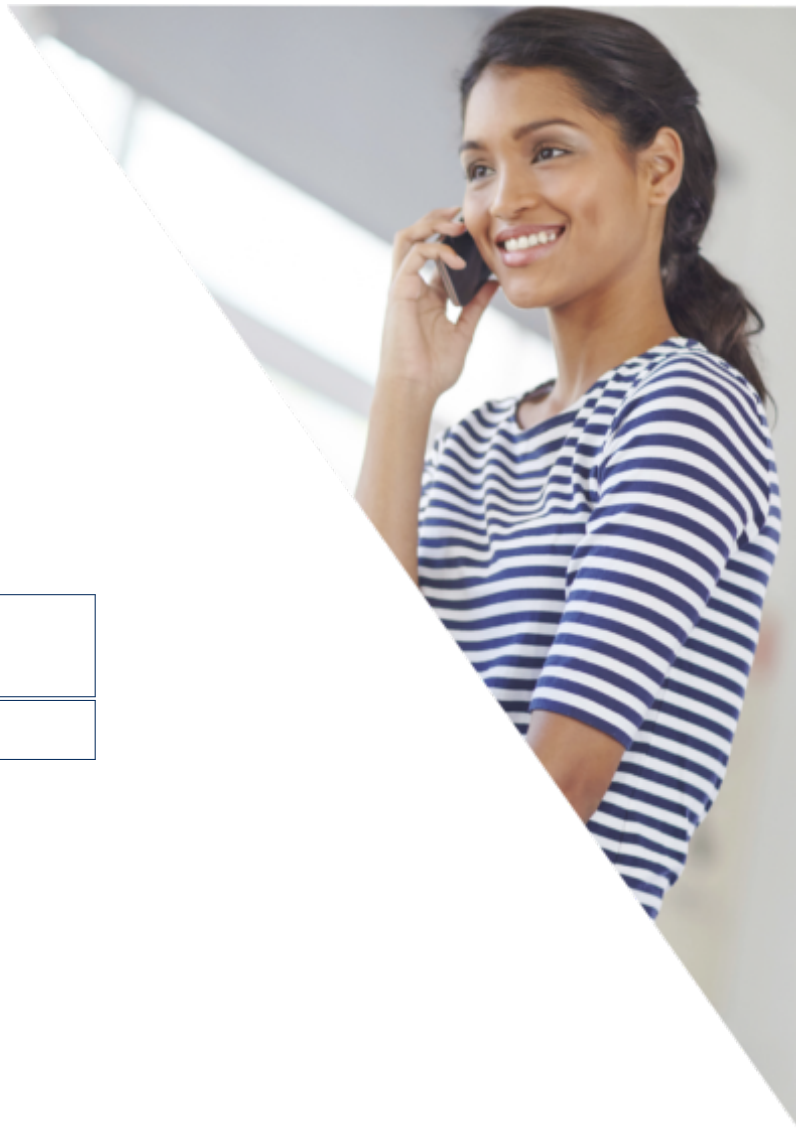
ERCOM



CRYPTOBOX V2.1

SECURITY TARGET

REFERENCE		CBOX_ASE
VERSION		2aa090b
DATE		2018-04-25
AUTHOR		Julien Kowalski

ERCOM ▲ 6, Rue Dewoitine - Bâtiment Rubis
78140 Vélizy-Villacoublay ▲ France
T: +33 1 39 46 50 50 ▲ F: +33 1 39 46 25 25 ▲ www.ercom.com

Contents

References

[ANS] ANSSI. Référentiel général de sécurité - Processus de qualification d'un produit de sécurité - niveau standard - version 1.2. Technical report, ANSSI.

[ANS12] ANSSI. RGS - annexe B2. Gestion des clés cryptographiques : Règles et recommandations concernant la gestion des clés utilisées dans des mécanismes cryptographiques. Technical report, ANSSI, Juin 2012.

[ANS14] ANSSI. RGS - annexe B1. Mécanismes cryptographiques : Règles et recommandations concernant le choix et le dimensionnement des mécanismes cryptographiques. Technical report, ANSSI, Février 2014.

[CC12a] CC. CCMB-2012-09-001 - Version 3.1 Revision 4 - common criteria for information technology security evaluation, part 1: Introduction and general model. Technical report, Common Criteria, September 2012.

[CC12b] CC. CCMB-2012-09-002 - Version 3.1 Revision 4 - common criteria for information technology security evaluation, part 2: Security functional requirements. Technical report, Common Criteria, September 2012.

[CC12c] CC. CCMB-2012-09-003 - Version 3.1 Revision 4 - common criteria for information technology security evaluation, part 3: Security assurance requirements. Technical report, Common Criteria, September 2012.

[CKS09] David Cash, Eike Kiltz, and Victor Shoup. The twin diffie-hellman problem and applications. *J. Cryptology*, 22(4):470–504, 2009.

[Kra03] Hugo Krawczyk. SIGMA: the 'sign-and-mac' approach to authenticated diffie-hellman and its use in the ike-protocols. In *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*, pages 400–425, 2003.

[NIoSaT01a] National Institute of Standards and Technology. Advanced Encryption Standard. *NIST FIPS PUB 197*, 2001.

[NIoSaT01b] National Institute of Standards and Technology. Recommendation for block cipher modes of operation. *NIST FIPS PUB 800-38A*, December 2001.

[NIoSaT07] National Institute of Standards and Technology. Recommendation for block cipher modes of operation:galois/counter mode (gcm) and gmac. *NIST FIPS PUB 800-38D*, November 2007.

[NIoSaT08] National Institute of Standards and Technology. The Keyed-Hash Message Authentication Code (HMAC). *NIST FIPS PUB 198-1*, July 2008.

[NIoSaT10] National Institute of Standards and Technology. Recommendation for password-based key derivation. *NIST FIPS PUB 800-132*, December 2010.

[NIoSaT13] National Institute of Standards and Technology. Digital signature standard (SHS). *NIST FIPS PUB 198-1*, July 2013.

[NIoSaT15a] National Institute of Standards and Technology. Recommendation for random number generation using deterministic random bit generators. *NIST FIPS PUB 800-90A Revision 1*, June 2015.

[NIoSaT15b] National Institute of Standards and Technology. Secure Hash Standard (SHS). *NIST FIPS PUB 198-1*, August 2015.

[Sho04] Victor Shoup. Fcd 18033-2 encryption algorithms — part 2: Asymmetric ciphers. Technical report, ISO, December 2004.

[ZJC11] P. Zimmermann, A. Johnston, and J. Callas. ZRTP: Media Path Key Agreement for Unicast Secure RTP. RFC 6189 (Informational), April 2011.

1 – Introduction

1.1 Security target and TOE identification

This document constitutes the Security Target (ST) of the Cryptobox product developed by Ercom.

- **ST name:** Cryptobox v2.1
- **ST version:** 2aa090b
- **ST Date :** 2018-04-25
- **TOE identifier:** Cryptobox
- **TOE version:** v2.1.48
- **TOE Developer:** Ercom

TOE version details

The TOE is :

- Cryptobox v2.1.48 on an Ubuntu 16.04.3 (64-bit) LTS (Xenial Xerus) server,
- postgreSQL 9.5.4 on an Ubuntu 16.04.3 (64-bit) LTS (Xenial Xerus) server,
- Web client on Windows 7 (x64) and Firefox ESR 52.6.0
- and Android client on Android 6.0 (Marshmallow) .

These versions are set at this security target release date and shall be kept up to date by TOE administrator.

1.2 Conformance claims

This Security Target claims conformance to the Common Criteria version 3.1 reference 4 with the referenced documents [CC12a][CC12b][CC12c].

Conformance is claimed as follows:

- Part 1: conformant

- Part 2: conformant
- Part 3: conformant to assurance level **EAL3** augmented with **AVA_VAN.3** and **ALC_FLR.3**.

Conformity to a protection profile: This Security Target does not claim conformance with any protection profile.

1.3 Conventions & Terminology

Term	Definition
UA	User Agent(client): the native or Web Application which interfaces the end user.
CSS	Cloud Security Server(server): main hub to interact with UA applications and to link them together.
SM	Security Module(server): sub component of the server which handles all the server-side cryptography
CSP	Cloud Service Provider: storage service for the encrypted data
EUS	The End-User Storage is a special block of encrypted data containing user keys
TOE	Target of Evaluation
OSP	Organizational Security Policies

Table 1.1: Glossary and acronyms

2 – Description

2.1 Cryptobox by Ercom

2.1.1 Context

Cloud and virtualisation technologies mesh really well with current business needs such as costs reduction, mobility, scalability or flexibility. However one of the major drawbacks is the need to keep the control of the data. Offloading critical pieces of intellectual property can be dangerous to a company, especially in the current context of rampant industrial espionage, cyber criminality and governmental monitoring.

Cryptobox solves this dilemma and allows customers to get the advantages of cloud-based data storage without any risk to their security or privacy. Our collaborative work platform ensures that sensitive data stay under the exclusive control of its rightful owner, protecting it against interception, theft and unauthorized access.

2.1.2 Cryptobox

Cryptobox is a collaborative work platform that is fully secure and leverages the advantages of the cloud.

It enables companies to:

- Create work groups, also called "Workspaces" in this document.
- Send, share, exchange files inside workspaces while guaranteeing their confidentiality and integrity and authenticity.
- Track file changes, comments on file.
- Store everything on the cloud for ease of access, lower costs and reliability.
- Use any device they want.

Cryptobox introduces the paradigm of "security plans" which can be added as different security layers to get in depth security. Cryptobox enforces:

1. the "Enterprise plan", which allows the enterprise to keep control of its data and consists in user authentication and access control enforcement based on this authentication,
2. the "User plan", where the trust is only enabled in a peer-to-peer mode, which allows end-to-end security.

2.1.3 HIGH LEVEL OVERVIEW

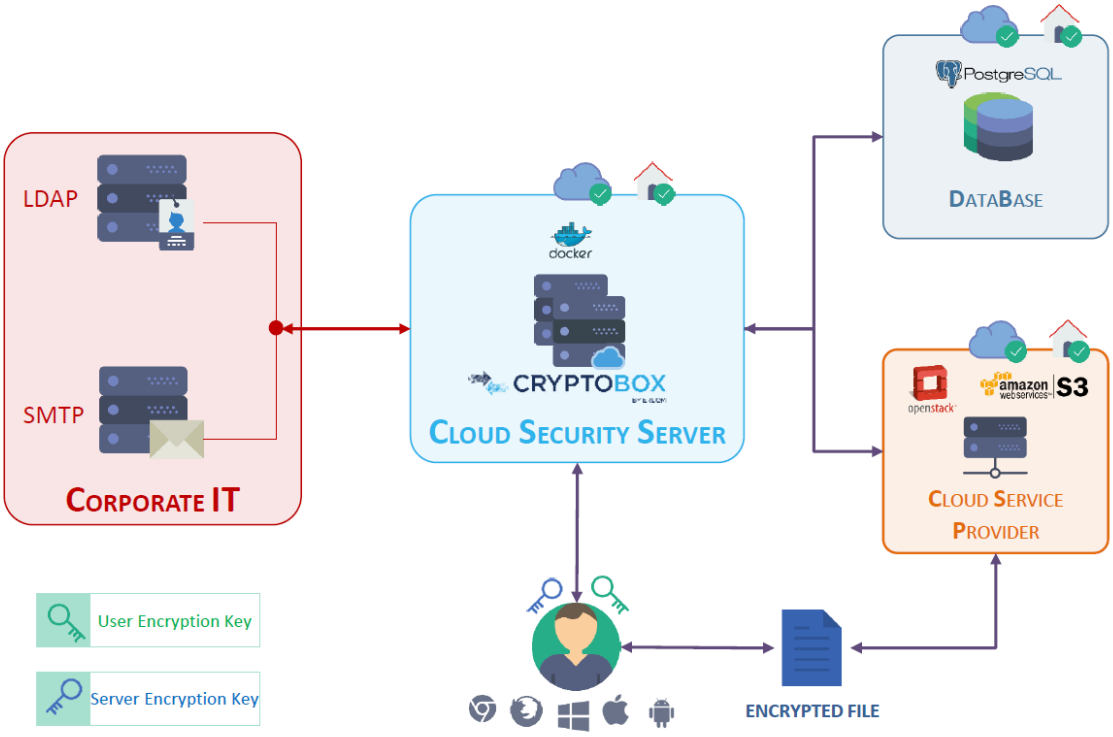


Figure 2.1: Cryptobox high level overview

Figure 2.1 illustrates how our platform secures data stored in the cloud. We can identify three major components: the client devices, the cloud security server (CSS) and the cloud service provider (CSP):

1. CRYPTOBOX users need to access, store, modify and share their files in a secure fashion. They can run either one of the CRYPTOBOX applications (web, Android, IOS or Windows).
2. The CSP stores data and makes it available to client devices. The CSP is not a trusted component. Therefore every data stored on the CSP is encrypted, in particular files content and their metadata. The CSP can be hosted on the company premises (on internal physical storage servers) or hosted on a cloud service (supporting OpenStack Swift or Amazon S3).
3. The CSS is responsible for authentication of the client devices, handling access controls, storing (parts of) encryption keys in the Database, interconnecting the storage(CSP) and the Database and the clients and maintaining audit logs.

