificate Revocation List (CRL) Validation Package**.

PP Objective Name	ST Objective Name	Objective Description
O.Accurate_Rev_Info	O.Accurate_Rev_Info	The TSF shall accept only accurate revocation information.
O.Auth_Rev_Info	O.Auth_Rev_Info	The TSF shall accept the revocation information from an authorized source for CRL.
O.Current_Rev_Info	O.Current_Rev_Info	The TSF shall accept only CRL that are current as of TOI.
O.User_Override_Time_CRL	O.User_Override_Time_CRL	The TSF shall permit the user to override the time checks on the CRL.

Table 6-61. Conformance with PP objectives for the TOE (Certificate Revocation List (CRL) Package)

The following table provides the evidence of this conformance for the **Audit Package**.

PP Objective Name	ST Objective Name	Objective Description
O.PKE_Audit	O.PKE_Audit	The TSF shall audit security relevant PKE events.

Table 6-62. Conformance with PP objectives for the TOE (Audit Package)

The following table provides the evidence of this conformance for the **Continuous Authentication Package**.

PP Objective Name	ST Objective Name	Objective Description

O.Continuous_I&A	O.Continuous_I&A	The TSF shall continuously authenticate the entity.

Table 6-63. Conformance with PP objectives for the TOE (Continuous Authentication Package)

Additionally the following objectives for the TOE have been defined: O.DIRECTIVE_ANNEX_III_PART_1a, O.DIRECTIVE_ANNEX_III_PART_1b O.DIRECTIVE_ANNEX_III_PART_1c, O.DIRECTIVE_ANNEX_III_PART_2, O.DIRECTIVE_ANNEX_IV_PART_b, O.DIRECTIVE_ANNEX_IV_PART_a, O.DIRECTIVE_ANNEX_IV_PART_c, O.DIRECTIVE_ANNEX_IV_PART_d, O.DIRECTIVE_ANNEX_IV_PART_e, O.DIRECTIVE_ANNEX_IV_PART_f and O.DIRECTIVE_ANNEX_IV_PART_g. The rationale included in section 5.4.1 Security Objectives Rationale demonstrates that the IT environment defined in this Security Target achieves these security objectives.

7. TOE Summary Specification

This section describes how the TOE satisfies all the Security Functional Requirements identified in Security Functional Requirements for the TOE, page 48, providing the general technical mechanism that the TOE uses for this purpose.

The TrustedX product described in this section is the TOE for this Security Target.

7.1. Certification Path Validation – Basic Package

The TOE implements the guidelines and recommendations for certification path validation contained in the [RFC5280]. All information processing implemented by the TrustedX services is X.509 and PKIX compliant.

The certification path validation is carried out when it is necessary to validate a certificate. Certificate validation is a service the TrustedX **TWS-DSV Service** offers to Web service/SOAP clients and an internal service offered to other TOE components, such as the TWS-AA service (e.g., for validating certificates that are considered authentication tokens) and the TWS-DE (e.g., for validating the encryption certificates used to encrypt data).

The interface of the TWS-DSV service follows the OASIS Digital Signature Service (DSS) specification [DSS]. A series of profiles for the most common scenarios has been defined to simplify client integration and interoperability.

FDP_CPD_(EXT).1 Certification path development

All the TOE's services described in TOE Description, page 5, make use of the certificate validation functionality. To verify a certificate, it is necessary to **develop a certification path** as defined in the X.509 standard.

This TOE has a specific behavior for path construction that allows the security functional requirements for the FDP_CPD_(EXT).1 family to be fulfilled.

Certification path development functionality is located in the function for verifying certificates. This function is used by all the TrustedX services, but it is located in the TrustedX Digital Signature Verification Service (TWS-DSV). Certificate verification carries out X.509 certificate path processing and, therefore, builds the certification path for the certificate to be validated.

The certification path is developed as per the following rules:

- If the certificate to be validated contains the authorityKeyIdentifier extension, this extension is used to build the link with the issuer certificate. In this case, all the possibilities specified in the [RFC5280] are supported: keyIdentifier, or issuer name and serial number of the issuer certificate. Where using the keyIdentifier field, the two methods included in the [RFC5280] are supported (keyIdentifier composed of the 160-bit SHA-1 hash of the value of the BIT STRING subjectPublicKey and keyIdentifier composed of a four-bit type field with the value 0100 followed by the least significant 60 bits of the SHA-1 hash of the value of the BIT STRING subjectPublicKey (excluding the tag, length and number of unused bits)).
- If the certificate to be validated does not contain the authorityKeyIdentifier extension, the issuer name contained in the certificate is used to find a CA certificate whose subject name is the same as this distinguished name and whose public key component can validate the signature in the certificate being evaluated.

Formally, because the authorityKeyIdentifier extension is optional, the information on the key of the issuer is not always is used (depending of the presence of this extension).

FDP_DAU_CPI_(EXT).1 Certification path initialization – basic

The root CA certificates are responsible for most of the system's security. These certificates must been registered in the service before any certificate related to these root CA certificates can be validated. **The management of trusted entities is the responsibility only of the Security Officer group (Web Services Consumer role)**. In the administration console (Trusted Entities option), this role can be used to create, delete and modify parameters of the root CA certificates defined in a TrustedX system.

Moreover, the TOE uses explicit trust development. This means that all CAs—roots and subordinates—must be explicitly introduced and configured in the system.

So TrustedX can actually use these CAs, besides introducing the CA certificates into the system, it is necessary to declare the CA certificates in a certificate validation policy. A certificate validation policy is an ordered set of validation rules in which each rule describes how to validate a certificate. Every certificate validation policy has an associated group of certification authorities (CAs). This group contains all the CAs that can be used to verify a certificate in accordance with a policy. Thus, a certificate validation policy defines how to process a certification path.

From a certification path processing perspective, there are three types of certificates:

- Self-signed trust anchor certificates (root CA certificates).
- Intermediate certificates (subordinate CA certificates).
- End certificates (End-entity certificates).

The security function included in the TOE that **verifies self-signed certificates** permits validating trust anchors (root CAs) based on the following checks:

- Verification that the subject and issuer distinguished names coincide.
- Verification of the signature using the subject public key included in the self-signed trust anchor.

- Verification that the validity period has not expired.

When a trust anchor certificate is validated, it is possible to obtain information on its fields (such as the distinguished name (DN), public key or the algorithm identifier).

The TOE's services provide the capability for validating a path as of a trusted time called the TOI. These services obtain the TOI from the local environment (machine where the TrustedX service is running), but the product provides the capability of working with **trusted times by synchronizing the local time with an NTP server**.

The TOE's system provides a command interpreter (shell) for the user to be able to manage and configure the system with ease. This shell can be used to manage IP addresses, services, logs, etc. The time and NTP commands can be used to configure the NTP servers that the TOE uses to synchronize its internal clock. Operations such as display, add, delete and synchronize NTP servers are supported by these commands.

FDP_DAU_CPV_(EXT).1 Certificate processing – basic

The end certificates of a certification path are the last certificates in the certification path and they are, typically, end-entity (i.e., not CA) certificates. The TOE supports **verifying an end certificate** using the certificate processing rules specified in the [RFC5280] and [X509] standards that consist of running basic checks, such as:

- Validation of the signature included in the certificate.
- Validation of the validity period of the certificate (the notBefore and notAfter fields).
- Validation of the names included in the certificate (issuer field is equal to the subject field of the issuer certificate).
- Processing of all the extensions marked as critical. If the TOE's services cannot process a critical extension being validated, the related certificate is rejected by the service.
- Validation of the revocation issues.

As well as these checks, the TOE applies **specific verification controls** that are carried out for certificates that need additional checks. An example of a certificate that needs additional check is the OCSP server certificate. In this case, the product processes the no-check extension as per [RFC2560], and, therefore, for these certificates, only the validity period is verified.

The checking of the revocation of a certificate can be configured via the **certificate validation policy**. A certificate validation policy is an ordered set of validation rules in which each rule describes how to validate a certificate. Every certificate validation policy has an associated group of certification authorities (CAs) that builds up one or several certification paths. This group contains all the CAs that can be used to verify a certificate in accordance with a policy. The Security Officer group (Web Services Consumer role) is responsible for managing all the validation, signature and other policies that are supported by the TrustedX product.

A validation policy is made up of a set of validation rules. Regardless of which rule is applied to validate a certificate, validation always entails verifying that the certificate has been issued by one of the CAs of the rule, verifying the integrity of the certificate by using the issuing authority's public key, verifying that the certificate's validity period

has not expired and verifying the revocation of the certificate. When a rule is not satisfied (e.g., the certificate has been revoked), the certificate is rejected.

Thus, a validation rule defines how to check if a certificate is still valid (i.e., that it has not been revoked), and if, when validating a certificate, it is necessary to validate its complete certificate chain. A rule describes a validation process that includes:

- Specifying which methods (i.e., CRL validation, OCSP response validation) must be used, in which order they have to be used and whether one or all of the methods have to be used.
- Where validation includes querying the CRL (i.e., CRL method), specifying which CA group is to be queried (i.e., the CA group that issued the CRL established by the rule, the group established by the validating certificate in the CRL Distribution Point extension), in which order these groups are to be queried (i.e., first, second) and if both CRL groups have to be queried or just one.
- Where validation includes the OCSP query (i.e., OCSP method), specifying which VA group is to be queried (i.e., the group established by the rule, the group established by the validating certificate in the authority information access extension), in which order these groups are to be queried (i.e., first, second) and if all or just one of the groups has to be queried.
- Specifying that certificate validation also entails validating the entire certificate chain and the certificate of the authority (i.e., CRL issuer, VA) that generated the evidence that proves the validity of the certificate.

The TOE's services allow inhibiting revocation checks. In this case, certificate processing is carried out in its entirety, but the revocation check is bypassed. This configuration is supported by making use of the rule mechanism being associated to a certificate validation policy. The <ValidationMechanisms> element contains information on the validation mechanisms that is used for determining the validity of a certificate (i.e., that it has not been revoked) through the execution of the rule. This element includes attributes and elements that can be configured to inhibit the revocation check. These attributes and elements are the following:

- check: An optional attribute that indicates if all the mechanisms or only one group must be used. This attribute can have the following values:
  - all: All validation mechanisms must be used (CRLs and OCSP responses).
  - any: The validation mechanism indicated by the first attribute is used. If this mechanism fails (i.e., it is not possible to access the necessary information), the mechanism indicated by the second attribute is invoked.
- first: An optional attribute that indicates which validation mechanism must be used first. This attribute supports the following values:
  - ocsp: OCSP responses are obtained first.
  - crl: certificate revocation lists (CRLs) are used first.
  - none: no validation mechanism is used first.
- second: An optional attribute that indicates which validation mechanism must be used second. This attribute supports the following values:

- ocsp: OCSP responses are obtained second.
- crl: certificate revocation lists (CRLs) are used second.
- none: no validation mechanism is second.

If the check attribute has the value "any", and the first and second attributes have the value "none", then the revocation check is bypassed.