The basic principle(see 2.3 more details) of the solution is:

- to generate the cryptographic keys on the client-side,
- to encrypt files on the UAs,
- to store those files in the cloud(CSP),
- to then cut the encryption keys in two keys,
- to share the first part of the keys between all the authorized users (members of a work group),
- and to give the other part to the CSS, which stores them encrypted in the Database and will only give it out to authorized users when they need it.

By storing all files in an encrypted fashion, CRYPTOBOX is capable of monitoring and logging all accesses to any file protected by the system.

2.2 TOE Components

Cryptobox components

The different parts of Cryptobox are (cf. Figure 2.1):

- The server composed by:
  - The CSS, Cloud Security Server;
  - The SM, Security Module.
  - The Database
- The different UAs(User Agents):
  - The Web client;
  - The Android client;
  - The Windows client;
  - The iOS client.

CSS

The Cloud Security server is the main hub to interact with the UAs and to link them together.
It handles and protects the internal databases.
It also takes care of the notifications, the billing and the audit trail.
The CSS is a software component being accessed by UAs through a web server.

SM

The SM is the cryptography core of Cryptobox server, thus the root of trust for ”Enterprise plan”. It handles the authentication of the clients, the access controls, the protection of (parts of) the encryption keys.
The main role of the SM component is to handle all the server-side cryptography. The users authenticate with it and securely receive their (super-encrypted) EUS.
The SM is a software library delivered with the CSS. Its interfaces are handled by the CSS.

Database

The database is used by the Cryptobox system in order to store all server internal data. Sensitive data are encrypted in this database.
The database contains (encrypted data are marked ”ENC”):

- User information:
  - email address (ENC)
  - signing certificate (ENC)
  - encryption certificate (ENC)
  - Salt for password derivation
  - Password authentication data (ENC)
  - EUS (ENC)
- Session state (both for password or certificate authentication):
  - related user (ENC)
  - session initiation and creation date
  - session id (ENC)
  - session information (ENC): user email, source IP and encryption context
- Workspace data:

- workspace management events (SIGNED)
- workspace membership (for access control):
  - user ID (ENC)
  - user role
  - encrypted ”space client key” (ENC)

- File or directory information:
  - identifier
  - encrypted ”space server key” (ENC) (files only)
  - parent identifier (used for directory structure)
  - metadata (ENC)
  - author signature (for top level directory) (cf. subsection 2.3.5)
- User notifications
- Audit logs:
  - log date
  - log data (ENC)
  - related space id or file id (if relevant)

UA

We call User Agent the native or Web Application which interfaces the end user.

It relies on a library common to all UAs for secure operations and interactions with the CRYPTOBOX server. It handles the following secure operations:

- User authentication.
- Workspace administration.
- File management(encryption, integrity, version control) on user plan.

TOE PERIMETER

The perimeter of the TOE is represented on figure 2.2 and comprises:

1. the CSS/SM evaluated on Ubuntu 16.04.3 (64-bit) LTS (Xenial Xerus) , with docker version 1.12.1 ;
2. the database evaluated on Ubuntu 16.04.3 (64-bit) LTS (Xenial Xerus)
3. the Android UA evaluated on Android 6.0 (Marshmallow) ;
4. and the Web UA evaluated on Firefox ESR 52.6.0and Windows 7 (x64).

The CRYPTOBOX server is delivered as a docker container that also comprises an NGINX instance with the configuration for an HTTPS connexion. This configuration is thus also part of the TOE.

DEPENDENCIES AND TOE ENVIRONMENT

The CRYPTOBOX server relies on the following dependencies, included in the server docker image:

- Ubuntu version 16.04.3 (64-bit) LTS (Xenial Xerus) with Linux kernel version 4.4.0-112-generic
- libpq5 version 9.5.11
- nginx version 1.10.3
- python version 3.5.1-3 and python dependences:
  - aiohttp version 1.2.0

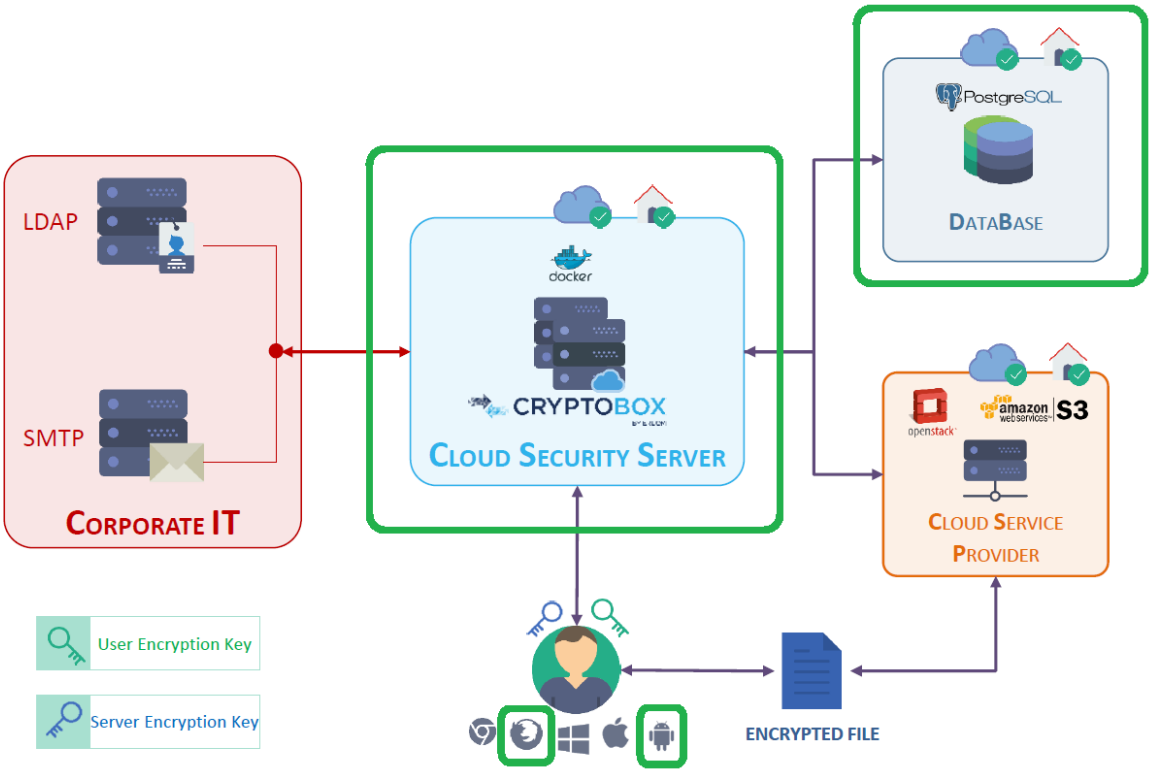


Figure 2.2: TOE perimeter (The components included in the TOE are framed in green.)

- ldap3 version 2.1.1
- python-dateutil version 2.6.0
- python-keystoneclient version 3.8.0
- voluptuous version 0.9.3
- aiopg version 0.13.0
- python-swiftclient version 3.0.0
- botocore version 1.4.87

The Ubuntu version and libraries version included in the server docker image are fixed for a given CRYPTOBOX version. These dependencies are updated with latest available version in Ubuntu repository at each CRYPTOBOX release.

In order to run the CRYPTOBOX server docker image requires:

- Any linux system supporting docker (eg. Ubuntu, Debian, etc.).
- docker version $\geq$ 1.12.1

The database server runs postgreSQL version 9.5.4 installed on Ubuntu 16.04.3 (64-bit) LTS (Xenial Xerus) ; and is part of the TOE.

The CRYPTOBOX clients relies for security functions on a common library based on Golang version 1.7. The web client furthermore relies on the Web Crypto API (W3C Proposed Recommendation 15 December 2016).

CRYPTOBOX relies on the following external services to provide availability:

- a Mail server,
- a CSP : OpenStack Swift 2.10.0
- *optionally* a LDAP server.

The Operating System of the server running the CRYPTOBOX server docker image; the Mail server, CSP and LDAP server, the web browser and the dependencies listed here are not part of the TOE but shall be kept up to date (i.e. with latest security patches applied) by their respective administrators.

2.3 Security functionalities

CRYPTOBOX allows users to work collaboratively in workspaces.

Each file is linked to (owned by) one and only one workspace.

A user can assume different roles in those workspaces:

- non-member,
- member, which can be refined to:
  - reader, with a read-only access to the documents
  - writer, with read and write access to the documents,
  - owner, with read and write access to the document and workspace member management.

A user can also assume the following roles in CRYPTOBOX :

- Administrator: a user listed in the server configuration file with user management and CRYPTOBOX monitoring capabilities,
- Trustee(2.3.7): a user required in account recovery.

2.3.1 Security Plans

CRYPTOBOX introduces the concept of security plan:

1. *User security plan*
2. *Enterprise security plan*

Those plans are orthogonal, therefore enforce security in CRYPTOBOX by providing two independent layers of protection.

The User plan guarantees end-to-end encryption between users. This is achieved by generating all cryptographic keys related to the user protection on the client side and by using signature for integrity. The main principle is to furthermore protect those encryption keys by a server key and a client key, both also generated client side. The CSS/SM handles(see **O::CSS.SM.SUPERENCRYPTION**) its keys received from the client, while the clients can encrypt their keys, for themselves or other users, thanks to encryption and signing certificates, and send it to the CSS/SM for storage in the Database. Hence, the users have full control of their documents and they can choose who has access to it. The root of trust for the user is his passphrase, used for authentication with the CSS/SM(Cloud Security Server/Security Module) and for EUS(End-User Storage) encryption.

The Enterprise plan guarantees access control on the user's data by handling the server keys. This allows the Enterprise to record access logs for audits and to deny access to documents, in case a user account is stolen, by not delivering the server key. Therefore document decryption by the client is impossible.

Those plans are emphasized in the following sections.

2.3.2 Workspaces

A group of users can work collaboratively in workspaces. Each file is linked to one and only one workspace and each workspace has at least one owner.

Workspace owners are users with the ability to manage workspace members and their respective rights. An owner invites users to the workspaces as follows:

- the owner enters the user(s) email address(es),
- he or she can check the user(s) certificates,
- if an invited user is not registered yet, he or she receives an invitation to join CRYPTOBOX first,
- if a user is registered, he or she is added to the group and receive a notification.

The list of users is part of the workspace definition. This workspace definition is cryptographically signed by a workspace owner first, then by the SM therefore ensuring its authenticity on the user plan and enterprise plan.