FDP_DAU_CPV_(EXT).2 Intermediate certificate processing – basic

Intermediate certificates are the non-root certificates issued to the CAs. When a certification chain is validated, these certificates are also processed by the TOE's services. These validations include most of the validations applied to end certificates and those for the CA certificates. These controls include:

- The BasicConstraint extension must be present and must have the value "TRUE" in the cA field, and the pathLenConstraint field must be coherent regarding the length of the certification path.
- The keyUsage Extension must have the value "keyCertSign" set.

FDP_DAU_CPO_(EXT).1 Certification path output – basic

On verifying an X.509 certificate (TWS-DSV service), the TOE has all the information necessary for composing the service response (VerifyResponse). It generates an XML response document that contains fixed and customizable elements. The customizable elements are as follows:

- The validation policy identifier under which the validation process has been performed and the trust level information (i.e., on the third trusted parties that issued the certificates and the revocation evidences—CRLs or OCSPs).
- Any dynamically-obtained additional information for each response.

The <dss:VerifyResponse> element (XML response) includes several attributes containing information returned in the response, such as the mandatory <dss:Result> attribute, which contains the operation result.

Once the system has verified a signature—in a generic document or a certificate—it has all the data necessary to build the response (VerifyResponse). The response XML document is composed by joining the following parts:

- Fixed elements (non-customizable). These elements are always generated by the system and cannot be modified or customized.
- Customizable elements. These elements contain additional information on the certificates, revocation data (CRL or OCSP data), time-stamps and any other additional information that can be obtained dynamically for each response.

The **style templates** are for customizing the responses issued by both the signature verification and the certificate validation services. They consist of:

- An XML style sheet (XSLT standard), which explains how to render (i.e., transform) a data structure that the system manages internally when this structure is to be included in a service response.

- An XML document indicating (i.e., it acts as a filter) which elements of the results obtained from the above-mentioned style transformation are to be included in a service response.

7.2. Certification Path Validation – Basic Policy Package

FDP_DAU_CPI_(EXT).2 Certification path initialization – basic policy, FDP_DAU_CPO_(EXT).2 Certification path output – basic policy

For certification path validation, the TOE's certificate validation engine follows the recommendations on the management of certification policies in the [RFC5280]. Aspects of this management include:

- Certification path initialization. In the graphical console (or using any specific application and via the platform's TWS-EP component), it is possible to configure the initial certification policies so this information is used in line with section "6.1.2. Initialization" of the [RFC5280]. These initial policies are configured in the certification validation policy offered by the product.

  A certificate validation policy is an ordered set of validation rules in which each rule describes how to validate a certificate. Every certificate validation policy has an associated group of certification authorities. This group contains all the CAs that can be used to verify a certificate as per a policy. The inclusion of initial certification policies are managed in the context of a validation policy of the product.

- Certification path output. As per [RFC5280], when certification path processing finishes, it is possible to return specific information on the certification path validated (as indicated in "6.1.6. Outputs" of [RFC5280]). This returning facility is implemented through the TOE's style templates. The **style templates** are for customizing the responses issued by the signature verification and the certificate validation services.

  The processing rules of the certification policies and the use of style templates supports complying with the functional requirements included in this Security Target for certification path output in certification path validation.

7.3. PKI Signature Generation Package

FDP_ETC_SIG_(EXT).1 Export of PKI Signature

The TOE's services work with approved FIPS 140-2 level 3 and PKCS11-compatible cryptographic modules as hardware cryptographic modules for performing cryptographic functions. The security of the cryptography and, therefore, of the sensitive data to which this cryptography is applied is based on the robustness of FIPS 140-2 level 3 security requirements that these cryptographic modules fulfill. The

compatibility between the hardware cryptographic modules and the TOE is guaranteed by the use of [PKCS#11] standard.

Signature generation is supported by the **TrustedX TWS-DS service**. This component is a remote service for digitally signing data. The interface of this service follows the OASIS Digital Signature Service [DSS] specification. A series of profiles for the most-commonly used scenarios has been defined to simplify client integration and interoperability.

This service component is neutral from the point of view of the signer (or the entity that requires the signature) since any entity, once authenticated and authorized, can request the signature service by providing the identifier or selector of the key it wants to use. The platform stores the entity signature material in repositories in a cryptographic-protected mode, thus making it accessible in a uniform and controlled manner through the TWS-EP Service.

This package functionality includes generation of signature information that identifies the signer and is useful in efficient signature verification. For TrustedX, digital evidences are considered complementary advanced signature information.

When a digital signature is generated, the signer does not incorporate evidences in the document that grant the probative value of this signature. For subsequent signature verification, the evidences are archived as fundamental data that can later be extracted and used by third parties as probative elements.

Digital evidences include information on the moment the signature was produced, all the certificates that make up the trust chain and reliable information on the status of the certificates at that time. In the TOE, it is the non-repudiation service (TWS-DR) that is responsible for incorporating such evidences in signed documents.

Access to the TOE's digital signature service is accomplished using the [DSS] protocols according to a set of profiles that adapt to the different **signature formats** managed by the TOE:

- DSS formats for the TOE's signature generation service
    - CMS/PKCS#7 signatures

      These signatures are produced in accordance with [PKCS#7] and [CMS] specifications.

      Single and multiple signatures (sequential or parallel) are allowed in enveloped or detached signature format.
    - CAdES signatures

      These signatures are produced in accordance with the ETSI [ETSITS101733] standard. The following formats are supported for the signature generation service:
        - CAdES Basic Electronic Signature (CAdES-BES). Basic electronic signature that has no time-stamps in which the signer declares the signing time.
        - Electronic Signature with Time-Stamp (CAdES-T). Time-stamped electronic signature where the signing time is backed by a TSA.

- CAdES Explicit Policy-based Electronic Signature (CAdES-EPES). Extends the definition of an electronic signature to conform to the identified signature policy.

- XAdES signatures

  These signatures are produced in accordance with the ETSI [ETSITS101903] standard. The following formats are supported for the signature generation service:

  - XAdES Basic Electronic Signature (XAdES-BES). Basic electronic signature that has no time-stamps in which the signer declares the signing time.

  - Electronic Signature with Time-Stamp (XAdES-T). Time-stamped electronic signature where the signing time is backed by a TSA.

  - XAdES Explicit Policy-based Electronic Signature (XAdES-EPES). Extends the definition of an electronic signature to conform to the identified signature policy.

- XML-Dsig signatures

  These signatures are produced in accordance with the [XML-Dsig] specification.

  Enveloped, enveloping and detached signatures may be produced, including signatures by reference at any node of an XML document.

- S/MIMEv2 and S/MIMEv3 signatures

  Generation of secure e-mail messages according to the S/MIME formats defined by IETF and following the [RFC2311] for S/MIMEv2 and [RFC2633] for S/MIMEv3 specifications.

- PFD signatures

  This profile supports the generation of signed Adobe PDF documents according to [PDFSignature] IETF PDF-Sig recommendations.

- DSS Formats for the TOE's signature update service

  The TOE's signature update service supports the following standards:

  - Time-stamped electronic signature where the signing time is backed by a TSA. This element groups the CAdES-T and XAdES-T formats.

  - Electronic signature with complete validation data that adds information on the certificate chain and certificate revocation status information. This element groups the CAdES-C and XAdES-C formats.

  - Electronic signature that archive validation data. Once the complete validation data is added and time-stamped, the signature is updated with successive time-stamps before the cryptographic algorithms become weak or the digital certificates expire. This type of signature is the basis for the long-term validity of electronic signatures. This element groups the CAdES-A and XAdES-A formats.

- Electronic signature with extended validation data that may include validation data to provide additional protection against the compromising of the CA and to maintain the integrity of the validation data used. This element groups the CAdES-XL and XAdES-XL formats.

Note: Where the CAdES format is mentioned in the section above, the information given is also applicable to PDF Signatures and the S/MIME formats.

7.4. PKI Signature Verification Package

FDP_ITC_SIG_(EXT).1 Import of PKI Signature

The signature verification services provided by the TOE are grouped in i) the **Signature Verification Service (TWS-DSV component)** and ii) the **Signature Update Service with Non-Repudiation Evidence (TWS-DR component)**. The former group verifies "basic" signature information useful in efficient signature verification and the latter group completes/updates signatures (performed, for example, by the TWS-DS) with non-repudiable information adding a time-stamp, validation chain certificates and/or certificate status information.

The **TWS-DSV** is the TOE's digital signature verification service and is responsible for the verification of digital signatures. It verifies the validity of signatures generated by the digital signature service (TWS-DS) and those updated by the non-repudiation service (TWS-DR). For digitally-signed documents, the TWS-DSV also supports validating X.509-formatted digital certificates.

The TWS-DSV component uses the services of a trusted third party (TTP) via the OCSP (Online Certificate Status Protocol—[RFC2560]). It can also connect to a validation authority responsible for the online validation of the status of the certificates included in the signature, thereby providing direct access to the different types of revocation information sources, e.g., direct access to databases or to CRLs published online, etc.

The TWS-DSV interface follows the OASIS DSS (Digital Signature Service de OASIS) specification [DSS]. The TWS-DSV component supports the same digital signature formats as the TWS-DR and the TWS-DS, which are the following:

- CMS/PKCS#7 signatures. These signatures are verified in accordance with [PKCS#7] and [CMS] specifications.
- CAdES signatures. These signatures are verified in accordance with the ETSI [ETSITS101733] standard.
- XAdES signatures. These signatures are verified in accordance with the ETSI [ETSITS101903] standard.
- XML-Dsig signatures. These signatures are verified in accordance with [XML-Dsig] the specification.
- S/MIMEv2 and S/MIMEv3 signatures. These signatures are verified in accordance with the S/MIME formats defined by IETF, following the [RFC2311] for S/MIMEv2 and [RFC2633] for S/MIMEv3 specifications.

- PFD signatures. These signatures are verified in accordance with the Adobe PDF signature formats [PDFSignature] from the IETF PDF-Sig recommendations.
- X.509 certificates. The signatures in the X.509 certificates are verified in accordance with [X.509] recommendations.

The **TrustedX TWS-DR** service completes/updates signatures (performed, for example, by the TWS-DS) with non-repudiable information adding a time-stamp, validation chain certificates and/or certificate status information.

This means that the service checks whether the certificates used are recognized by the platform and that the digital signature generated—by completing/updating the input signature—includes validity evidences to prevent its repudiation. The maintenance and custody of these evidences is performed by another service, the TrustedX Data Signature Custody (TWS-DSC) service, which is in charge of requesting their custody and update before the keys and cryptographic material become vulnerable. This service provides custody for and maintains the validation data of digital signatures (long-term signatures) by periodically requesting that the TWS-DR refreshes this cryptographic material. TWS-DSC requests the renewal of electronic evidences before the time-stamp expires. This renewal is obtained by temporarily time-stamping the signature and evidence and by adding certificate information and status. This process is repeated every time the protection that is used for temporarily time-stamping the evidence becomes vulnerable.

The TWS-DR component adds the following evidences to a previously-generated signature:

- A time-stamp issued by a trusted third party, a TSA (time-stamping authority), is included in the signature. The time-stamp ensures that both the document's original data and the status token of certificates were generated before a specific date. The time-stamp format follows the standard defined in IETF TSP.
- Revocation Information: A token ensuring that the signature certificate is valid is included. This token is generated by a trusted third party: i) a VA (validation authority) in the format of an OCSP response, or ii) a CA (certification authority) in the format of a CRL.
- Certificate Information: The service automatically includes all certificates present in the certification chain of all involved signatures and tokens.

The TWS-DR component uses the services of trusted third parties (TTPs) through TSP (Time-Stamp Protocol—[RFC3161]) and OCSP (Online Certificate Status Protocol—[RFC2560]) protocols.

FDP_DAU_SIG_(EXT).1 Signature Blob Verification

In the signature generation service (TWS-DS), as in the signature verification and updating service (TWS-DSV and TWS-DR), the TOE's technology uses approved FIPS 140-2 level 3 and PKCS11-compatible cryptographic modules for performing cryptographic functions.

The use of secure cryptographic modules and the checks done in the signature verification process assure that the signature verifications are carried out with the strictest security guarantees.

The TWS-DSV component always executes a certificate validation path process to verify digital signatures on signed data as stated in the FDP_DAU_SIG_(EXT).1.1. The controls in the FDP_DAU_SIG_(EXT).1.2 and FDP_DAU_SIG_(EXT).1.3 requirements are also present.

The product can be configured to require a valid value for the KeyUsage extension of the certificate that the TOE must process. In order to establish coherence in the processing of the key usage extension, this control also requires the same values in the digital signature generation processes.

Another basic control in signature verification is the check that the subject distinguished name from the certification path validation is the same as the one included in the signed data.

7.5. PKI Encryption Using Key Transfer Algorithms Package

FDP_ETC_ENC_(EXT).1 Export of PKI Encryption – Key Transfer Algorithms

The TrustedX **TWS-DE** service supports the encryption and decryption of data according to IETF CMS and RSA PKCS #7 formats, the W3C XML-Enc XML encryption format, WS-Security SOAP message format, and S/MIMEv2 and S/MIMEv3 message formats. The component uses the TWS-EP services to obtain the encryption certificates of recipients.

The TWS-DE interface follows the Digital Encryption Service (DES) proprietary specification. The following profiles are supported:

- PKCS #7 and CMS Encryption. These profiles are supported in accordance with [PKCS#7] and [CMS] specifications.
- XML-Enc Encryption. These profiles are supported in accordance with [XMLEnc].
- WS-Security SOAP Encryption. These profiles are supported in accordance with [SOAPServicesSec].
- S/MIMEv2 and S/MIMEv3 Encryption. These profiles support secure e-mail messages according to the S/MIME formats defined by IETF, following the [RFC2311] for S/MIMEv2 and [RFC2633] for S/MIMEv3 specifications.

The TOE's services work with approved FIPS 140-2 level 3 and PKCS11-compatible cryptographic modules as hardware cryptographic modules for performing cryptographic functions. Security of cryptography and, therefore, of sensitive data to which this cryptography is applied, is based on the robustness of FIPS 140-2 security requirements that these cryptographic modules fulfill. The compatibility between the hardware cryptographic modules and the TOE is guaranteed by the use of [PKCS#11] standard. All hardware cryptography for the TWS-DE component is performed by the validated cryptographic modules making use of the [PKCS#11] specifications.

FDP_DAU_ENC_(EXT).1 PKI Encryption Verification – Key Transfer

The TrustedX **TWS-DE** service carries out a strict check of the identity of the agent that encrypts the data. This check entails verifying that the name associated with the certificate matches the name in the encrypted data and other controls on the extensions included in related certificate.

One of these controls is on the keyUsage extensions. This control is configurable and supports requiring a valid value for this extension in encryption and decryption processes.

7.6. PKI Decryption Using Key Transfer Algorithms Package

FDP_ITC_ENC_(EXT).1 Import of PKI Encryption – Key Transfer Algorithms

The TOE's services work with approved FIPS 140-2 level 3 and PKCS11-compatible cryptographic modules as hardware cryptographic modules for performing cryptographic functions. Security of cryptography and, therefore, of sensitive data to which this cryptography is applied is based on the robustness of FIPS 140-2 security requirements that these cryptographic modules fulfill. The compatibility between the hardware cryptographic modules and the TOE is guaranteed by the use of PKCS#11 standard. All hardware cryptography related to the TWS-DE component is performed by the validated cryptographic modules using the [PKCS#11] specifications.

7.7. PKI Based Entity Authentication Package

Authentication Mechanisms

This section deals with the FIA_UAU_SIG_(EXT).1 requirements family.

The TOE supports multiple authentication mechanisms (see Authentication Mechanisms in 7.12 Authentication and Access Control Package). Several of these authentication mechanisms are based on PKI. This section covers these mechanisms based on PKI.

FIA_UAU_SIG_(EXT).1 Entity Authentication

The TOE works with approved FIPS 140-2 level 3 and PKCS11-compatible cryptographic modules as hardware cryptographic modules for performing cryptographic functions. Security of cryptography and, therefore, of sensitive data to which this cryptography is applied is based on the robustness of FIPS 140-2 level 3 security requirements that these cryptographic modules fulfill. The compatibility between the hardware cryptographic modules and the TOE is guaranteed by the use of PKCS#11 standard. All hardware cryptography for the TOE's authentication services

is performed by the validated cryptographic modules using the [PKCS#11] specifications.

The use of secure cryptographic modules and the checks done in the signature verification process assure that the signature verifications are carried out with the strictest security guarantees.

Some of these controls are those in the FIA_UAU_SIG_(EXT).1.2 and FIA_UAU_SIG_(EXT).1.3 requirements.

The product can be configured to require a valid value for the KeyUsage extension of the certificate that the TOE must process in the authentication function.

Another basic control in signature verification is the check that the subject distinguished name from the certification path validation for the authentication process is the same as the one included in the signed data. Authentication of the entity in the TWS-AA service checks that the subject DN from the certification path validation matches that of the entity being authenticated. This verification is carried out in a coherent and integrated way with the **TrustedX Identification Policy** (see 7.12 Authentication and Access Control Package).

7.8. Online Certificate Status Protocol Client Package

FDP_DAU_OCS_(EXT).1 Basic OCSP Client

The TOE's technology supports using the Online Certificate Status Protocol (OCSP) to request the status of certificates from an OCSP responder. The TWS-DSV (TrustedX's signature verification service) is the TrustedX component that uses the OCSP.

TWS-DSV is responsible for the verification of digital signatures. It verifies the validity of signatures generated by the digital signature service (TWS-DS) and those updated by the non-repudiation service (TWS-DR). For this reason, the TWS-DSV uses the services of a trusted third party (TTP) via the OCSP (Online Certificate Status Protocol). It can connect to a validation authority responsible for the online validation of the status of the certificates included in the signature. The TWS-DSV follows the OCSP specified in [RFC2560].

The OCSP responders that the TWS-DSV component contacts are configured beforehand in the platform's console. The administration console supports managing trusted entities (certification authorities, validation authorities and time-stamp authorities). In this case, the OCSP responders must be registered as recognised VAs. In the validation of an OCSP response, the TrustedX technology verifies that the responder identification contained in this OCSP responder corresponds with the information previously registered (responder certificate).

The OCSP responder certificate is validated by the TWS-DSV TrustedX service and, therefore, all the security requirements offered by this service are assured. The OCSP responder certificates must also comply with the controls applied to validation servers (e.g., the extendedKeyUsage extension).

When an OCSP response is obtained from a remote VA, it is copied to a memory cache. When the product needs revocation information for a certificate, if it has been accessed previously, the memory cache is used. These caches have a lifetime; when the memory cache expires, the revocation information is deleted from the cache. The lifetime of the caches managed by the TOE can be configured by the TrustedX administrator.

All the cryptography for the TOE's OCSP validation services is performed by the validated cryptographic modules using the [PKCS#11] specifications. The TOE works with approved FIPS 140-2 level 3 and PKCS11-compatible cryptographic modules as hardware cryptographic modules for performing cryptographic functions.

With regards to the time checks on an OCSP response, the TOE can be configured to 1) not check the times included in the OCSP response (and, therefore, mitigate the T.DOS_OCSP threat regarding the loss of system availability) or 2) carry out checks on the times included in the OCSP response (and, therefore, mitigate the T.Replay_OCSP_Info threat regarding accepting an OCSP response from well before the TOI that results in accepting a revoked certificate).

The checks on these possibilities correspond to the ones included in the FDP_DAU_OCSP_(EXT).1.9 functional requirement. In this case, the offset (X) for the time controls can be configured. A large value for this X parameter supports not checking these times.

7.9. Certificate Revocation List Validation Package

FDP_DAU_CRL_(EXT).1 Basic CRL Checking

The TOE's technology supports processing CRLs following the specifications indicated in the X.509 standard [X509] and the [RFC5280] recommendation. The TWS-DSV (TrustedX's signature verification service) is the TrustedX component that uses the CRL checking functionality.

TWS-DSV is responsible for the verification of digital signatures. It verifies the validity of signatures generated by the digital signature service (TWS-DS) and those updated by the non-repudiation service (TWS-DR). For this reason, the TWS-DSV manages CRLs (i.e., it retrieves and validates them).

The TOE can obtain the CRLs from a location pointed to by the CRL distribution point extension of the certificate to be validated, or it can be obtained from a CRL distribution point configured by the administrator.

When a CA is registered in the administration console as a trusted entity, it is possible to assign a CRL distribution point (CDP) for this CA. In this case, this location is consulted from the validation rule of certificates for this CA, and the CRL obtained for revocation subjects is the one found in this location.

Another possibility is that the CRL consulted in the validation rule of a certificate is the URL whose URLs are indicated in the CRL distribution point extension of the certificate

being validated. These URLs can be configured from the <certExtension> attribute of the <Crls> element of the validation rule of the certificate validation policy.

When a CRL is downloaded from a remote repository, it is copied to a disc cache and a memory cache. When the product wants to access a CRL that has already been accessed, the memory cache is used. These caches have a lifetime; when the memory cache expires, it is updated from the disc cache. When the CRL maintained in the disc cache expires, it is updated from the remote repository. The lifetime for the caches managed by the TrustedX product can be configured by the TrustedX administrator.

Adhering completely to the X.509 standards [X509] and [RFC5280], TrustedX verifies the CRLs it uses to validate certificates. This entails the following checks:

- Time validation (check of the thisUpdate and nextUpdate fields).
- Validation of the signature contained in the CRL
- Validation of the CRL issuer certificate and the certification patch for this certificate. For this validation, specific controls are applied for checking critical extensions and that the distinguished names coincide.

All the hardware cryptography for the TWS-DSV is performed by the validated cryptographic modules making use of the [PKCS#11] specifications. The TOE works with approved FIPS 140-2 level 3 and PKCS11-compatible cryptographic modules as hardware cryptographic modules for performing cryptographic functions.

With regards to the time checks of a CRL, the TOE can be configured to 1) not check the times included in the CRL (thus mitigating the T.DOS_CRL threat on the loss of system availability), or 2) carry out checks on the times included in the CRLs (thus mitigating the T.Replay_Revoc_Info_CRL threat on accepting a CRL from well before the TOI resulting in accepting a revoked certificate).