2.3.3 User authentication

After registration, a user gets authenticated to the SM by using his email address and a passphrase. A symmetric key is securely derived from this passphrase to decrypt his End User Storage(EUS). The EUS URL at the CSP(Cloud Service Provider) is sent by the CSS/SM at the end of the authentication procedure.

The EUS contains:

- a signing certificate and the corresponding private key,
- an encryption certificate and the corresponding private key,
- a list of the workspaces the user belongs to and their owners
- a list of the user's trustees(2.3.7).

The user can then perform a second authentication step by performing an authenticated key exchange with the CSS/SM, using his signing pair of key.

The user is granted access to his workspaces and files once he is logged in.

Note: The administrators can enforce in-depth security by adding a strong passphrase policy and email white-list (for company email) policy during the user registration.

2.3.4 File access control

Cryptobox implements the *two-man rule* to protect the data and metadata in the cloud: a user needs an authorized account **and** the server access key.

This two man rule allows cryptographically enforcing respectively the user and the enterprise plan access control:

- A workspace owner shall have added the user to the workspace and therefore have shared (encrypted with his or her public encryption key and stored by the CSS/SM) the user plan key with him or her,
- The user shall be properly authenticated to the CSS/SM and be present in the workspace definition for being granted the enterprise plan access. The user can then decrypt the file. The enterprise plan access control is therefore enforced for each file access.

This allows us to deny access to the documents when a terminal / user account is stolen or the server(CSS/SM) compromised. User plan revocation uses "lazy" revocation: a new workspace key is generated for encryption of the new contents. We can also record access logs for audits.

The file access works as follows(figure 2.3):

1. The user is already authenticated, thus has access to his or her EUS and shares a symmetric encryption key with the server). He or she requests an access to a file in a workspace he or she is a member of.
2. The CSS/SM checks that the user is authenticated and has the right to access the workspace. It then sends the server content key of the file requested, using the authentication shared key, to the user. The user also receives the user content key that was encrypted using his or her public encryption key by a workspace owner.
3. The user decrypts the server content key of the file requested, using the authentication shared key.
4. The user decrypts the user content key, using his or her private encryption key from the EUS.
5. The user can then combine the user and the server content keys to retrieve the content key that encrypts the file.
6. The user can decrypt the file data and metadata retrieved from the CSP.

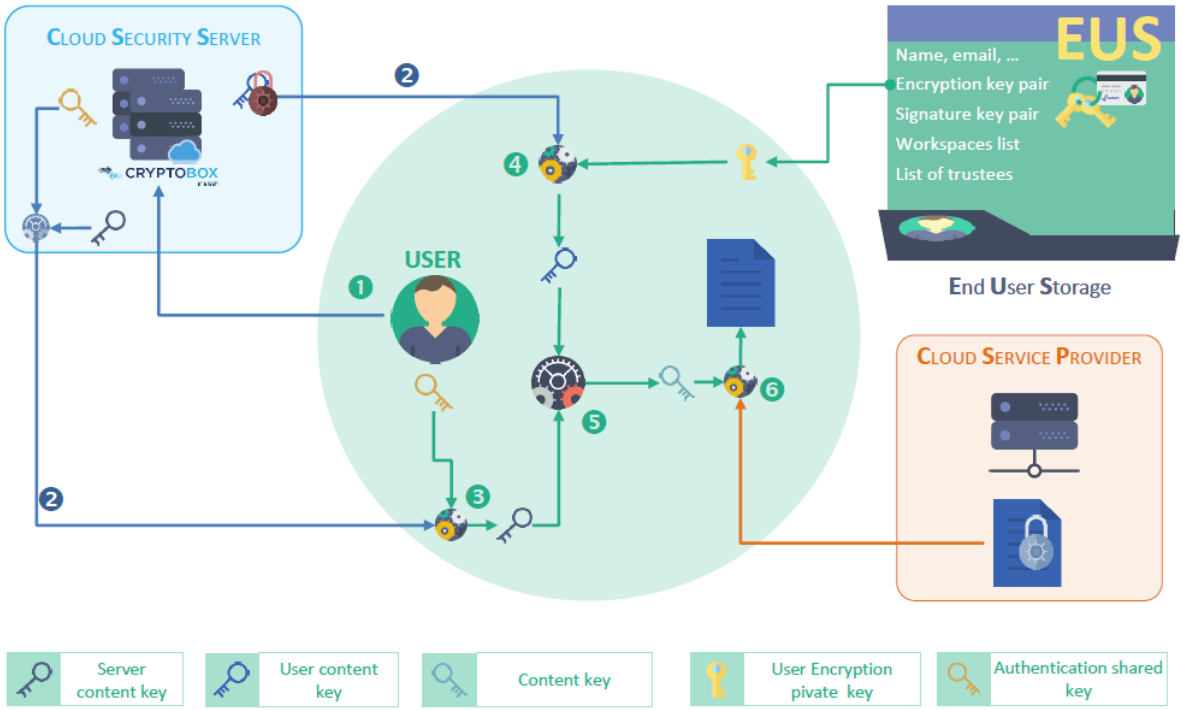


Figure 2.3: File access control

2.3.5 File creation and file update

Cryptobox performs the following actions for storing or updating files:

1. Files are split into chunks which are then encrypted before being uploaded, which allows saving bandwidth in case of a local file modification,
2. Those chunks are send to the CSS, who store them in the CSP,
3. The encryption key, generated for each file addition or update, is then split into two parts, one for the user plan, the other one for the enterprise plan
4. The encrypted file and keys are then added to the *File Revision*, a Merkle tree representation of the file directory tree of the workspace :
    a. The root hash of *File Revision* is signed by the user ensuring accountability for the writing and sent to the CSS/SM.
    b. The *File Revision* of the workspace and its signature are verified by the CSS/SM for checking correctness of the operation and enforcing access control rules.
    c. The *File Revision* of the workspace content (with writer's signature) is then signed by the CSS/SM and updated in its Database allowing other workspace members to get the new files.

Workspace content signature also allows other members to verify the validity of data sent by the CSS/SM in a manner independent from the enterprise plan.

2.3.6 URL File sharing

The users can share the files with non-members of the workspace.

To do so, they split the file encryption key into two newly generated keys, a client sharing symmetric key and a server sharing symmetric key, in order to allow previously defined *two-man rule* to be enforced.

The user can set an expiration date for the link.

Finally the user can either copy and paste the sharing link with the client sharing key in it or let Cryptobox send an email without it and provide that key in an out-of-band manner.

The sharing link also contains an access token that the CSS/SM will use to authenticated the guest user and send him or her the sever sharing key.

2.3.7 Account recovery

In CryptoBox , the users have full control of their keys. As such, account recovery cannot be implemented in a traditional way: the server(CSS/SM) shall not have access to user's secrets.

If a user wants to recover his data, it requires the collaboration of another user, the *trustee*. He can name one or more *trustees*.

The user's EUS key is backed up in this way: the authenticated(step 1) user splits (step 2) his EUS key into two halves using a secret sharing scheme and encrypts (step 3) them respectively for the trustee and for the SM. The encrypted shares are finally stored by the CSS(step 4).

This allows us to request an administrator validation before recovery. The user can then securely retrieve his key from the trustee and the SM through a secure CryptoBox connection and a back channel.

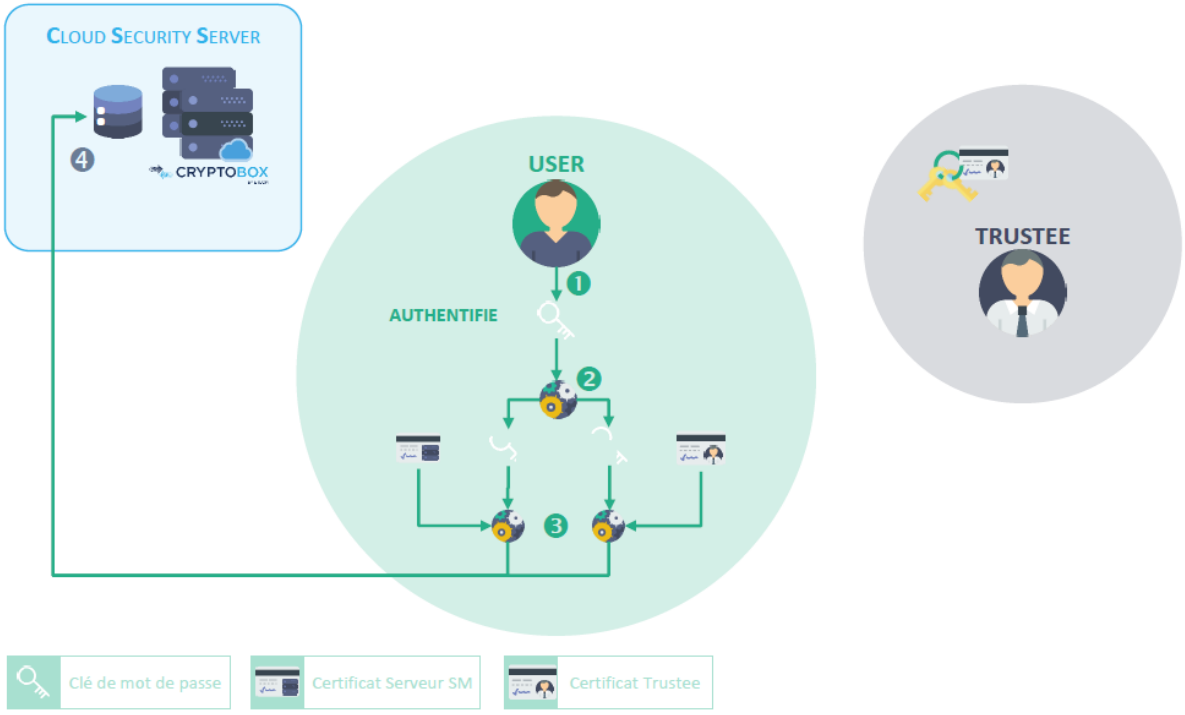


Figure 2.4: Trustee choice procedure

3 – Security problem definition

3.1 Assets

Assets for the TOE are divided into three categories:

- Primary assets which are user data from IT environment which shall be protected using the security functions offered by the TOE. This intended protection is addressed by organizational security policies;
- Secondary assets which are needed by the TOE to offer its security services. We divide secondary assets into:
  - User assets: these are data created by or for the user.
  - TSF assets: these are data created by the TOE or for the TOE internal usage.

Notation The following notation are used to specify required protection on assets:

- C: Confidentiality
- I: Integrity
- A: Authenticity
- A2: Availability

3.1.1 Primary Assets

D::DOC.DATA *(Requires C/I/A protection)*
This asset represents the files stored and protected by the TOE.

D::DOC.METADATA *(Requires C/I/A protection)*
This asset represents the metadata of the files stored and protected by the TOE (D::DOC.DATA). They include the documents names and their full path in the workspace.

D::WORKSPACE.DEFINITION *(Requires C/I/A protection)*
This asset represents the definition of the workspace:

- Workspace members with their respective access rights,
- (Encrypted) Workspace keys which are used by members for accessing workspace content,
- the file revisions of the workspace,
- the tags of the workspace.

3.1.2 Secondary Assets

3.1.2.1 User Assets

D::USER.PASSPHRASE *(Requires C/I/A protection)*
This represents the user passphrase which is used for authenticating to the server and decrypting the user's EUS.

D::USER.EUS *(Requires C/I/A protection)*
This represents the Encrypted User Storage, stored encrypted on the CSP. It is encrypted by a key securely derived from the user passphrase, following state of the art passphrase protection and key derivation. The user's EUS contains the following information:

- The user given name, surname and email.
- The user signature certificate and private key.
- The user encryption certificate and private key.
- the list of the user's trustees
- the SM certificate pins, for certificate pinning
- For each workspace the user belongs to:
  - The workspace identification.
  - The URL of the CSS providing access to this space.
  - List of the owners certificates for the last known workspace revision.
  - Number of the last known space revision.
- For each verified user (i.e. other users whom certificate has been explicitly verified by the user):
  - Signature certificate.
  - Information about the verification.