The checks for these possibilities correspond to the ones included in the FDP_DAU_CRL_(EXT).1.6 functional requirement. In this case, the offset (X) for the time controls can be configured. A large value for this X parameter supports not checking these times.

7.10. Audit Package

FAU_GEN.1-NIAP-0407:2 Audit data generation – TOE, FAU_GEN.2-NIAP-0410:2 User identity association - TOE

The system generates log records for any relevant operations performed by end entities (e.g., starting and ending a session, modifying the system configuration, using a resource). It also provides statistics on the system and on the uses of the following services: encryption and decryption and digital-signature generation, verification and update (non-repudiation).

The TOE's log system logs the following information on events:

- Date: event date and time.

- Category: category of the log record.
- Description: description and result of the logged event.
- Event information: event parameters and a second description with more detailed information.
- User information: information on the user and the request that generated the event (identifier of the session opened by the user, distinguished name with which the user started the session, IP address of the machine from where the user invoked the action, unique identifier of the request, name of the action executed by the user, etc.).
- Service information: information on the service related to the event that generated the event (unique identifier of the service processing the event, name of the machine in which the service is requested, IP address of the machine where the service is executed, unique identifier of the thread processing the request, etc.).

From the TOE's administration console, it is possible to browse log records, define and execute log record search filters, and consult statistics on system operation and use of the encryption and decryption services and digital-signature generation, verification and update non-repudiation.

7.11. Continuous Authentication Package

FIA_UAU.6 Re-authenticating

TrustedX re-authenticates the user for each transaction. For performance purposes, the system maintains a cache of the different entity sessions. This means that when an entity accesses a service, it is not necessary to create a new session again (providing that the copy of the cache has not expired). In such cases, the entity authentication is as simple as:

- The use phase of the authentication mechanism: the entity presents its credentials (e.g., username and password, SSL/TLS certificate, digital signature) directly to the TWS-AA component.
- Checking that the following parameters match those registered in the session stored in the cache: the authentication policy, credentials, the authentication mechanism and the IP address of the entity (if IP restriction is enabled). If these match, the authorization phase commences, followed by service consumption. If, however, any of the parameters do not match, authentication is deemed as failed and the process ends.

The authentication policy applied to create a session cache determines the expiry of that session. While the session remains in the cache, the time conditions of the rules are not re-evaluated for every service request. These conditions are evaluated in the first service request (and they are not checked again until the session context has expired). The rules of the authorization policies can also include time restrictions. These rules are evaluated whenever an operation from a service is requested.

The TWS-AA re-authenticates the user when the lifetime of the session expires. In consequence, the TrustedX product offers a user the option to authenticate before each signature or to authenticate once before the creation of signatures within a certain timeframe (session cache).

In the product installation phase, it is required to specify if the product is to be installed in CC EAL4+ mode. If so, the maximum life of a session must be fixed; this value defines the overall upper limit for the session lifetimes that the product manages and cannot be re-configured again. Thus, authentication sessions can last for a time less or equal to this fixed value, but not more.

Where the product is not configured in CC EAL4+ mode, the life of the session can be fixed and modified to any value during the operation phase by the Security Officer group (Web Services Consumer role); the changes made to this security information is registered in the product logging system.

7.12. Authentication and Access Control Package

This section is on the following requirement families: FIA_UAU.1 Timing of authentication, FIA_UID.1 Timing of identification, FIA_UAU.4 Single-use authentication mechanisms and FIA_UAU.6 Re-authenticating.

The requirements on the authentication mechanisms based exclusively on PKI are explained in 7.7 PKI Based Entity Authentication Package.

Authentication and Authorization Component (TWS-AA)

The **TWS-AA** service is a key element of the TrustedX platform and is responsible for identifying, authenticating and authorizing entities that access the system. The service provided by the TWS-AA component is based on SAML assertions [SAML], enabling single-sign on (SSO) and federated identity management of other domains (between users, Web services and applications).

The service is based on a Secure Token Service (STS), which is based on username and password, X.509 and SAML (Security Assertion Markup Language) tokens as defined by OASIS WS-Security, Liberty Alliance or WS-Trust. A SAML token has a lifetime; when this time expires, the TWS-AA requires re-authentication. It is possible to configure the lifetime of a token using the authentication policy. The <AuthenticationLifetime> element of this policy supports managing these times.

Using the authentication token means that:

- Single sign-on (SSO) is available in the entire TrustedX platform.
- From the point of view of authentication, an end-to-end Web services security model may be implemented without completely relegating security to the transport mechanism. Other security services such as integrity and confidentiality can be relegated to the used transport protocol, generally SSL/TLS.

Authentication Mechanisms

The TOE **supports all authentication mechanisms**, and **implements some pre-defined mechanisms** (one-factor and two-factor authentication mechanisms). The following are the authentication mechanisms implemented:

- Authentication of end entities using credentials inside the SOAP message (following the [SOAPServicesSec] standard)
  - Username and password (over a SSL/TLS channel)
  - X.509 PKI-based signature of SOAP messages (XML-DSig WS-Security) (over a SSL/TLS channel)
- Authentication of end entities using credentials at the transport level
  - Authentication based on a X.509 PKI-based certificate (over a SSL/TLS channel)
- Authentication of authentication agents using credentials inside a proprietary SOAP message
  - Authentication based on a HMAC cryptographic function (over a SSL/TLS channel)

All these authentication mechanisms are considered completely implemented in the TOE since the authentication mechanism is executed with a proof-of-possession check, i.e., it is checked that the entity under authentication possesses a secret, namely, a password or a PKI secret key component.

The authentication mechanism based on a HMAC cryptographic function is used by the TOE's authentication agents. Using TrustedX authentication agents supports authenticating end entities with any additional authentication mechanism not initially-defined in the TrustedX platform. Therefore, the system supports additional authentication mechanisms (not included in the TOE) that it recognizes to be defined, which, in turn, must be implemented in the corresponding **authentication agents**.

So, the TOE includes the implemented authentication mechanisms and the authentication protocol with the additional authentication mechanisms that can be added later.

There are two types of authentication agents: internal and external agents. Internal agents are TrustedX internal mechanisms used by the system in any authentication mechanism not associated to an external agent. External agents are TrustedX mechanisms that support user authentication via any authentication mechanism that is not known beforehand or is not fully implemented by the TOE.

In the context of the external agents, the TOE provides support for other forms of PKI-based authentication mechanisms (besides the mechanisms of authentication identified above), but they need external agents to implement the proof-of-possession check. These authentication mechanisms are the following:

- PKI-Based PKCS#7/CMS formatted digital signature mechanism
- PKI-Based XML-DSig formatted digital signature mechanism
- PKI-Based X.509 certificate validation mechanism

Note that these three mechanisms are deactivated by default (the internal agent does not support them) and that a new external authentication agent is needed to completely implement these mechanisms.

See Authentication Agents for more detail on the types of agents.

Authentication based on HMAC cryptographic function

External authentication agents use an authentication mechanism based on the HMAC cryptographic function to authenticate for TrustedX (AuthN operations). The agents use AuthN authentication operations to register (authenticate) end entities in the TrustedX system.

The agent performs the authentication using a secret shared between it and the TWS-AA component. The following are the steps for this authentication:

1 The TWS-AA component generates a secret that is shared between it and the external agent.

2 Using secure off-line mechanisms to be defined, this secret is passed to the external agent. This secret key can only be accessed by the Security Officer group (Web Services Consumer role) and is managed as a shared secret (as a result, only the Security Officer group can deploy external agents).

3 The external agent applies the HMAC cryptographic function to the information on the authentication (the secret, a unique identifier of the agent, random data recently generated for the present interaction, date and time, and the IP address from where the authenticated end entity connects).

4 The external agent sends the HMAC result and all the information associated with the authentication (except the secret) as part of the AuthN protocol to the TWS-AA component.

5 When the TWS-AA component receives this information, it applies the HMAC cryptographic function to the received information and to the secret generated for this agent, and it compares the result with the received HMAC. If the result is successful, the TWS-AA has authenticated the user and allows the agent to register the end entity in the TrustedX system. From this moment on, this end entity can request services from the TOE.

When an end entity is authenticated via an external agent, information on the agent and end entity is logged (IP address of the agent, IP address of the endentity, etc.). Any further end-entity activity is also logged and can always be traced back to the external agent that handled the authentication.

Authentication confidentiality is protected through the use of a Secure Socket Layer between the agent and the TWS-AA service.

Conceptually, the internal agents are exactly the same as external agents, but the first fully implements the security mechanisms inside the TrustedX system and no secret is shared with an external component.

The HMAC authentication mechanism is part of the TOE. In addition, the following external authentication agents are also considered in the TOE:

- Username and password mechanism over SSL/TLS channel (mechanism also included as an internal authentication agent).
- Authentication mechanism based on a X.509 PKI-based certificate over SSL/TLS channel (mechanism also included as an internal authentication agent).

Where the use of other external authentication agents is considered, the authentication mechanism used by the TOE is the HMAC between the TWS-AA component and the agent; therefore, this case is considered as already included in the TOE.

To internally classify the security strength of authentication mechanisms, an abstract concept of "authentication level" that defines several authentication strengths has been defined.

Authentication Levels

The definition of four authentication levels means the security strength of different authentication mechanisms can be quantified so that the system can evaluate security irrespective of the selected mechanism. These levels are as follows:

- Low-level security mechanism (level 0): Less secure mechanisms (e.g., no authentication or unspecified authentication).
- Medium-level security mechanism (level 1). Mechanisms with an intermediate level of security (e.g., mechanisms that use username and password).
- High-level security mechanism (level 2). Mechanisms with a high level of security (e.g., Kerberos with symmetric keys or single-use passwords).
- Very high-level security mechanism (level 3). Mechanisms with a very high level of security (e.g., SSL/TLS, PKCS #7 or XML Digital Signature).

Each authentication mechanism is assigned a certain level of authentication, and, through the authentication phase, a certain level of authentication can be required to access a TrustedX service (e.g., digital signature or key management service) and a specific operation related to this service (e.g., digital signature operation of the TWS-DS service or key-pair generation of the TWS-KM service).

In the product installation phase, it must be specified if the product is to be installed in CC EAL4+ mode. If this mode is specified, the parameters indicated in the paragraph above (i.e., list of available authentication mechanisms and the assigning of authentication levels to authentication mechanisms) are fixed and cannot be modified in the product operation phase.

Where the product is not configured in CC EAL4+ mode, the authentication levels can be modified during the operation phase by the Security Officer group (Web Services Consumer role); changes made to this security information are recorded in the log system.

Thus, it is possible to configure a typical scenario in which a two-factor authentication mechanism is required (e.g., a smartcard containing a private key, an X.509 certificate and a password/PIN for authentication) for the operations that require the use of a private key for digital signing.

In this case, this two-factor authentication mechanism should be given a high (level 2) or very high (level 3) authentication level, while this selected authentication level is associated to certain operations of the TWS-DS and TWS-KM services (such as the digital-signature or key-pair generation operations).

Authentication Agents

The TWS-AA Authentication Service can register an entity in the platform (start an authentication session context). This service may delegate some of its responsibilities to an authentication agent. An authentication agent is, basically, a front-end that communicates using a protocol (AuthN and Logout authentication operations) that implements a specific authentication mechanism. The TWS-AA service component delegates the following functions to the authentication agents:

- Collection of credentials for identifying the entities to be authenticated.
- Optionally, validation of credentials (i.e., authentication).
- Delivery of the collected data to the TWS-AA service component, and, where appropriate, the result of the authentication, so that the service may determine a user authentication context in the platform.

Thus, agents collaborate with the TWS-AA service component. The agents collect the credentials and, where necessary, they verify them in accordance with the supported authentication protocols. Usually, externally-implemented authentication agents must include a proof-of-possession check for the given mechanism.

The system supports as many external agents as necessary (e.g., biometric devices and one-time password devices). Thus, the TOE supports all authentication mechanisms.

By default, the system includes two internal authentication agents: the gateway's internal agent and the services' internal agents. The first one is used to obtain the credentials of an end entity (received through the gateway) and send them to the AA component for this component to validate these credentials. The second type is for authentications that are initiated from system internal components (e.g., tasks of the DSC Service) that also have an own entity (for instance CN=Digital Signature Custody) and that initiate sessions in the AA component (the same case as that of the end entities).

The gateway's internal agent implements the following authentication mechanisms:

- Login and password mechanism in a SOAP message request over a SSL/TLS channel. Classified by default as authentication level 1 or medium.
- X.509 PKI-based digital signature (WS-Security) mechanism in a SOAP message request. Classified by default as authentication level 2 or high.
- SSL/TLS client-authentication mechanism (authentication based on a X.509 PKI-based certificate). Classified by default as authentication level 2 or high.

In the TrustedX platform, the authentication process always occurs through the interaction of an authentication agent and the TWS-AA component. There are, however, different ways of interacting to achieve the authentication; it can be done:

- Explicitly via an external agent: an external agent implements the authentication mechanism and, when executed successfully, the agent requests the AuthN (agent mode) service from the TWS-AA component to register the entity being authenticated. If the TWS-AA component correctly and successfully processes the request, the entity is registered in the system and an authentication context is created. The AuthN service responds with an assertion, reference or artifact for that context. The entity, or application on its behalf, performs further service requests by including the assertion, reference or artifact without having to explicitly authenticate and so achieving SSO (single sign-on).

- Explicitly via the internal agent: An entity, or an application on its behalf, sends an explicit authentication service request (AuthN in direct mode) to the TrustedX platform. This AuthN request is internally routed to the internal agent that executes the selected authentication mechanisms. If the process is successful, the internal agent tries to register the entity in the system, which, in turn, assigns an authentication context for that entity. The AuthN service responds with an assertion, reference or artifact for that context. The entity, or application on its behalf, performs further service requests by including the assertion, reference or artifact without having to explicitly authenticate and so achieving SSO (single sign-on).

- Implicitly through the internal agent: An entity, or an application on its behalf, sends a service request (e.g., a sign request) to the TrustedX platform. The GW (XML Gateway) component extracts the authentication information from the service request and routes it to the internal agent that executes the selected authentication mechanisms. If successful, the internal agent tries to register the entity in the system, which, in turn, if the process is successful, then assigns an authentication context for that entity. The service request is executed and a service response is returned. Note that in this case, because the authentication process occurs implicitly, there is no response containing an assertion, reference or artifact.

When using the internal agent (explicitly or implicitly), the authentication can be performed by passing the credentials at the transport level instead of (as explained above) in the SOAP service request message. The TOE supports entity authentication through the SSL/TLS protocol over a HTTP connection. An entity, or an application on its behalf, can set up an SSL/TLS connection to the TOE (via the GW component) with client–authentication exchange. The internal GW component executes the SSL/TLS protocol against the entity and requests client credentials (i.e., an X.509 PKI-based certificate). If the SSL/TLS exchange concludes successfully, the internal agent tries to register the entity in the system, which, in turn, if the process is successful, then assigns an authentication context for that entity. Thus, an authenticated and secure channel is created in which the entity, or application on its behalf, can perform service requests.

When authenticated, an entity, or application on its behalf, can invoke SOAP service request messages including an assertion, reference or artifact as a pointer to an authentication context that exists from of a previous authentication process, thus achieving single sign-on.

The figure below shows the described interactions and use cases.

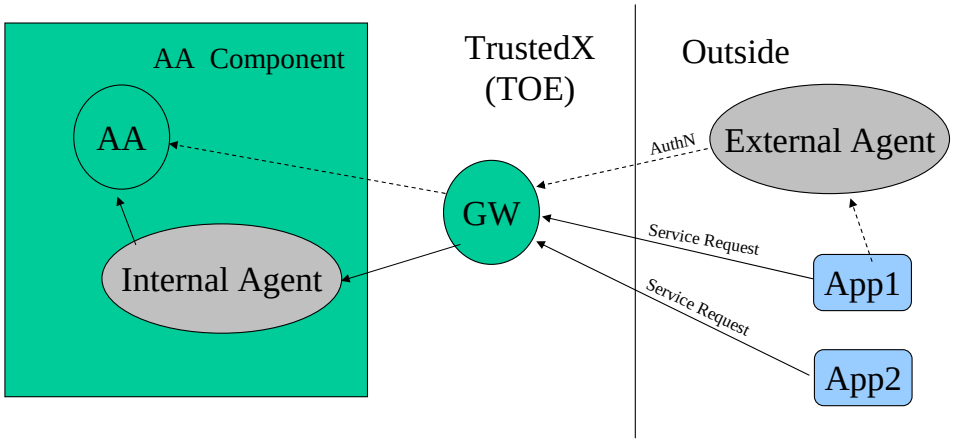

Figure 7-1. TWS-AA and Agents interactions

From a security point of view, the TWS-AA component only trusts external agents that have been previously defined in the system. In an external agent set-up, a privileged user defines the name of the agent, the authentication mechanisms that it implements, the authentication policies it supports, etc., and it also establishes a symmetric secret key that is only to be shared with the deployed agent that contacts the TrustedX platform. The external agent protects the communication channel using SSL/TLS and signs the SOAP requests (AuthN in agent mode) using the symmetric shared key. Thus, authorized agents can request authentication on behalf of entities.

The syntax, interface and protocol of the AuthN service are well-documented and constitute the basis for extending the TrustedX platform with new (even currently unknown) authentication mechanisms.

By default, the TrustedX platform comes with one authentication agent, the internal agent, which cannot be deleted or modified. When the product is installed and configured to work in the CC EAL4+ mode, during the initial setup phase, the operator can define one or more additional external agents in the system. However, when this configuration phase has concluded, the product cannot be re-configured, and the only available agents (and their configurations) in the system are those defined in the configuration phase, and they cannot be changed.

Authentication Levels and Mechanisms Mapping

As SSL/TLS, PKCS#7 and XML-DSig formatted mechanisms can be considered by default as level 2 or high security mechanisms, these mechanisms amount to PKI private key operations, thus demonstrating proof of possession (something you have). It is also usual to use a password or secret (something you know) to protect the access to the PKI private key. Thus, these become two-factor mechanisms. If the PKI material is securely stored and protected in a smart card system, these can be considered very-high security or level 3 mechanisms.

The other mechanisms, i.e., login and password and X.509 certificate validation, can be classified as level 0 or 1 by default since they are also considered 0 or 1 factor authentication mechanisms.

When the product is installed and configured to work in CC EAL4+ mode, during the initial setup phase, the operator can assign authentication level settings for the different mechanisms. However, when this configuration phase concludes and has been accepted, the product cannot be re-configured, and the values and mappings for the authentication mechanisms and levels become permanent and inalterable.

This behavior can be used to enforce that some services, for instance, digital signing or any other private key operation using material managed by the product, can only be initiated and controlled by users that have been identified and authenticated using high (level 2) or very high (level 3) authentication mechanisms.

Entity Identification

In the TOE, after authentication, an **Identification Policy** is used to map the name of an entity (expressed in the credential it uses in the authentication process) to a local name in the system.

Once the authentication process is correctly completed, the system can identify the authenticated entity because the credential used by the entity includes a name that identifies the entity. However, in situations such as the one described below, it is advisable to have an additional identification policy, such as the one provided by the identification policies.

- Entities can have several types of credentials (username and password, certificates, one-time password, etc.); however, they all refer to the same physical entity.
- Entities can have several identities that may be used in the service request; however, it may be preferable to refer locally to all these identities as one by mentioning the same entity. This is identity federation, alias assignment, etc.
- For some types of integration, it is necessary to convert the authenticated name format (identity) to a local format because there are existing databases and/or identity repositories.
- The session context of the authenticated entity can be expanded with additional information on attributes associated to the entity. This functionality may, for example, be useful when generating SAML attribute tokens.

The identification phase, which is characterized by an identification policy, takes place after the authentication phase. In fact, the identification policy complements the authentication policy. However, the identification policy can be omitted, meaning that the identification phase in the authentication process can be omitted.

An identification filter is used to define an identification policy. The available filters enable obtaining a final identity (final distinguished name or DN) and, optionally, other attributes, taking the following data as the starting point:

- Initial identity (initial distinguished name or DN0) taken from the authentication phase.
- Identity data of the session context obtained in the authentication process (e.g., the credentials).

Figure 7-1. The identification process

Logout Service

An entity, or application on its behalf, may send a logout service request to the TrustedX platform to release a previously-established authentication context. As with any other service request, the logout service requires an active authentication context to release it. If successfully executed, the logout service releases the context, after which the entity cannot perform more service requests using this context, thus finishing the single sign-on session.

After being logged out, if the entity wants to perform more service requests, it needs to execute a new authentication process.

Re-Authentication

See 7.11 Continuous Authentication Package.

Authentication and Access Control in the Administration Console

The authentication process is also applicable to the **TrustedX Administration Console**. The system has an advanced graphical user interface (GUI) for administering and accessing all the system information in a uniform and centralized manner via a Web browser.

The administration functions of the platform's console provide for:

- End-entity management: management of groups of privileged users, users, applications or services and the groups of end entities.
- Management of trusted entities: management of certification, validation and time-stamp authorities.
- Management of authentication and authorization policies: for defining a set of rules and actions that are applied depending on the type of authentication, authenticated entity and resource requested.
- Management of digital signature generation policies: for defining and making changes to the policies applied for generating digital signatures.
- Management of digital signature verification policies: for defining and managing the digital signature verification policies, including the digital certificate validation policies.

- System configuration management: for defining the configuration of the platform's own service components and the configuration of the databases, the directory, etc.
- Management of logs and audits: for browsing all the events generated by all the platform's service components.

The end entities must start a session in the system before performing an operation. In order to start a session, the entity must present credentials for the system to authenticate them. Once the credentials have been successfully authenticated, the session remains active until one of the following takes place:

- The authentication validity expires.
- The end entity or the system decides to close the session.

The TrustedX administration console supports the following authentication mechanisms:

- Authentication with username and password (over SSL/TLS).
- Authentication with an SSL certificate installed in the browser.