D::USER.KEY.EUS *(Requires C/I/A2 protection)*
This is the key needed for EUS encryption and that is recovered through the account recovery.

Application note: This asset can be derived from the user passphrase that protect D::USER.EUS but is defined as a specific asset as it is managed independently from the user passphrase when related to account recovery.

D::USERS.PUBLIC.CERTIFICATES *(Requires I/A protection)*
Users Public key certificates stored by the CSS/SM for the users' authentication. This asset also includes the revocation status of the certificate.

3.1.2.2 TSF Assets

D::USER.PLAN.KEYS *(Requires C/I/A protection)*
This asset represents all the cryptographic keys (symmetric or asymmetric) necessary for the operations in the CryptoBox system managed by the **user**. They enforce the user security plan.

D::ENTERPRISE.PLAN.KEYS *(Requires C/I/A protection)*
This asset represents all the cryptographic keys (symmetric or asymmetric) necessary for the operations in the CryptoBox system managed by the **server**. They enforce the enterprise security plan.

D::USER.AGENT *(Requires I protection)*
This asset represents the UAs that enable the user of CryptoBox to connect to the CSS/SM.

D::CONFIG.CSS.SM *(Requires C/I protection)*
This asset represents the configuration of the CSS/SM.

D::AUDIT.LOGS *(Requires C/I protection)*
This asset represents the CSS records logs for the audit trail stored in its database.

3.1.2.3 Security needs

Assets	Confidentiality	Integrity	Authenticity	Availability
D::DOC.DATA	YES	YES	YES	NO
D::DOC.METADATA	YES	YES	YES	NO
D::WORKSPACE.DEFINITION	YES	YES	YES	NO
D::USER.PASSPHRASE	YES	YES	YES	NO
D::USER.EUS	YES	YES	YES	NO
D::USER.KEY.EUS	YES	YES	NO	YES
D::USERS.PUBLIC.CERTIFICATES	NO	YES	YES	NO
D::USER.PLAN.KEYS	YES	YES	YES	NO
D::ENTERPRISE.PLAN.KEYS	YES	YES	YES	NO
D::USER.AGENT	NO	YES	NO	NO
D::CONFIG.CSS.SM	YES	YES	NO	NO
D::AUDIT.LOGS	YES	YES	NO	NO

Table 3.1: Security needs

3.2 Assumptions

A::ROLE.ADMIN CryptoBox and database administrators are not hostile and are competent persons with necessary resources for the implementation of their tasks. They are trained to perform the operations for which they are responsible and they follow manuals and administration procedures.

A::ROLE.USER Users are not careless or wilfully negligent and will follow the instructions provided by documentation to perform their roles(owner,member). Specifically, they will protect their passphrase.

A::TLS We shall assume that the browser or client OS implementation of TLS is correct and secure.

A::SERVER.SOFTWARE.ENVIRONMENT The operating systems of the CSS/SM and of the database provide adequate security, including domain separation and non-bypassability, to counter the perceived threats. The OS ensures applicative memory separation (no other applicative process can access the CSS/SM or database memory); and software libraries on which the CSS and the SM rely correctly provide the services they are intended for, including security services.

Moreover the operating system of the CSS/SM and of the database and software libraries on which the CSS and the SM rely are up to date with all security patches applied.

A::SERVER.PHYSICAL.ENVIRONMENT The server operating the CSS, the SM and the database server are not physically accessible by an attacker.

A::UA.OPERATING.ENVIRONMENT The system operating the UAs provides adequate security, including domain separation and non-bypassability, to counter the perceived threats. It ensures:

- Applicative memory separation (no other applicative process or browser window/tab can access the UA memory),
- Sensitive input protection (concerning passphrase entry for instance),
- For the Android UA, signature validation of the CRYPTOBOX application before installation and update.

The system operating is the Android Operating System in the case of the Android UA and the Web browser in the case of the Web UA and the operating system.

Moreover UA's operating system is up to date with all security patches applied.

3.3 THREATS

This section considers that there are two types of threatening agents:

- A **local attacker** who has physical access to a physical device containing a CRYPTOBOX client. This access can either be permanent or temporary. For instance a local attacker maybe a legitimate user.
- A **remote attacker** who is able to intercept and manipulate the data on the network (without physical access to any physical device containing a TOE UA) or at the cloud provider level.

3.3.1 THREATS ON PRIMARY ASSETS

Threats on primary assets are the main motivations for which CRYPTOBOX is deployed.

T::DOC.DATA.COMPROMISE An attacker has an illicit access to a document and is able to read or modify its content.

Assets covered D::DOC.DATA

T::DOC.METADATA.COMPROMISE An attacker gains an access to the metadata of a document and manages to read or modify them.

Assets covered D::DOC.METADATA

T::WORKSPACE.COMPROMISE An attacker reads or modifies a workspace definition. He can for instance elevate his credential from non-member to member or from member to owner.

Assets covered D::WORKSPACE.DEFINITION

3.3.2 THREATS ON USER ASSETS

T::PASSPHRASE.COMPROMISE An attacker is able to compromise the user passphrase. He manages to guess or replace that passphrase.

Assets covered D::USER.PASSPHRASE D::USER.PLAN.KEYS

T::PASSPHRASE.LOSS A user forgets his or her credentials and therefore loose the availability of D::USER.KEY.EUS.

Assets covered D::USER.KEY.EUS

T::EUS.COMPROMISE An attacker is able to compromise the EUS to read or modify its contents.

Assets covered D::USER.EUS

T::KEY.EUS.COMPROMISE This threat is considered for account recovery. An attacker gets an illicit access to one or the two halves of the key encrypting the EUS. He can either read or replace them.

Assets covered D::USER.KEY.EUS

T::USER.PUBLIC.CERTIFICATES.MODIFICATION An attacker modifies the users' public key certificates stored at the CSS/SM.

Assets covered D::USERS.PUBLIC.CERTIFICATES

T::UA.CSS.SM.COMMUNICATION.COMPROMISE An attacker eavesdrops the conversation between the client and the server. He can try in addition to modify the data transiting.

Assets covered D::USER.PASSPHRASE D::WORKSPACE.DEFINITION D::USERS.PUBLIC.CERTIFICATES D::USER.KEY.EUS D::USER.PLAN.KEYS D::ENTERPRISE.PLAN.KEYS D::AUDIT.LOGS

3.3.3 Threats on TSF assets

T::USER.ROLE.USURPATION An attacker elevates his credentials to one of the user roles.

Assets covered D::WORKSPACE.DEFINITION D::DOC.DATA D::DOC.METADATA D::USER.PLAN.KEYS D::ENTERPR

T::USER.AGENT.COMPROMISE An attacker modifies the UA presented to the user.

Assets covered D::USER.AGENT

Threats on the server

This section only considers the remote attacker case due to the **A::SERVER.PHYSICAL.ENVIRONMENT** assumption.

T::CONFIG.ACCESS An attacker is able to read or write the CSS configuration.

Assets covered D::CONFIG.CSS.SM D::ENTERPRISE.PLAN.KEYS

T::ADMIN.ROLE.USURPATION An attacker gets the admin credentials, either by accessing the administrator private keys or passphrase, either by adding his or her email on the configuration file.

Assets covered D::CONFIG.CSS.SM D::ENTERPRISE.PLAN.KEYS

T::LOGS.PROTECTION A remote attacker is able to read or modify the content of the CSS logs.

Assets covered D::AUDIT.LOGS

T::DATABASE.MODIFICATION A remote attacker is able to modify the content of the database by inserting or removing lines or performing database content rollback; in order to (non exhaustive list):

- Remove the revoked status of an user
- Delete logs
- Restore a workspace access control list in a previous state (before a given user has been removed from this workspace for instance)
- Be able to re-initiate an account recovery procedure without administrator validation[1].

This threat does not concern raw modification of sensitive data contained in the database, which is covered by other threats[2] therefore these data have an independent integrity protection. This threat covers another class of attacks which only uses previously valid data.

Assets covered D::USER.PLAN.KEYS D::ENTERPRISE.PLAN.KEYS D::AUDIT.LOGS D::WORKSPACE.DEFINITION D::USERS.PUBLIC.CERTIFICATES

[1] Note that the access recovery procedure itself forbids the attacker to recover the account

[2] **T::DOC.DATA.COMPROMISE**, **T::DOC.METADATA.COMPROMISE**, **T::WORKSPACE.COMPROMISE**, etc.

3.4 OSPs

This part describes the organizational security policies(OSPs) CRYPTOBOX shall conform to. We divide these OSPs into 3 categories :

- the directives and rules that CRYPTOBOX shall follow;
- the security policies that CRYPTOBOX usage allows to cover;
- OSPs which allow reaching the aimed security level.

3.4.1 DIRECTIVES AND RULES

OSP::RGS.CRYPTO Cryptographic mechanisms are conform to rules and recommendations from [ANS14]. Keys are generated with mechanisms conformant to rules and recommendations from [ANS12].

3.4.2 USAGE OSPS

OSP::DOC.PROTECTION The TOE shall provide security services for protecting the users' documents.The TOE shall provide at least two layers of protection:

1. an end-to-end layer for the users,
2. an Enterprise control layer on the assets access.

Protected assets D::DOC.DATA

OSP::METADATA.PROTECTION The TOE shall provide security services for protecting the users' documents metadata. The TOE shall provide at least two layers of protection:

1. an end-to-end layer for the users,
2. an Enterprise control layer on the assets access.

Protected assets D::DOC.METADATA

3.4.3 SECURITY ENSURING OSPS

OSP::PASSPHRASE.STRENGTH The TOE administrators can choose the minimum strength level required for the user's passphrase to allow registration on the TOE. The metrics are established as following:

- match: enumerates the common used passphrase patterns detected in the user's passphrase, like keyboard sequences or names,
- estimation: estimates the entropy of each matched pattern
- entropy: searches for the combination of pattern that gives the lowest entropy i.e the smallest patterns combination that gives that passphrase,
- cracking: estimates the cracking time of the previous combination with various scenarios
- score: gives a final score from the resulting time.

OSP::ROLE The TOE must limit the extent of users' abilities to use the TOE functions in accordance with the current user authenticated state and defined role: i.e. only authenticated users with the credentials of a specific role, can exercise the corresponding rights.

OSP::AUDIT The TOE shall record activity logs for audit.

4 – Security objectives

4.1 Security objectives

4.1.1 Security objectives on services provided by the TOE

O::DATA.PROTECTION The TOE shall provide mechanisms to protect the confidentiality, integrity and authenticity of the data of the users' documents. These mechanisms shall enforce the User and Enterprise security plans and be compliant with the rules and recommendations from [ANS14].

O::METADATA.PROTECTION The TOE shall provide mechanisms to protect the confidentiality, integrity and authenticity of the metadata of the users' documents. These mechanisms shall enforce the User and Enterprise security plans and be compliant with the rules and recommendations from [ANS14].

O::WORKSPACE.PROTECTION The TOE shall provide mechanisms to protect the confidentiality, integrity and authenticity of the user workspaces. These mechanisms shall enforce the User and Enterprise security plans and be compliant with the rules and recommendations from [ANS14].

O::EUS.PROTECTION The TOE shall provide mechanisms to protect the confidentiality, integrity and authenticity of the EUS. These mechanisms shall enforce the User and Enterprise security plans and be compliant with the rules and recommendations from [ANS14].

O::AUTHENTICATION The TOE shall provide mechanisms to authenticate the users and the administrators. These mechanisms shall enforce the User and Enterprise security plans and be compliant with the rules and recommendations from [ANS14].

O::PASSPHRASE.STRENGTH The TOE shall provide mechanisms to enforce quality metrics on the users' passphrase. These mechanisms shall enforce the Enterprise security.

O::PASSPHRASE.CSS.SM.PROTECTION The TOE shall provide mechanisms to authenticate the users and the administrators without storing the passphrases in clear at the CSS/SM. These mechanisms shall enforce the Enterprise security plan and be compliant with the rules and recommendations from [ANS14].

O::ACCOUNT.RECOVERY The TOE shall provide mechanisms for the users and the administrators to retrieve their accounts in case of credentials loss. These mechanisms shall enforce the User and Enterprise security plans and be compliant with the rules and recommendations from [ANS14].

O::CSS.SM.SUPERENCRYPTION The TOE shall provide mechanisms to protect the confidentiality, integrity and authenticity of the data of the CSS/SM that are stored in its database. These mechanisms shall enforce the Enterprise security plan and be compliant with the rules and recommendations from [ANS14].

O::COMPONENT.COMMUNICATION The TOE shall provide mechanisms to secure the connection between the SM and the clients. These mechanisms shall enforce the User and Enterprise security plans and be compliant with the rules and recommendations from [ANS14].

The TOE shall provide mechanisms to secure the connection between the CSS/SM and the database. These mechanisms shall be compliant with the rules and recommendations from [ANS14].

4.1.2 Security objectives to protect the TOE sensitive assets

O::ROLE.SEPARATION The TOE must limit the extent of users' abilities to use the TOE functions in accordance with the current user/administrator authenticated state and defined role: only authenticated users with the credentials of a specific role, can exercise the corresponding rights.

O::AUDIT.LOGS The TOE shall provide mechanisms to record audit logs.

O::AUDIT.PROTECTION The TOE shall provide mechanisms to protect the audit logs sensitive contents in confidentiality, integrity and authenticity.

4.2 Security Objectives for the environment of the TOE

OE::TRAINED.ADMIN The TOE administrators shall not be hostile and shall be competent persons with necessary resources for the implementation of their tasks. They shall be trained to perform the operations for which they are responsible and they follow manuals and administration procedures. In particular, they are able to choose passphrases with adequate security.

OE::TRAINED.USER Users shall not be careless, wilfully negligent, or hostile, and shall follow the instructions provided by documentation to perform their roles(owner,member). In particular, they are able to choose passphrases with adequate security. They also have another channel to do out-of-band verifications. For instance two users may have a phone call to verify their respective certificates fingerprints.

OE::PHYSICAL.ENVIRONMENT The servers operating the CSS/SM and the database are not physically accessible by an attacker.

OE::SOFTWARE.ENVIRONMENT The operating systems of the CSS/SM, the database and the UAs provide adequate security, including domain separation and non-bypassability, to counter the perceived threats. The OS ensures:

- applicative memory separation (no other applicative process can access the CSS/SM or UAs memory),
- for the UAs only, sensitive input protection (concerning passphrase entry for instance),
- For the Android UA, signature validation of the Cryptobox application before installation and update.

Moreover software libraries on which the CSS/SM rely correctly provide the services they are intended for, including security services.

Administrators of the CSS/SM server and database server keep them up to date and apply security patches when available.

Owners of UA keep their device operating system (either Android or PC OS and browser) up to date and apply security patches when available.

OE::TLS The client device has a correct and secure implementation of TLS. This encapsulates the connection between the client and the server.

5 – Security requirements

5.1 Security functional requirements

5.1.1 Definitions

5.1.2 Policy elements definition

This policies defined in this document use the definitions from this subsection.

5.1.2.1 Subjects

- **S::User** This is the TOE user, more precisely the user of the client application. The TOE user has the following attributes:
  - **ATT::Status** This is the user status that can be set to the following values:
    - **ATT::Status.Authorised** The user has access to the TOE,
    - **ATT::Status.Deleted** An administrator removes the user's access to the TOE.
  - **ATT::Signature.Certificate** : This is the X.509 certificate of the user embedding his signature public key.
  - **ATT::Encryption.Certificate** : This is the X.509 certificate of the user embedding his encryption public key.
  - **ATT::Sigma.Auth.Key** : This represents the authentication key obtained from the Sigma protocol.
  - **ATT::Spake.Auth.Key** : This represents the authentication key obtained from the Spake protocol.
  - **ATT::Admin** This user is a TOE administrator.
  - **ATT::Has.Trustee** This represents the user trustee(s).
  - **ATT::Is.Trustee** this represents if the user is someone trustee.
- **S::Guest** This is an entity external to the system. This subject allows defining the file sharing feature.

5.1.2.2 Objects

- **OB::Workspace** This is a CryptoBox workspace, with the following attributes

- **ATT::Author.Signature** This is the list of signatures of workspace definitions by legitimate workspace users.
- **ATT::SM.Signature** This is the signature of the workspace definition by the SM.
- **ATT::Owners** This is the list of users defined as workspace owners.
- **ATT::Readers** This is the list of users who may read documents from the workspace.
- **ATT::Writers** This is the list of users who may write documents to the workspace.

- **OB::Document** This is a document handled by the CRYPTOBOX system, with the following attributes
  - **ATT::Workspace** This is the workspace the document belongs to. It references a OB::Workspace object.
  - **ATT::File.Sharing.Token** This is the authentication token provided by a file sharing link.
  - **ATT::File.Sharing.Validity** This represents the validity information of a file sharing key, including expiration date.
  - **ATT::Server.File.Sharing.Key** This represents the server key part in the case of a file sharing.
  - **ATT::Server.Content.Key** This is the server managed part of the key, used for document protection.
- **OB::Metadata** This represents document's metadata, with the following attributes
  - **ATT::Document** This is the document the metadata are attached to. It references a OB::Document object.
- **OB::EUS** This is an EUS.
- **OB::Passphrase.Derived.Key** This is the user key derived from his passphrase. This object is used during account recovery.
- **OB::Administration.Interface** This is the interface that enables administration functions such as user management.

5.1.2.3 OPERATIONS

- **OPE::WRITE** This operation represents the creation or update of the document data or metadata
- **OPE::READ** This operation represents reading the contents of the document data or metadata.
- **OPE::DOCUMENT.SHARE** This operation represents the action of a user sharing a document with a S::Guest.
- **OPE::WORKSPACE.CREATE** This operation represents the action of creating the workspace definition.
- **OPE::WORKSPACE.UPDATE** This operation represents the action of updating the workspace definition (i.e. managing OB::Workspace attributes).
- **OPE::EUS.GET** This operation represents the action of accessing the EUS.
- **OPE::EUS.UPLOAD** This operation represents the action of uploading the EUS.
- **OPE::AUTHENTICATION.SPAKE** This authentication represent the first step of the user authentication, based on the passphrase.
- **OPE::AUTHENTICATION.SIGMA** This operation represents the second step of the user authentication, based on the signature certificate.
- **OPE::START.RECOVERY** This operation represents the user starting an account recovery process.
- **OPE::JOIN.RECOVERY** This operation represents the user's trustee joining the account recovery process.
- **OPE::ADMINISTRATION** This operation represents TOE administration functions.

5.1.3 CRYPTOGRAPHIC REQUIREMENTS

This section defines security requirements for cryptographic operation used for enforcing several security objectives.

FCS_COP.1/aes.gcm Cryptographic operations

FCS_COP.1.1/aes.gcm The TSF shall perform *[file chunk and EUS encryption and integrity protection]* in accordance with a specified cryptographic algorithm *[AES in GCM mode]* and cryptographic key sizes *[256]* that meet the following: *[[NIoSaT07] and [NIoSaT01a] and ANSSI cryptographic referentials ([ANS14] and [ANS12])]*.

FCS_COP.1/aes.cbc Cryptographic operations

FCS_COP.1.1/aes.cbc The TSF shall perform *[file chunk key and EUS encryption protection]* in accordance with a specified cryptographic algorithm *[AES in CBC mode]* and cryptographic key sizes *[256]* that meet the following: *[[NIoSaT01b] and [NIoSaT01a] and ANSSI cryptographic referentials ([ANS14] and [ANS12])]*.

5.1.4 Workspace and document protection related SFRs

5.1.4.1 User security plan

FCS_COP.1/signature Cryptographic operations

FCS_COP.1.1/signature The TSF shall perform *[workspace definition signature and verification]* in accordance with a specified cryptographic algorithm *[ECDSA with sha256]* and cryptographic key sizes *[256]* that meet the following: *[[NIoSaT13] and [NIoSaT15b] and ANSSI cryptographic referentials ([ANS14] and [ANS12])]*.

FCS_CKM.3/user.key Cryptographic key access

FCS_CKM.3.1/user.key The TSF shall perform *[workspace user key access control]* in accordance with a specified cryptographic key access method *[workspace user key wrapping]* that meets the following: *[ECIES as defined in [Sho04], and ANSSI cryptographic referentials ([ANS14] and [ANS12])]*.

5.1.4.2 Access control rules to workspaces and documents (Enterprise security plan)

See **SF::Workspace.Access.Control** for details on **WORKSPACE.ACCESS.CONTROL.SFP**.

FDP_ACC.1/workspace Subset access control

FDP_ACC.1.1/workspace The TSF shall enforce the *[**WORKSPACE.ACCESS.CONTROL.SFP**]* on *[*

- *Subjects: S::User*
- *Objects: OB::Workspace, OB::Document, OB::Metadata.*
- *Operations :*
  - *Workspace operations: OPE::WORKSPACE.CREATE, OPE::WORKSPACE.UPDATE*
  - *Document and Metadata operations: OPE::WRITE, OPE::READ, OPE::DOCUMENT.SHARE*

].

FDP_ACF.1/workspace Security attribute based access control

FDP_ACF.1.1/workspace The TSF shall enforce the *[**WORKSPACE.ACCESS.CONTROL.SFP**]* to objects based on the following: *[*

SUBJECT OR OBJECT	SECURITY ATTRIBUTES
S::Guest	*None.*
S::User	*ATT::Sigma.Auth.Key*
	ATT::Status
OB::Workspace	*ATT::Owners*
	ATT::Readers
	ATT::Writers
	ATT::Author.Signature
	ATT::SM.Signature
OB::Document	*ATT::Workspace*
	ATT::Server.File.Sharing.Key
	ATT::File.Sharing.Token
	ATT::File.Sharing.Validity
OB::Metadata	*ATT::Document*

].

FDP_ACF.1.2/workspace The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: *[*

- *S::User subject may perform the OPE::WORKSPACE.UPDATE operation on OB::Workspace only if*
  - *S::User appears in the ATT::Owners list of this OB::Workspace*
  - *The update contains an updated ATT::Author.Signature attribute which is a valid signature of last the workspace definition by S::User*

  At the end of the operation the SM updates the ATT::SM.Signature attribute.
- *S::User subject may perform the OPE::WRITE operation on OB::Document or OB::Metadata only if*
  - *the ATT::Workspace references a valid OB::Workspace object in case of operation on OB::Document*
  - *the ATT::Document references a valid OB::Document object in case of operation on OB::Metadata*
  - *S::User appears in the ATT::Writers list of this OB::Workspace*

  At the end of the operation the SM stores the ATT::Server.Content.Key sent by the subject.
- *S::User subject may perform the OPE::READ operation on OB::Document only if*
  - *the ATT::Workspace references a valid OB::Workspace object in case of operation on OB::Document*
  - *the ATT::Document references a valid OB::Document object in case of operation on OB::Metadata*
  - *S::User appears in the ATT::Readers list of this OB::Workspace*
- *S::User subject may perform the OPE::DOCUMENT.SHARE operation on OB::Document only if*
  - *the ATT::Workspace references a valid OB::Workspace object*

- ◇ *S::User appears in the ATT::Readers list of this OB::Workspace*

At the end of the operation the SM stores the ATT::File.Sharing.Token, ATT::Server.File.Sharing.Key and ATT::File.Sharing.Validity sent by the subject.

].