In both cases, the HTTPS protocol is used to establish the connection (with the objective of the SSL server certificate guaranteeing the confidentiality of the transmitted data).

For the TWS-AA service and the administration console, the identification mechanism and request for password (where the username–password mechanism is used) are performed before the user is authenticated.

For the administration console, a TLS secure channel [TLS] is established before the user is identified. In addition, the TrustedX console requests the username or certificate to identify this user.

With respect to the TWS-AA component, the administration console uses the internal agent for authentication and, thus, all the requirements and restrictions described above apply in this case. The figure below shows the relationship between the AA component, the internal agent and the administration console.

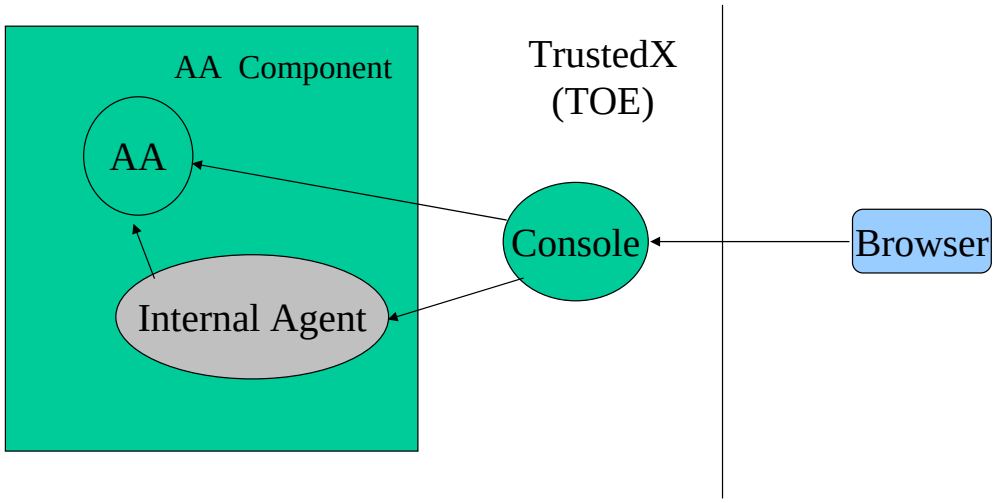

Figure 7-1. TWS-AA, internal agent and administration console interactions

7.13. Annex III and Annex IV of the [EUROPEAN_DIRECTIVE]

This section describes how the TOE satisfies all the Security Functional Requirements identified in section Security Functional Requirements for the TOE, page 48, and derived from Annex III ("Requirements for Secure Signature-Creation Devices") and Annex IV ("Recommendations for Secure Signature Verification") of the [EUROPEAN_DIRECTIVE].

Therefore, this section explains how the TOE complies with the functional requirements in 5.2.1.12 Annex III and Annex IV of the [EUROPEAN_DIRECTIVE], 5.2.1.7 PKI Based Entity Authentication Package and 5.2.1.11 Continuous Authentication Package.

Requirements for the secrecy of the user private key, robustness of the cryptographic algorithms and security of the generated signatures: FPT_TEE.1

The **FPT_TEE.1** requirements force the use of an approved FIPS 140-2 level 3 cryptographic module for the secrecy of the user private key, robustness of the cryptographic algorithms and the security of the generated signatures.

The TOE's services work with approved FIPS 140-2 level 3 and PKCS11-compatible cryptographic modules as hardware cryptographic modules for performing cryptographic functions.

Regarding the protection of the user private keys, a **FIPS 140-2 level 3 approved HSM is required**. Security of cryptography and, therefore, of sensitive data to which this cryptography is applied is based on the robustness of FIPS 140-2 level 3 security requirements that these cryptographic modules fulfill.

The use of FIPS 140-2 level 3 approved HSMs supports protecting the user keys through tamper-evident physical security mechanisms and preventing the intruder from accessing critical security parameters held in the cryptographic module.

The TOE detects if the cryptographic module that protects the user private keys has been FIPS 140-2 level 3 configured. If this configuration is not detected, the TOE's services are blocked until the required FIPS configuration is achieved.

Requirements for the exclusive use of the private key: FDP_ACC.1, FDP_ACF.1, FIA_UAU.1, FIA_UID.1 and FPT_TEE.1

FDP_ACC.1 and FDP_ACF.1 requirements

The TOE guarantees the exclusivity of the user private key. Basically, the TOE has several security mechanisms for protecting the sensitive information of users from illegitimate users.

One important mechanism from the security point of view that helps in the implementation of the exclusivity of the user private key is the **access control** that the TOE requires for using the private key. The **FDP_ACC.1** and **FDP_ACF.1** requirements are related to the access control mechanisms implemented by this TOE.

The TOE implements a strict access control based on only allowing authorized users (that have the correct permissions) to access resources though the operations offered by the system.

From a security point of view, the TOE's purpose and the security objectives of this Security Target, the user private keys are especially-critical resources. The TOE protects the user private key from being used by illegitimate users, ensuring exclusive use of the private key by the owner user. To protect this sensitive information, a strict access control policy (TrustedX access control policy) for the operations where a private key is involved is implemented.

The **FMT_MSA.3**, **FMT_MSA.1**, **FMT_SMR.1** and **FMT_SMF.1** requirements are related to the FDP_ACC.1 and FDP_ACF.1 requirements (which are Common Criteria dependencies). These requirements collaborate indirectly in compliance with the access control requirements.

For the FMT_SMR.1 (security roles) requirements, the TOE implements role management to fortify the authentication and access control of the product. The TOE's roles are a group of entities that can be granted a set of privileges. The system implicitly recognizes a set of roles that are entities that are granted privileges for administering the system or consuming a set of services offered by the product. These roles are the following:

- Web Services Consumer: user that accesses and administers the TOE and consumes Web services through the Web interface or the SOAP interface (based on the permissions assigned).
- Administrator through the Command Shell: user that administers TOE configuration parameters through a command shell interface.

Additionally, the system implicitly recognizes the members of specific groups of users as being privileged entities: entities that are granted privileges for administering the system. These groups are the following:

- First System Administrator

  This is the role of the first system user (i.e., the user who installs and starts up the system for the first time). The First System Administrator is the only user who can recover the system in the event of a disaster or serious failure.

  This user can perform the same tasks as a system administrator, in addition to being able to create the following users:

  - First Security Officer.
  - First System Auditor.
  - System Administrators.

- First Security Officer

  This is the role of the first system user who can perform Security Officer tasks. This user is created by the First System Administrator. In addition to performing Security Officer tasks, this user can create new Security Officer users.

- First System Auditor

  This is the role of the first system user who can perform System Auditor tasks.

- System Administrators

  System Administrators perform actions that include:

  - Mapping devices.
  - Registering repositories (databases, LDAP and files), cryptographic hardware, service components, end users, etc.

  Thus, users with this role can perform tasks that involve the installation, configuration and maintenance of the operating part of the platform.

- Security Officers

  Security Officers can perform actions such as the following on policies, entities and groups:

  - Signature and validation policies.
  - Trusted (TSA, VA) third parties.
  - Trusted certificates (of both root and subordinate CAs).
  - Authentication and authorization policies.
  - User groups.

  Thus, they perform tasks that involve the installation, configuration and maintenance of the platform's security parameters.

- System Auditors

  System Auditors are granted read-only access to:

- All the system configuration information.
- All traces and log records generated by the system.

These roles can be configured with the console's graphical interface.

For the FMT_MSA.1 (management of security attributes), FMT_MSA.3 (static attribute initialization) and FMT_SMF.1 (specification of management functions) requirements, the TOE implements controlled management of the security attributes and security management functions.

The TOE supports the controlled management of the security attributes of the entities and the system. The Web Services Consumer role (Security Officer Group) manages the system security parameters, these users authenticating for executing a security operation of this type. In addition, all these security operations are logged so that traces of all the executed changes are maintained. Users manage their own sensitive information (keys, certificates) stored in a private secure repository in the system. Any operation on this information is carried out by the owner of this repository.

All the TOE's security attributes have restrictive values by default. In the initialization of the security attributes, the Web Services Consumer role (Security Officer Group) can replace the default values with others; this operation is logged in a security log file.

FIA_UAU.1 and FIA_UID.1 requirements

The TOE guarantees that the signature key cannot be used until the holder is identified and authenticated. The authentication mechanism responds to any authentication requirement that can be imposed on an electronic signature product.

The TOE supports several authentication mechanisms and includes a set of them (single- and multiple-factor authentication mechanisms). The system also supports additional (not included in this TOE) authentication mechanisms that it recognizes to be defined, which, in turn, must be implemented in the corresponding authentication agents.

For each authentication mechanisms, the TOE assigns a certain authentication level (i.e., lower for single-factor authentication and higher for multiple-factor authentication). This allows defining the minimum authentication mechanism level required for the operations that require the use of a private key (e.g., cryptographic smartcard containing PIN-protected RSA keys).

In the TOE, it is possible to define a specific authentication level required for a particular TOE service (e.g., digital signature service or key management service) and to associate it to a specific operation of this service (e.g., digital signature operation of the TWS-DS service or key pair generation of the TWS-KM service).

In the above scenario, this multiple-factor authentication mechanism should be associated to a high or very high authentication level; the selected authentication level should also be associated to certain operations of the TWS-DS and TWS-KM services (e.g., the digital-signature or key-pair generation operation).

Finally, the TOE can be defined to require authentication for each private key operation or to allow the use of the private key within a certain timeframe. This allows TrustedX to be used for bulk/batch signature purposes.

See Authentication Mechanisms for more information on the access control applied to the use of private keys.

FPT_TEE.1 requirements

The TOE ensures that any resource sensitive content is only available to users explicitly granted access to the data. The most important sensitive content is the user private key. The TOE ensures the confidentiality of this key by delegating its custody to the hardware cryptographic module (FIPS 140-2 Level 3 certified). For the storage of the private key, the TOE can operate in one of the following modes:

- The private key cannot leave the HSM. In this case, the protection is provided by the security mechanisms of the HSM (FIPS 140-2 Level 3).
- The private key leaves the HSM. In this case, the key is never decrypted (in clear) outside the HSM. The private key only leaves the HSM when it is encrypted. In this case, the protection is also provided by the HSM.

The TOE forces the use of a high level of security in the cryptographic devices. The TOE detects if the cryptographic module is **FIPS 140-2 level 3** configured. Each time the TOE's services are started, a security test is launched to check that the cryptographic device is configured in this way. If the test fails (the cryptographic device is not configured as a level 3 FIPS 140-2 device), SOAP access is blocked (users cannot access the TOE's services). If it is configured in this way, this guarantees controlled access to the private key from the cryptographic device environment.

The TrustedX component for the management of keys is the **TWS-KM**. This TOE component is a key management service based on the XML Key Management Service (XKMS) standard. It is the responsible for carrying out the operations identified above. Through the use of XML, this service provides access to the generation, registration, browsing and revocation (deletion), etc. of X.509 keys. So, the TWS-KM service is responsible for managing the keystores of the TrustedX users.

As the TWS-KM component implements strict access control, only the user–owner of the keystore can access the services offered by the TWS-KM component that manages the sensitive material for this user.