FDP_ACF.1.3/workspace The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: *[*

- ○ *S::User subject may perform the OPE::WORKSPACE.CREATE operation on OB::Workspace. At the end of the operation the SM updates the ATT::SM.Signature attribute.*
- ○ *S::User subject may perform the OPE::WORKSPACE.UPDATE operation on OB::Workspace if the only event operation was an OPE::WORKSPACE.CREATE operation and S::User requested OPE::WORKSPACE.CREATE*
- ○ *S::Guest subject may perform the OPE::READ operation on OB::Document if*
  - ◇ *S::Guest presented an authentication token matching ATT::File.Sharing.Token*
  - ◇ *the ATT::File.Sharing.Validity attribute is valid. i.e.*
    - ∗ *the sharing has not expired*

At the end of the operation the SM returns the ATT::Server.File.Sharing.Key to the subject.

].

FDP_ACF.1.4/workspace The TSF shall explicitly deny access of subjects to objects based on the following additional rules: *[*

- ○ *Any operation of S::User on OB::Workspace shall be refused if the ATT::SM.Signature is not a valid signature by the SM of the workspace definition*
- ○ *Any operation of S::User on OB::Document shall be refused if the ATT::SM.Signature of the OB::Workspace referenced by ATT::Workspace is not a valid signature by the SM of the workspace definition*
- ○ *Any operation of S::User on OB::Workspace or OB::Document shall be refused if the operation is not properly authenticated using ATT::Sigma.Auth.Key.*

].

FMT_MSA.3/workspace Static attribute initialisation

FMT_MSA.3.1/workspace The TSF shall enforce the *[**WORKSPACE.ACCESS.CONTROL.SFP**]* to provide *[restrictive]* default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/workspace The TSF shall allow the *[Workspace Owners]* to specify alternative initial values to override the default values when an object or information is created.

FMT_MSA.1/sharing Management of security attributes

FMT_MSA.1.1/sharing The TSF shall enforce the *[**WORKSPACE.ACCESS.CONTROL.SFP**]* to restrict the ability to *[change_default, modify]* the security attributes *[ATT::File.Sharing.Validity]* to *[the Workspace writer who initiates the file sharing]* and to the *[Workspace Owners]*.

FMT_MSA.1/workspace Management of security attributes

FMT_MSA.1.1/workspace The TSF shall enforce the *[**WORKSPACE.ACCESS.CONTROL.SFP**]* to restrict the ability to *[modify]* the security attributes *[ATT::Owners, ATT::Readers, ATT::Writers]* to *[Workspace Owners]*.

FMT_SMF.1/workspace Specification of Management Functions

FMT_SMF.1.1/workspace The TSF shall be capable of performing the following management functions: *[*

- ○ *Change a user role inside a workspace*
- ○ *Modify the file sharing attributes*

].

5.1.5 User Role related SFRs

FMT_SMR.1 Security roles

FMT_SMR.1.1 The TSF shall maintain the roles [

- TOE User. This role can be refined inside a given workspace to:
  - Guest if the user is a not member of the workspace
  - Workspace Member otherwise, once again refined into:
    - Workspace Owner
    - Workspace Reader
    - Workspace Writer
- TOE User Trustee, for account recovery.
- TOE Administrator.

].

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

5.1.6 Server access related SFRs

This section describe all server access related SFRs before accessing workspaces and documents. See **SF::Server.Access.Control** for details on **SERVER.ACCESS.CONTROL.SFP**.

The EUS is protected using the followings SFRs:

1. FCS_COP.1/aes.gcm
2. FCS_COP.1/aes.cbc

FDP_ACC.1/server Subset access control

FDP_ACC.1.1/server The TSF shall enforce the *[**SERVER.ACCESS.CONTROL.SFP**]* on *[*

- *Subjects: S::User*
- *Objects:*
  - *OB::EUS*
  - *OB::Administration.Interface*
- *Operations :*
  - *OPE::EUS.GET*
  - *OPE::EUS.UPLOAD*
  - *OPE::AUTHENTICATION.SPAKE*
  - *OPE::AUTHENTICATION.SIGMA*
  - *OPE::ADMINISTRATION*

].

FDP_ACF.1/server Security attribute based access control

FDP_ACF.1.1/server The TSF shall enforce the ***[SERVER.ACCESS.CONTROL.SFP]*** to objects based on the following: *[*

Subject or Object	Security attributes
S::User	ATT::Status
	ATT::Spake.Auth.Key
	ATT::Admin
OB::EUS	none
OB::Administration.Interface	none

].

FDP_ACF.1.2/server The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: *[*

- *S::User may perform OPE::EUS.UPLOAD only if he is authenticated to the server thus has ATT::Spake.Auth.Key.*
- *S::User may perform OPE::EUS.GET only if he is authenticated to the server thus has ATT::Spake.Auth.Key.*

].

FDP_ACF.1.3/server The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: *[ ].*

FDP_ACF.1.4/server The TSF shall explicitly deny access of subjects to objects based on the following additional rules:*[*

- *S::User shall be denied access to OB::Administration.Interface if he or she is not in the TOE administrators list in the server configuration file thus has attribute ATT::Admin.*
- *S::User shall be denied access to OB::EUS if S::User has the status ATT::Status.Deleted.*

].

FMT_MSA.3/server Static attribute initialization

FMT_MSA.3.1/server The TSF shall enforce the ***[SERVER.ACCESS.CONTROL.SFP]*** to provide *[restrictive]* default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/server The TSF shall allow the *[TOE Administrators]* to specify alternative initial values to override the default values when an object or information is created.

FMT_MSA.1/server Management of security attributes

FMT_MSA.1.1/server The TSF shall enforce the ***[SERVER.ACCESS.CONTROL.SFP]*** to restrict the ability to *[modify]* the security attributes *[ATT::Status]* to *[TOE Administrator(s)]*.

FMT_SMF.1/server Specification of Management Functions

FMT_SMF.1.1/server The TSF shall be capable of performing the following management functions: *[Change user status, Validate user account recovery]*.

5.1.7 Clients/Server Communication protection related SFRs

FCS_COP.1/hmac Cryptographic operations

FCS_COP.1.1/hmac The TSF shall perform *[HMAC-based key derivation]* in accordance with a specified cryptographic algorithm *[HMAC with SHA256]* and cryptographic key sizes *[256 bits]* that meet the following: *[[NIoSaT08] and [NIoSaT15b] and ANSSI cryptographic referentials ([ANS14] and [ANS12])]*.

FPT_ITT.1/communication Internal TOE TSF data transfer

FPT_ITT.1.1/communication The TSF shall protect TSF data from *[disclosure and modification]*when it is transmitted between separate parts of the TOE.

5.1.8 AUTHENTICATION RELATED SFRS

FCS_COP.1/ecies Cryptographic operations

FCS_COP.1.1/ecies The TSF shall perform *[key wrapping]* in accordance with a specified cryptographic algorithm *[ECIES]* and cryptographic key sizes *[256]* that meet the following: *[ECIES as defined in [Sho04] and ANSSI cryptographic referentials ([ANS14] and [ANS12])]*.

FCS_COP.1/spake2+ Cryptographic operation

FCS_COP.1.1/spake2+ The TSF shall perform *[passphrase-based authenticated key exchange]* in accordance with a specified cryptographic algorithm *[SPAKE2+]* and cryptographic key sizes *[256]* that meet the following: *[[CKS09] and ANSSI cryptographic referentials ([ANS14] and [ANS12])]*.

FCS_COP.1/sigma Cryptographic operation

FCS_COP.1.1/sigma The TSF shall perform *[authenticated key exchange]* in accordance with a specified cryptographic algorithm *[SIGMA]* and cryptographic key sizes *[256]* that meet the following: *[[Kra03] and ANSSI cryptographic referentials ([ANS14] and [ANS12])]*.

FIA_AFL.1/failure Authentication failure handling

FIA_AFL.1.1/failure The TSF shall detect when *[one]* unsuccessful authentication attempts occur related to *[user passphrase authentication]*.

FIA_AFL.1.2/failure When the defined number of unsuccessful authentication attempts has been *[met]*, the TSF shall *[*

- *add a delay of 500 milliseconds before responding to the client,*
- *queue the subsequent requests as long as one is being processed(thus delaying them)*

].

FIA_ATD.1 User attribute definition

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users: *[*

- *ATT::Status*
- *ATT::Signature.Certificate*
- *ATT::Encryption.Certificate*
- *ATT::Sigma.Auth.Key*
- *ATT::Spake.Auth.Key*
- *ATT::Has.Trustee*
- *ATT::Is.Trustee*
- *ATT::Admin*

].

FIA_SOS.1 Verification of secrets

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that secrets meet *[passphrase quality metrics]*.

FIA_UAU.2 User authentication before any action

FIA_UAU.2.1 The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

FIA_UAU.3 Unforgeable authentication

FIA_UAU.3.1 The TSF shall *[prevent]* use of authentication data that has been forged by any user of the TSF.

FIA_UAU.3.2 The TSF shall *[prevent]* use of authentication data that has been copied from any other user of the TSF.

5.1.9 Account recovery related SFRs

5.1.9.1 Backup process

The backup procedure is based on the following SFRs:

- FCS_COP.1/ecies
- FCS_COP.1/spake2+

5.1.9.2 Recovery process

FCS_COP.1/zrtp Cryptographic operations

FCS_COP.1.1/zrtp The TSF shall perform *[user and trustee real-time communication]* in accordance with a specified cryptographic algorithm *[ZRTP]* and cryptographic key sizes *[256]* that meet the following: *[ZRTP, as defined in [ZJC11] with ECDH [Sho04] and HMAC [NIoSaT08] and SHA256 [NIoSaT15b] and ANSSI cryptographic referentials ([ANS14] and [ANS12])]*.

See **SF::Account.Recovery** for details on **RECOVERY.ACCESS.CONTROL.SFP**.

FDP_ACC.1/recovery Subset access control

FDP_ACC.1.1/recovery The TSF shall enforce the *[**RECOVERY.ACCESS.CONTROL.SFP**]* on *[*

- *Subjects: S::User*
- *Objects: OB::Passphrase.Derived.Key*
- *Operations:*
  - *OPE::START.RECOVERY*
  - *OPE::JOIN.RECOVERY*

].

FDP_ACF.1/recovery Security attribute based access control

FDP_ACF.1.1/recovery The TSF shall enforce the *[**RECOVERY.ACCESS.CONTROL.SFP**]* to objects based on the following: *[*

Subject or Object	Security attributes
S::User	ATT::Status
	ATT::Has.Trustee
	ATT::Is.Trustee
OB::Passphrase.Derived.Key	none

].

FDP_ACF.1.2/recovery The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: *[*

- *S::User may perform OPE::START.RECOVERY only if he has declared a trustee thus has attribute ATT::Has.Trustee.*
- *S::User may perform OPE::JOIN.RECOVERY only if he is a trustee of that user thus has attribute ATT::Is.Trustee.*

].

FDP_ACF.1.3/recovery The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: *[ ]*.

FDP_ACF.1.4/recovery The TSF shall explicitly deny access of subjects to objects based on the following additional rules: *[*

- *S::User shall be denied access to OB::EUS if S::User has the status ATT::Status.Deleted.*
- *Any operation of S::User with attributes ATT::Has.Trustee or ATT::Is.Trustee on OB::Passphrase.Derived.Key shall be refused if recovery is not approved(email confirmation) by a S::User with attributes ATT::Admin.*
- *Any operation of S::User with attributes ATT::Has.Trustee or ATT::Is.Trustee on OB::Passphrase.Derived.Key shall be refused if S::User has the status ATT::Status.Deleted.*
- *S::User with attributes ATT::Has.Trustee shall be denied access to OB::Passphrase.Derived.Key if S::User has not confirmed the reception of the recovery email.*

].

FMT_MSA.3/recovery Static attribute initialization

FMT_MSA.3.1/recovery The TSF shall enforce the *[**RECOVERY.ACCESS.CONTROL.SFP**]* to provide *[restrictive]* default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/recovery The TSF shall allow the *[TOE Users]* to specify alternative initial values to override the default values when an object or information is created.

FMT_MSA.1/recovery Management of security attributes

FMT_MSA.1.1/recovery The TSF shall enforce the *[**RECOVERY.ACCESS.CONTROL.SFP**]* to restrict the ability to *[modify]* the security attributes *[ATT::Has.Trustee ATT::Is.Trustee]* to *[TOE Users]*.