The access control implemented to support the access control policy referenced is based on the authentication provided by the TWS-AA component. This authentication is described in Authentication Mechanisms (TWS-AA service). So, all operations carried out by the TWS-KM component are considered PKI private/public key operations and are therefore subject to the same authentication and authorization controls as for when generating a digital signature (TWS-DS component) when the product is configured to work in CC EAL4+ mode.

Requirements for the integrity of the data exchanged between the TOE and the user requesting the generation/verification of a digital signature: FTP_TRP.1

The TOE provides assured identification of its end points and protection of channel data from modification and disclosure. This security functionality can be summarized as the following:

- Integrity mechanisms that assure that the user data (data sent from/to the user) is not modified during the communication with the TOE product.

- Authentication mechanisms applied to the communications between the user and the TOE product.
- Confidentiality mechanisms that assure that sensitive information is not disclosed during the communication between the user and the TOE.

Therefore, any communication between the user and the TOE (to/from a user) is protected. The SSL/TLS protocol is used to provide the security services related to this protection.

Thus, when a user sends data to be signed to the TOE's TWS-DS component, this data is protected by the integrity and confidentiality services via the SSL/TLS protocol. Likewise, when the TOE's TWS-DSV component sends back information to the user on the digital signature verification process, this information is also protected by the SSL/TLS protocol.

The **FTP_TRP.1** requirements force the implementation of the SSL/TLS security protocols in any communication between the TOE and the user requesting the generation/verification of a digital signature.

The SSL/TLS security protocols support guaranteeing the integrity and the confidentiality of all the data exchanged between the user and the TOE and, in particular, the following data:

- Data to be signed (sent from the user to the TOE).
- Data used for verifying the signature (sent from the TOE to the user).
- Result of the signature validation (sent from the TOE to the user).
- Content of the signed data to be verified (sent from the TOE to the user).
- Signer's identity (sent from the TOE to the user).

In the TOE's context, the integrity is assured by means of the A.PHYSICAL assumption that assumes that the environment provides the TOE with appropriate physical security.

Requirements for the reliable verification of the digital signatures: FDP_DAU_SIG_(EXT).1, FDP_ITC_SIG_(EXT).1 and FPT_TEE.1

The TOE provides reliable verification of digital signatures via the following technical properties:

- The TOE guarantees the operation of its services in an execution environment with a FIPS 140-2 level 3 security level. This assures that the validation of the signatures is carried out in a suitable environment complying with all the guarantees provided by the FIPS 140-2 level 3 certification. This property is assured by the FPT_TEE.1 requirements.
- The TOE verifies all the information on the signature (building of certification paths, processing of the X.509 extensions and all aspects relating to the [X.509] standard) and invokes cryptographic hardware for the cryptographic verification of the signatures. This property is assured by the FDP_DAU_SIG_(EXT).1 and FDP_ITC_SIG_(EXT).1requirements.

Requirements for the reliable verification of the certificates related to digital signatures: FDP_DAU_CPV_(EXT).1, FDP_DAU_CPI_(EXT).1, FDP_DAU_CPO_(EXT).1, FDP_CPD_(EXT).1, FDP_DAU_CPV_(EXT).2 and FPT_TEE.1

The TOE provides reliable verification of the certificates for related to digital signatures by means of the following technical properties:

- The TOE guarantees the operation of its services in an execution environment with a FIPS 140-2 level 3 security level. This assures that the validation of the certificates is carried out in a suitable environment complying with all the guarantees provided by the FIPS 140-2 level 3 certification. This property is assured by the FPT_TEE.1 requirements.
- The TOE verifies all the information related to the certificate (building of certification paths, processing of the X.509 extensions and all aspects relating to the [X.509] standard) and invokes a cryptographic hardware device for cryptographically verifying the signature included in the certificate. This property is assured by the FDP_DAU_CPV_(EXT).1, FDP_DAU_CPI_(EXT).1, FDP_DAU_CPO_(EXT).1, FDP_CPD_(EXT).1 and FDP_DAU_CPV_(EXT).2 requirements.

Requirements for the support of pseudonyms: FDP_ITC_SIG_(EXT).1

The TOE ensures that during the signature-verification process the use of pseudonyms is supported. The TOE assures this objective by supporting the use of X.509 public key certificates in the signatures verification process (the X.509 certificates support extensions and attributes in which it is possible to indicate all type of information, including pseudonyms). This property is assured by the implementation of the FDP_ITC_SIG_(EXT).1 requirements.

Requirements for the security audit events: FAU_GEN.1-NIAP-0407:2

The TOE ensures with reasonable certainty that any security-relevant change is detected. The TOE assures this objective by implementing mechanisms for logging and detecting security events. This property is assured by the implementation of the FAU_GEN.1-NIAP-0407:2 requirements.

CHAPTER 8

8. Bibliography, Definitions and Acronyms

8.1. Bibliography

The following documents are referenced in this document:

Reference	*Referenced document*
[X509]	X509v3: ITU-T Recommendation X.509 | ISO/IEC International Standard 9594-8, Information Technology – Open Systems Interconnection – The Directory: Authentication Framework
[RFC2560]	RFC 2560: Online Certificate Status Protocol - OCSP
[RFC3161]	RFC 3161: Time-Stamp Protocol (TSP)
[RFC5280]	RFC 5280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile
[DSS]	Digital Signature Service Core Protocols, Elements, and Bindings. Working Draft 27, 4 October 2004
[PKCS#7]	PKCS #7: Cryptographic Message Syntax Standard
[PKCS#11]	PKCS #11: Cryptographic Token Interface Standard
[CMS]	RFC 3369: Cryptographic Message Syntax (CMS)
[ETSITS101733]	ETSI TS 101 733 V1.7.3 (2007-01). Electronic Signatures and Infrastructures (ESI); CMS Advanced Electronic Signatures (CAdES)
[ETSITS101903]	ETSI TS 101 903 V1.2.2 (2004-04). XML Advanced Electronic Signatures (XAdES)
[XML-Dsig]	W3C recommendation. XML Signature Syntax and Processing (Second Edition). W3C Recommendation 10 June 2008
[RFC2311]	RFC 2311: S/MIME Version 2 Message Specification
[RFC2633]	RFC 2633: S/MIME Version 3 Message Specification

Reference	Referenced document
[PDFSignature]	PDF Reference. Fourth edition. Adobe Portable Document Format. Version 1.5
[XMLEnc]	XML Encryption Syntax and Processing. W3C Recommendation 10 December 2002
[SAML]	Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V1.1 OASIS Standard, 2 September 2003
[SOAPServicesSec]	Web Services Security: SOAP Message Security 1.1(WS-Security 2004) OASIS Standard Specification, 1 February 2006
[EUROPEAN_DIRECTIVE]	DIRECTIVE 1999/93/EC OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 13 December 1999 on a Community framework for electronic signatures
[PKE_PP]	US Government Family of Protection Profiles. Public Key-Enabled Applications For Basic Robustness Environments. May 1, 2007. Version 2.8
[TLS]	RFC 2246: The TLS Protocol Version 1.0
[LEGISLATION_DSS_OASIS]	http://www.oasis-open.org/committees/download.php/30460/EU%20Electronic%20Signature%20Legislation%20Requirements%20for%20DSS%20v%200%200%204.doc
[FESA]	http://www.fesa.eu/public-documents/PublicStatement-ServerBasedSignatureServices-20051027.pdf

8.2. Definitions

Access -- Interaction between an entity and an object that results in the flow or modification of data.

Access Control -- Security service that controls the use of resources (including hardware and software) and the disclosure and modification of data.(including stored and communicated)

Accountability -- Property that allows activities in an IT system to be traced to the entity responsible for the activity.

Administrator -- A user who has been specifically granted the authority to manage some portion or all of the TOE and whose actions may affect the TSP. Administrators may possess special privileges that provide capabilities to override portions of the TSP.

Assurance -- A measure of confidence that the security features of an IT system are sufficient to enforce its' security policy.

Attack -- An intentional act attempting to violate the security policy of an IT system.

Authentication -- Security measure that verifies a claimed identity.

Authentication data -- Information used to verify a claimed identity.

Authorization -- Permission, granted by an entity authorized to do so, to perform functions and access data.

Authorized user -- An authenticated user who may, in accordance with the TSP, perform an operation.

Availability – Timely (according to a defined metric), reliable access to IT resources.

Certificate Revocation List (CRL) -- A list of the certificates that relying parties should no longer use or trust because the certificates have been revoked. Normally, the CA that issued the certificates also issues the CRL. The CA may assign responsibility for issuing CRLs to another entity. The CRL is a data structure that the issuer digitally signed.

Compromise -- Violation of a security policy.

Confidentiality -- A security policy pertaining to disclosure of data.

Cryptographic boundary -- An explicitly defined contiguous perimeter that establishes the physical bounds (for hardware) or logical bounds (for software) of a cryptographic module.

Cryptographic key (key) -- A parameter used in conjunction with a cryptographic algorithm that determines [7]:

- the transformation of plaintext data into ciphertext data,
- the transformation of cipher text data into plaintext data,
- a digital signature computed from data,
- the verification of a digital signature computed from data, or
- a digital authentication code computed from data.

Cryptographic Module -- The set of hardware, software, firmware, or some combination thereof that implements cryptographic logic or processes, including cryptographic algorithms, and is contained within the cryptographic boundary of the module.

Digital Signature (or Signature) -- A value determined from first computing a hash of the data to be signed and then applying a cryptographic function (the signature algorithm) to a hash value using the private key of the signer.

Digitally Signed Data -- A collection of data (the signed data) and a value (the digital signature) computed from that data. The signature is the result of applying an asymmetric cryptographic algorithm to the data (or an intermediate value derived from the data). The collection may also include information to assist in authenticating the entity that signed the data.

Entity -- A subject, object, user or another IT device, which interacts with TOE objects, data, or resources.

Expired Certificate -- A certificate with the not after component of its validity field having a value earlier than the current date. Certificates may or may not appear in CRLs issued after their expiration.

External IT entity -- Any trusted Information Technology (IT) product or system, outside of the TOE, which may, in accordance with the TSP, perform an operation.

Hash Algorithm -- An algorithm that maps variable length inputs into a fixed length output value known as the digest or hash. The algorithm is a many-to-one function; multiple inputs may result in the same output. However, discovering an input value that results in a desired or given output is computationally infeasible.

Identity -- A representation (e.g., a string) uniquely identifying an authorized user, which can either be the full or abbreviated name of that user or a pseudonym.

Integrity -- A security policy pertaining to the corruption of data and TSF mechanisms.

Key Pair -- A set of two keys used in asymmetric cryptography. A key generation algorithm creates the keys.

Named Object -- An object that exhibits all of the following characteristics:

- The object may be used to transfer information between subjects of differing user identities within the TSF.
- Subjects in the TOE must be able to request a specific instance of the object.
- The name used to refer to a specific instance of the object must exist in a context that potentially allows subjects with different user identities to request the same instance of the object.

Non-repudiation -- The inability to deny performing an action. Non-repudiation is evidence of the identity of the signer of a message and message integrity, sufficient to prevent a party from successfully denying the origin, submission, or delivery of a message and the integrity of its contents.