FMT_SMF.1/recovery Specification of Management Functions

FMT_SMF.1.1/recovery The TSF shall be capable of performing the following management functions: *[Choose a trustee]*.

5.1.10 SUPER-ENCRYPTION PROTECTION RELATED SFRS

The CSS/SM super-encryption (see **O::CSS.SM.SUPERENCRYPTION**) is based on the followings SFRs:

1. FCS_COP.1/aes.gcm
2. FCS_COP.1/aes.cbc
3. FCS_CKM.1/random

5.1.11 AUDIT RELATED SFR

The sensitive data of the audit logs are protected using the followings SFRs:

- FCS_COP.1/aes.gcm

FAU_GEN.1 Audit data generation

FAU_GEN.1.1 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 for the *[not specified]* level of audit; and

c) [

- *file management: file access and file revision commit*
- *workspace management: workspace creation and workspace revision commit*
- *user authentication: invitation,registration, log in and passphrase change*
- *account recovery: trustee choice, recovery initiation and administrator validation*
- *user management: user registration, user passphrase modification and user suppression.*

] .

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

a) Date and time of the event, type of event, subject identity (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 PP/ST, *[not specified]*.

Application note: This SFR only defines the minimum audit requirements i.e. audit for events related to the TOE as required by the security object **O::AUDIT.LOGS** and **O::AUDIT.PROTECTION**. The TOE will also generate more audit data but those are not to be considered for the evaluation. This explains the *[not specified]* assignments in this SFR.

FPT_STM.1 Reliable time stamps

FPT_STM.1.1 The TSF shall be able to provide reliable time stamps.

5.1.12 GENERIC DEPENDENCIES

5.1.12.1 CRYPTOGRAPHIC KEYS MANAGEMENT

FCS_CKM.1/random Cryptographic key generation

FCS_CKM.1.1/random The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm *[random generation]* and specified cryptographic key sizes *[256 bits]* that meet the following: *[HMAC_DRBG with sha256 ([NIoSaT15a]) and ANSSI cryptographic referentials ([ANS14] and [ANS12])]*.

FCS_CKM.1/kdf.pbkdf Cryptographic key generation

FCS_CKM.1.1/kdf.pbkdf The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm *[PBKDF2 with SHA256 and HMAC with SHA256]* and specified cryptographic key sizes *[256 bits]* that meet the following: *[[NIoSaT10] and [NIoSaT08]) and [NIoSaT15b] and ANSSI cryptographic referentials ([ANS14] and [ANS12])]*.

FCS_CKM.1/kdf.sha256 Cryptographic key generation

FCS_CKM.1.1/kdf.sha256 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm *[SHA256]* and specified cryptographic key sizes *[256 bits]* that meet the following: *[[NIoSaT15b] and ANSSI cryptographic referentials ([ANS14] and [ANS12])]*.

FCS_CKM.4 Cryptographic key destruction

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method *[zeroization]* that meets the following: *[none]*.

FIA_UID.2 User identification before any action

FIA_UID.2.1 The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

5.2 TOE SECURITY ASSURANCE REQUIREMENTS

The evaluation target must comply with parts 2 and 3 of the Common Criteria version 3.1 reference 4 for the EAL3 level, augmented with ALC_FLR.3 and AVA_VAN.3. Table 5.4 summarizes the aimed assurance components.

ASSURANCE CLASS	ASSURANCE COMPONENT	DEFINITION
Development	ADV_ARC.1	Security architecture description
	ADV_FSP.3	Functional specification with complete summary
	ADV_TDS.2	Architectural design
Guidance documents	AGD_OPE.1	Operational user guidance
	AGD_PRE.1	Preparative procedures
Life-cycle support	ALC_CMC.3	Authorisation controls
	ALC_CMS.3	Implementation representation CM coverage
	ALC_DEL.1	Delivery procedures
	ALC_DVS.1	Identification of security measures
	ALC_LCD.1	Developer defined life-cycle model
	ALC_FLR.3	Systematic flaw remediation
Security Target evaluation	ASE_CCL.1	Conformance claims

Assurance class	Assurance component	Definition
	ASE_ECD.1	Extended components definition
	ASE_INT.1	ST introduction
	ASE_OBJ.2	Security objectives
	ASE_REQ.2	Derived security requirements
	ASE_SPD.1	Security problem definition
	ASE_TSS.1	TOE summary specification
Tests	ATE_COV.2	Analysis of coverage
	ATE_DPT.1	Testing: basic design
	ATE_FUN.1	Functional testing
	ATE_IND.2	Independent testing - sample
Vulnerability assessment	AVA_VAN.3	Vulnerability analysis

Table 5.4: Glossary and acronyms

6 – TOE SUMMARY SPECIFICATION

This chapter provides the TOE summary specification, a high-level definition of the security functions claimed to meet the functional and assurance requirements.

6.1 CRYPTOGRAPHY

SF::Cryptographic.Operations The TOE provides confidentiality, integrity and authenticity protection for users' documents data and metadata. The cryptographic operations provided are:

- encryption and decryption: AES scheme in GCM and CBC modes and ECIES,
- signature and hashing: ECDSA and SHA256,
- key exchange: SPAKE2+, SIGMA and ZRTP,
- key generation, derivation and destruction: HMAC_DRBG, HMAC, SHA256, PBKDF.

6.2 WORKSPACES AND DOCUMENTS

SF::Workspace.Upload The TOE users can work collaboratively with other users in workspaces. This security function enables the users to :

- create workspaces, which involves
  - the generation of the workspace encryption key
  - the addition of the workspace members and their corresponding rights
  - the encryption of the workspace key for each member
  - the signature of the workspace definition
- manage the members' roles by redefining the workspace definition and signing it
- add files and directories to the workspace, which involves:
  - the encryption of the files divided in chunk
  - the wrapping of the chunk keys with the workspace key and a server key
  - the generation of a hash tree of the workspace file directory and signature of the file revision by the writer

The TOE provides an interface to create or update a workspace definition and to add or update directories and files to the workspace. This security function uses **SF::Cryptographic.Operations** for cryptographic operations.

SF::Workspace.Access.Control The TOE implements access controls on the workspace so that only the users with the right access credentials can perform actions on the workspace or the documents in it. The access controls are:

1. logical to check if the user is a member or a guest of the workspace,
2. then cryptographic to check if the user has the decryption key needed and the workspace and hash tree signature are valid.

This security function uses **SF::Cryptographic.Operations** for cryptographic operations.

SF::Documents.Sharing The TOE enables users to share documents from a workspace with external collaborators, which then have a read access on the documents. This security function generate a new key per sharing. It also generates a shared file token needed, with the key, to access the file.

This security function uses **SF::Cryptographic.Operations** for cryptographic operations.

6.3 Inter-component communication

SF::Component.Communication The TOE provides secure communications between the user agents and the CSS/SM server; and between the CSS/SM and the database. Those communications are protected by the following mechanisms:

- Between the UA and the CSS/SM:
  1. Before Login: HTTPS with the CSS
  2. Login step 1: HTTPS with the CSS and SPAKE with the SM
  3. Login step 2: HTTPS with the CSS and SIGMA with the SM
  4. After Login: HTTPS with the CSS and SIGMA with the SM
- Between the CSS/SM and the database: TLS

This security function uses **SF::Cryptographic.Operations** for cryptographic operations.

6.4 Authentication

SF::User.Registration The TOE provides an interface for client applications to register new users. During the registration through the UA, the user:

- chooses a passphrase,
- generates a signature private key and certificate,
- generates a encryption private key and certificate,
- fill the information of his or her EUS,
- encrypts the EUS with a passphrase derived key,
- send the registration request to the CSS/SM.

This security function uses **SF::Cryptographic.Operations** for cryptographic operations.

SF::User.Login The TOE provides an interface for UAs applications to log users in. The authentication steps are:

1. passphrase based authentication to decrypt the EUS (SPAKE),
2. certificate based authentication with the signature certificate (SIGMA).

This security function uses **SF::Cryptographic.Operations** for cryptographic operations.

SF::Server.Access.Control The TOE implements access controls on the server so that only the users with the right access credentials can use the services the TOE provides. For instance users cannot access their EUS if their access is removed in the Enterprise plan. The access controls are:

1. logical to check if the user is registered or is deleted,
2. cryptographic with the SPAKE and the SIGMA.

This security function uses **SF::Cryptographic.Operations** for cryptographic operations.

6.5 Account recovery

SF::Account.Backup The TOE user can backup their credentials to prevent its loss. This function enables the user to:

1. choose a trustee
2. split the EUS key in two halves
3. encrypt one half for the trustee
4. encrypt the second half for the SM
5. send both encrypted halves to the CSS to process and store

This security function uses **SF::Cryptographic.Operations** for cryptographic operations.

SF::Account.Recovery The TOE users are able to recover their lost credentials if they have performed a account backup(**SF::Account.Backup**) first. The TOE implements access controls on the user deposit so that only the users with the right access credentials, provided with the help of one of their trustees, can perform an account recovery. The access controls are:

1. logical at first with email confirmations from the user, the trustee and the administrator,
2. then cryptographic with a ZRTP connection between the trustee and the user through the CSS/SM.

This security function uses **SF::Cryptographic.Operations** for cryptographic operations.

6.6 Administration

SF::Administration The TOE provides administration functions accessible only to the TOE users listed as administrators in the TOE configuration. The administrators can:

- set the passphrase policy **OSP::PASSPHRASE.STRENGTH**
- validate a user's account recovery
- delete a user
- see users details
- see workspaces name and members
- monitor the database and storage usage
- read the audit logs content

6.7 AUDIT

SF::Audit The TOE generates audit logs about security events and protects the sensitive content of those logs. They include the following fields : date, encrypted data, workspace ID (if appropriate) and file ID (if appropriate). The encrypted data are:

- CSS host name
- user email
- user certificate (hash)
- source IP
- user agent
- operation type
- user ID (When an action is done *on* a user)

This security function uses **SF::Cryptographic.Operations** for cryptographic operations.

7 – Rationales

7.1 Security objectives rationale

7.1.1 Coverage tables

Threat	Security objectives or security objective for environment
T::DOC.DATA.COMPROMISE	O::DATA.PROTECTION
T::DOC.METADATA.COMPROMISE	O::METADATA.PROTECTION
T::WORKSPACE.COMPROMISE	O::WORKSPACE.PROTECTION
T::PASSPHRASE.COMPROMISE	O::AUTHENTICATION O::PASSPHRASE.STRENG O::PASSPHRASE.CSS.SM.PROTECTION OE::TRAINED.USER
T::PASSPHRASE.LOSS	O::ACCOUNT.RECOVERY
T::EUS.COMPROMISE	O::EUS.PROTECTION O::PASSPHRASE.STRENG O::PASSPHRASE.CSS.SM.PROTECTION O::CSS.SM.SUPERENCRYPTION
T::KEY.EUS.COMPROMISE	O::ACCOUNT.RECOVERY O::CSS.SM.SUPERENCRYPTION
T::USER.PUBLIC.CERTIFICATES.MODIFICATION	O::CSS.SM.SUPERENCRYPTION
T::UA.CSS.SM.COMMUNICATION.COMPROMISE	O::COMPONENT.COMMUNICATION OE::TLS
T::USER.ROLE.USURPATION	O::WORKSPACE.PROTECTION O::AUTHENTICATI O::PASSPHRASE.STRENGTH O::PASSPHRASE.CSS.SM.PRO OE::TRAINED.USER

Threat	Security objectives or security objective for the environment
T::USER.AGENT.COMPROMISE	OE::SOFTWARE.ENVIRONMENT OE::TLS
T::CONFIG.ACCESS	OE::PHYSICAL.ENVIRONMENT OE::SOFTWARE.ENVIRONME
T::ADMIN.ROLE.USURPATION	O::AUTHENTICATION O::PASSPHRASE.STRENG O::PASSPHRASE.CSS.SM.PROTECTION OE::TRAINED.ADM
T::LOGS.PROTECTION	O::AUDIT.PROTECTION
T::DATABASE.MODIFICATION	O::COMPONENT.COMMUNICATION OE::TRAINED.ADMIN OE::PHYSICAL.ENVIRONME OE::SOFTWARE.ENVIRONMENT