Non-Repudiation -- A security policy pertaining to providing one or more of the following:

- To the sender of data, proof of delivery to the intended recipient,
- To the recipient of data, proof of the identity of the user who sent the data.

Object -- An entity within the TSC that contains or receives information and upon which subjects perform operations.

Operating Environment -- The total environment in which a TOE operates. It includes the physical facility and any physical, procedural, administrative and personnel controls.

Operating System (OS) -- An entity within the TSC that causes operations to be performed. Subjects can come in two forms: trusted and untrusted. Trusted subjects are exempt from part or all of the TOE security policies. Untrusted subjects are bound by all TOE security policies.

Path Processing -- The means employed by a relying party to ensure that the certificates in a path leading from a relying party trust point to subscriber's public key

certificate, are all valid. The validation activity includes chaining the subscriber and issuer names, using the subject public key from the parent certificate to verify the signature on a certificate, applying constraints imposed by the various extensions in the certificate, verifying that none of the certificates have expired or been revoked, and other X.509 certification path validation rules.

Private Key -- A number, known only to the particular entity, its owner (i.e., the owner keeps the key secret). Owners use private keys to compute signatures on data they send and to decrypt information sent to them.

Public Key Certificate -- A digitally signed statement from one entity, the Certification Authority, binding the public key (and some other information) and the identity of the owner of the corresponding private key. The owner may be an individual, a system or device, an organization, or function.

Public Key Infrastructure -- The resources (people, systems, processes, and procedures) that provide services to register and identify new certificate owners, retrieve certificates, and determine the current validity of certificates.

Relying Party -- An entity or an organization that depends on a certificate (i.e., uses the public key in the certificate for digital signature and/or encryption) and its association of the subscriber's identity (i.e., subject name) and public key.

Revoked Certificate -- A certificate that relying parties should not trust or use. The CA that issued the certificate (or some similar authority) may revoke the certificate when conditions warrant. Conditions that may warrant revocation include suspected or actual compromise of the key or departure of the subscriber from the organization. CRLs issued by the CA always include all revoked, unexpired certificates (see Expired Certificate). Optionally, the CA may include revoked, expired certificates.

Robustness -- A characterization of the strength of a security function, mechanism, service or solution, and the assurance (or confidence) that it is implemented and functioning correctly.

Root Certificate -- The certificate at the top of the certification authority hierarchy. The certificate is self-signed; that is, the certificate issuer and the subject are the same entity, the Root CA. The certificate is generally a trust point. Since self-signed certificates do not have any trust in them, the root certificate or any other self-signed certificate must be distributed using secure means.

Secure State -- Condition in which all TOE security policies are enforced.

Security attributes -- TSF data associated with subjects, objects, and users that are used for the enforcement of the TSP.

Signature Verification -- The process of verifying a signature that includes the following steps: 1. Certification path validation in order to establish trust in the signer public key, 2. Calculating the hash for the message to be verified, and 3. Using applicable cryptographic algorithm with the signer public key (from step 1), calculated hash (from step 2), and signature to determine if the signature is valid.

Subject -- An entity within the TSC that causes operations to be performed.

Subscriber -- The entity (e.g., an individual) that has possession of the private key corresponding to the public key in a certificate. The certificate's subject field names the subscriber.

Threat -- Capabilities, intentions and attack methods of adversaries, or any circumstance or event, with the potential to violate the TOE security policy.

Threat Agent - Any human user or Information Technology (IT) product or system, which may attempt to violate the TSP and perform an unauthorized operation with the TOE.

Trust anchor -- A public key that a relying party directly trusts. A trust anchor can be in the form of a self-signed certificate. The self-signed certificate may belong to either a CA or an end-entity. The trust anchor is trusted because the relying party obtained the public key by reliable means outside of the PKI and believes that the trust anchor information (i.e., subject DN, public key, public key algorithms, and public key parameters (if applicable) are accurate. If the trust anchor is a CA, the relying party trusts any certificates the CA issues. This trust is transitive to the extent the X.509 certificate extensions permit; if the CA issues a certificate to another CA, the relying party also trusts the second CA if the X.509 path validation logic succeeds.

Trusted Third Party (TTP) -- An entity that other entities believe reliable, trustworthy and beyond reproach for purposes of performing some service. The TTP generally has no bias and is neutral for purposes of performing the service.

User -- Any entity (human user or external IT entity) outside the TOE that interacts with the TOE.

Vulnerability -- A weakness that can be exploited to violate the TOE security policy.

8.3. Acronyms

CA Certification Authority

CC Common Criteria

CPV Certification Path Validation

CRL Certificate Revocation List

DH Diffie Hellman

DN Distinguished Name

DoD Department of Defense

EAL Evaluation Assurance Level

ECDH Elliptic Curve Diffie Hellman

FIPS Federal Information Processing Standard

HMAC Hash based Message Authentication Code

IEC International Electrotechnical Committee

IETF Internet Engineering Task Force

ISO International Organisation for Standards

IP Internet Protocol

IT Information Technology

NTP Network Time Protocol

OCSP On-line Certificate Status Protocol

OS Operating System

OSP Organizational Security Policies

PKCS Public Key Cryptography Standard

PKE Public Key Enabled

PKI Public Key Infrastructure

PKIX Public Key Infrastructure Working Group -- IETF

PP Protection Profile

RFC Request for Comment

RSA Rivest, Shamir, and Adelman

SAML Security Assertion Markup Language

SSL Secure Socket Layer

ST Security Target

TLS Transport Layer Security

TOE Target of Evaluation

TOI Time Of Interest

TSC TSF Scope of Control

TSA Time Stamping Authority

TSF TOE Security Function

TSP Time-Stamp Protocol

TTP Trusted Third Party

APPENDIX A

Permissions of the privileged users

	First System Administrator	System Administrator	First Security Officer	Security Officer	First System Auditor	System Auditor
Password of the First System Administrator	Modify, Consult					
Condition of the First Security Officer/First System Auditor/System Administrator	Grant, Remove, Consult				Consult	Consult
Condition of the Security Officer			Grant, Remove, Consult		Consult	Consult
Condition of the System Auditor					Grant, Remove, Consult	Consult
Information about the session	Consult	Consult	Consult	Consult	Consult	Consult

	First System Administrator	System Administrator	First Security Officer	Security Officer	First System Auditor	System Auditor
Users, Applications and Web Services			Register, Modify data, Remove, Consult	Register, Modify data, Remove, Consult	Consult	Consult
Groups of Users and Applications, Web Services Groups, Mixed Groups, Organizational groups, Dynamic Groups, Groups of Groups			Create, Remove, Modify data, Add and remove members, Consult	Create, Remove, Modify data, Add and remove members, Consult	Consult	Consult
Templates for dynamic groups			Create, Remove, Modify, Consult	Create, Remove, Modify, Consult	Consult	Consult
Administration of trusted entities: CAs, VAs, TSAs, SSAs			Register, Modify data, Remove, Consult	Register, Modify data, Remove, Consult	Consult	Consult

	First System Administrator	System Administrator	First Security Officer	Security Officer	First System Auditor	System Auditor
Groups of CAs and SSAs			Create, Remove, Modify data, Add and remove members, Consult	Create, Remove, Modify data, Add and remove members, Consult	Consult	Consult
Security Policy			Establish, Modify, Setting (“freezing”)	Establish, Modify, Setting (“freezing”)	Consult	Consult
Mapping of attributes, Identification Policies			Create, Remove, Modify data, Consult	Create, Remove, Modify data, Consult	Consult	Consult
Filter of certificates			Modify, Consult	Modify, Consult	Consult	Consult
Authentication mechanisms			Register, Remove, Consult	Register, Remove, Consult	Consult	Consult
Groups of authentication mechanisms			Create, Remove, Modify data, Add and remove member, Consult	Create, Remove, Modify data, Add and remove members, Consult	Consult	Consult

	First System Administrator	System Administrator	First Security Officer	Security Officer	First System Auditor	System Auditor
Authentication Agents, Authentication Rules, Authentication Policies, Protocols, Resources, Authorization rules, Authorization policies, Accounting policies			Register, Modify data, Remove, Consult	Register, Modify data, Remove, Consult	Consult	Consult
Groups of resources			Create, Remove, Modify data, Add and remove members, Consult	Create, Remove, Modify data, Add and remove members, Consult	Consult	Consult

	First System Administrator	System Administrator	First Security Officer	Security Officer	First System Auditor	System Auditor
Algorithms, Commitments of the signature, Roles of the signer, Style templates defined by the user, Extern signature Policies, Rules and Policies for validating certificates, Rules and Policies for generating signatures, Rules and Policies for validating signatures			Register, Modify data, Remove, Consult	Register, Modify data, Remove, Consult	Consult	Consult
Groups of equivalent policies (for generation of signatures, for verification of signatures, and for validation of certificates)			Create, Remove, Modify data, Add and remove members, Consult	Create, Remove, Modify data, Add and remove members, Consult	Consult	Consult

	First System Administrator	System Administrator	First Security Officer	Security Officer	First System Auditor	System Auditor
Symmetric and Asymmetric algorithms, Security labels, Encryption and Decryption rules and policies			Create, Modify data, Remove, Consult	Create, Modify data, Remove, Consult	Consult	Consult
Policies of key management			Create, Modify data, Remove, Consult	Create, Modify data, Remove, Consult	Consult	Consult
Groups of equivalent policies (for signature custody)			Create, Remove, Modify data, Add and remove members, Consult	Create, Remove, Modify data, Add and remove members, Consult	Consult	Consult
Policies of signature custody			Create, Modify data, Remove, Consult	Create, Modify data, Remove, Consult	Consult	Consult
Policies of Management of symmetric keys			Create, Modify data, Remove, Consult	Create, Modify data, Remove, Consult	Consult	Consult

	First System Administrator	System Administrator	First Security Officer	Security Officer	First System Auditor	System Auditor
Services configuration	Modify, Consult, Reload	Modify, Consult, Reload			Consult	Consult
Repositories, EP Schema, HSM Devices	Create, Modify data, Remove, Consult	Create, Modify data, Remove, Consult			Consult	Consult
Backups	Create, Import, Remove				Consult	Consult
Logs	Consult	Consult	Consult	Consult	Consult	Consult
Statistics	Consult, Reload	Consult, Reload			Consult	Consult

Table 8-1. Table of permissions of the privileged users

In an CC EAL4+ configuration, the following privileges must not be assigned to newly-created groups:

- Privileges for accessing xpath resources.
- Privileges for accessing the Administration Console resource.

New groups can, however, be given privileges for accessing Web service resources.


SAFELAYER

SAFELAYER SECURE COMMUNICATIONS, S.A.

Edificio Valrealty C/ Basauri, 17 Edifico B Pl. Baja Izq. Of. B 28023 Madrid (SPAIN) Tel.: +34 91 7080480 Fax: +34 91 3076652
Edif. World Trade Center (S-4), Moll de Barcelona S/N 08039 Barcelona (SPAIN) Tel.: +34 93 5088090 Fax: +34 93 5088091

WWW.SAFELAYER.COM