Table 7.1: Threats coverage table

Assumption	Security objectives or security objective for the environment
A::ROLE.ADMIN	OE::TRAINED.ADMIN
A::ROLE.USER	OE::TRAINED.USER
A::TLS	OE::TLS
A::SERVER.SOFTWARE.ENVIRONMENT	OE::SOFTWARE.ENVIRONMENT
A::SERVER.PHYSICAL.ENVIRONMENT	OE::PHYSICAL.ENVIRONMENT
A::UA.OPERATING.ENVIRONMENT	OE::SOFTWARE.ENVIRONMENT

Table 7.2: Assumption coverage table

OSP	Security objectives or security objective for the environment
OSP::RGS.CRYPTO	O::DATA.PROTECTION O::METADATA.PROTECTION O::WORKSPACE.PROTECTION O::EUS.PROTECTION O::AUTHENTICATION O::PASSPHRASE.CSS.SM.PROTECTION O::ACCOUNT.RECOVERY O::CSS.SM.SUPERENCRYPTION O::COMPONENT.COMMUNICATION O::ROLE.SEPARATION O::AUDIT.PROTECTION
OSP::DOC.PROTECTION	O::DATA.PROTECTION
OSP::METADATA.PROTECTION	O::METADATA.PROTECTION
OSP::PASSPHRASE.STRENGTH	O::PASSPHRASE.STRENGTH

OSP	SECURITY OBJECTIVES OR SECURITY OBJECTIVE FOR THE ENVIRONMENT
OSP::ROLE	O::WORKSPACE.PROTECTION O::ROLE.SEPARATION
OSP::AUDIT	O::AUDIT.LOGS O::AUDIT.PROTECTION

Table 7.3: OSP coverage table

7.1.2 REVERSE COVERAGE TABLES

SECURITY OBJECTIVE FOR THE ENVIRONMENT	THREAT(S), OSP(S) OR ASSUMPTION
OE::TRAINED.ADMIN	A::ROLE.ADMIN T::ADMIN.ROLE.USURPATION T::DATABASE.MODIFICATION
OE::TRAINED.USER	A::ROLE.USER T::USER.ROLE.USURPATION T::PASSPHRASE.COMPROMISE
OE::PHYSICAL.ENVIRONMENT	A::SERVER.PHYSICAL.ENVIRONMENT T::DATABASE.MODIFICATION T::CONFIG.ACCESS
OE::SOFTWARE.ENVIRONMENT	A::SERVER.SOFTWARE.ENVIRONMENT A::UA.OPERATING.ENVIRONMENT T::USER.AGENT.COMPROMISE T::DATABASE.MODIFICATION T::CONFIG.ACCESS
OE::TLS	A::TLS T::UA.CSS.SM.COMMUNICATION.COMPROMISE T::USER.AGENT.COMPROMISE

Table 7.4: Security objective for the environment reverse coverage table

SECURITY OBJECTIVE	THREAT(S), OSP(S) OR ASSUMPTION
O::DATA.PROTECTION	T::DOC.DATA.COMPROMISE OSP::DOC.PROTECTION OSP::RGS.CRYPTO
O::METADATA.PROTECTION	T::DOC.METADATA.COMPROMISE OSP::METADATA.PROTECTION OSP::RGS.CRYPTO
O::WORKSPACE.PROTECTION	T::WORKSPACE.COMPROMISE T::USER.ROLE.USURPATION OSP::RGS.CRYPTO OSP::ROLE
O::EUS.PROTECTION	OSP::RGS.CRYPTO T::EUS.COMPROMISE
O::AUTHENTICATION	T::PASSPHRASE.COMPROMISE T::USER.ROLE.USURPATION T::ADMIN.ROLE.USURPATION OSP::RGS.CRYPTO
O::PASSPHRASE.STRENGTH	T::PASSPHRASE.COMPROMISE T::EUS.COMPROMISE T::USER.ROLE.USURPATION T::ADMIN.ROLE.USURPATION OSP::PASSPHRASE.STRENGTH

Security objective	Threat(s), OSP(s) or assumption
O::PASSPHRASE.CSS.SM.PROTECTION	OSP::RGS.CRYPTO T::PASSPHRASE.COMPROMISE T::EUS.COMPROMISE T::USER.ROLE.USURPATION T::ADMIN.ROLE.USURPATION
O::ACCOUNT.RECOVERY	T::PASSPHRASE.LOSS T::KEY.EUS.COMPROMISE OSP::RGS.CRYPTO
O::CSS.SM.SUPERENCRYPTION	T::KEY.EUS.COMPROMISE T::EUS.COMPROMISE T::USER.PUBLIC.CERTIFICATES.MODIFICATION OSP::RGS.CRYPTO
O::COMPONENT.COMMUNICATION	T::UA.CSS.SM.COMMUNICATION.COMPROMISE OSP::RGS.CRYPTO T::DATABASE.MODIFICATION
O::ROLE.SEPARATION	OSP::RGS.CRYPTO OSP::ROLE
O::AUDIT.LOGS	OSP::AUDIT
O::AUDIT.PROTECTION	OSP::RGS.CRYPTO OSP::AUDIT T::LOGS.PROTECTION

Table 7.5: Security objective reverse coverage table

7.1.3 Rationales

7.1.3.1 Threat covering

T::DOC.DATA.COMPROMISE

- **O::DATA.PROTECTION:**This security objective covers this threat by ensuring confidentiality, integrity and authenticity protection for the data of the users' documents.

T::DOC.METADATA.COMPROMISE

- **O::METADATA.PROTECTION:**This security objective covers this threat by ensuring confidentiality, integrity and authenticity protection for the metadata of the users' documents.

T::WORKSPACE.COMPROMISE

- **O::WORKSPACE.PROTECTION:**This security objective covers this threat by ensuring roles separation for the users in the workspace definition.

T::PASSPHRASE.COMPROMISE

- **O::AUTHENTICATION:** This security objectives covers this threat by ensuring that the server implements passphrase authentication mechanisms that reduce the risk of passphrase compromise.
- **O::PASSPHRASE.STRENGTH:** This security objectives covers this threat by ensuring quality metrics on the users' passphrase.
- **O::PASSPHRASE.CSS.SM.PROTECTION:** This security objectives covers this threat by ensuring that the server implements passphrase authentication mechanisms that reduce the risk of passphrase compromise.
- **OE::TRAINED.USER:**This OE forbids passphrase compromise by social engineering methods: users shall handle correctly their passphrase (they do not write it anywhere, and type it while not being watched).

T::PASSPHRASE.LOSS

- **O::ACCOUNT.RECOVERY:** This security objectives covers this threat by ensuring that the user can recover his account in case of passphrase loss if he has at least one *trustee*.

T::EUS.COMPROMISE

- **O::EUS.PROTECTION:**This security objectives covers this threat by ensuring that the TOE provides adequate protection for the EUS.
- **O::PASSPHRASE.STRENGTH:** This security objectives covers this threat by ensuring quality metrics on the users' passphrase.
- **O::PASSPHRASE.CSS.SM.PROTECTION:** This security objectives covers this threat by ensuring that the server implements passphrase authentication mechanisms that reduce the risk of passphrase compromise.
- **O::CSS.SM.SUPERENCRYPTION:** This security objectives covers this threat by ensuring that the EUS is stored in the database in a super-encrypted form by the CSS, on the top of user encryption.

T::KEY.EUS.COMPROMISE

- **O::ACCOUNT.RECOVERY:** This security objectives covers this threat by ensuring that the halves of the key encrypting the EUS are securely generated and stored.
- **O::CSS.SM.SUPERENCRYPTION:** This security objectives covers this threat by ensuring that the server half of the user EUS key is stored in the database in a super-encrypted form by the CSS.

T::USER.PUBLIC.CERTIFICATES.MODIFICATION

- **O::CSS.SM.SUPERENCRYPTION:** This security objectives covers this threat by ensuring the integrity and the authenticity of the user public certificates that are stored in the database .

T::UA.CSS.SM.COMMUNICATION.COMPROMISE

- **O::COMPONENT.COMMUNICATION:** This security objective covers this threat by ensuring secure communicatio between the clients and the server during sensitive content exchanges.
- **OE::TLS:** This security objective partially covers this threat by ensuring secure communications between the clients and the server.

T::USER.ROLE.USURPATION

- **O::WORKSPACE.PROTECTION:**This security objective covers this threat by ensuring roles separation for the users in the workspace definition.
- **O::AUTHENTICATION:** This security objectives covers this threat by ensuring that the server implements passphrase authentication mechanisms that reduce the risk of passphrase compromise.
- **O::PASSPHRASE.STRENGTH:** This security objectives covers this threat by ensuring quality metrics on the users' passphrase.
- **O::PASSPHRASE.CSS.SM.PROTECTION:** This security objectives covers this threat by ensuring that the server implements passphrase authentication mechanisms that reduce the risk of passphrase compromise.
- **OE::TRAINED.USER:**This OE forbids passphrase compromise by social engineering methods: users shall handle correctly their passphrase (they do not write it anywhere, and type it while not being watched).

T::USER.AGENT.COMPROMISE

- **OE::SOFTWARE.ENVIRONMENT:**This security objective covers this threat by guaranteeing signature validation before installation and update of the CRYPTOBOX Android UA.
- **OE::TLS:**This security objective covers this threat by guaranteeing integrity protection of the CRYPTOBOX Web UA.

T::CONFIG.ACCESS

- **OE::PHYSICAL.ENVIRONMENT:** This OE ensures that an attacker will not be able to physically access the CSS server to modify or read the CSS configuration.
- **OE::SOFTWARE.ENVIRONMENT:**This OE ensures that an attacker will not be able to exploit CSS server OS weakness to access CSS configuration.

T::ADMIN.ROLE.USURPATION

- **O::AUTHENTICATION:** This security objectives covers this threat by ensuring that the server implements passphrase authentication mechanisms that reduce the risk of passphrase compromise.
- **O::PASSPHRASE.STRENGTH:** This security objectives covers this threat by ensuring quality metrics on the users' passphrase.
- **O::PASSPHRASE.CSS.SM.PROTECTION:** This security objectives covers this threat by ensuring that the server implements passphrase authentication mechanisms that reduce the risk of passphrase compromise.
- **OE::TRAINED.ADMIN:**This OE forbids passphrase compromise by social engineering methods: administrators shall handle correctly their passphrase (they do not write it anywhere, and type it while not being watched).

T::LOGS.PROTECTION

- **O::AUDIT.PROTECTION:** This security objective covers this threat by ensuring protection of the sensitive contents of the audit logs.

T::DATABASE.MODIFICATION

- **O::COMPONENT.COMMUNICATION:** This security objective covers this threat by ensuring secure communicatio between the CSS/SM server and the database during sensitive content exchanges. In particular the database password is protected as well as requests to the database.
- **OE::TRAINED.ADMIN:** This OE ensures that TOE (database) administrator will not arm TOE internal data.
- **OE::PHYSICAL.ENVIRONMENT:** This OE ensures that an attacker will not be able to physically access the database to perform any attack.
- **OE::SOFTWARE.ENVIRONMENT:**This OE ensures that an attacker will not be able to exploit database server OS weakness to perform any attack.

7.1.3.2 OSP COVERING

OSP::RGS.CRYPTO

- **O::DATA.PROTECTION:** This security objectives ensures that cryptographic algorithms and key generation algorithms for data protection conform to the [ANS14] [ANS12].
- **O::METADATA.PROTECTION:** This security objectives ensures that cryptographic algorithms and key generation algorithms for metadata protection conform to the [ANS14] [ANS12].
- **O::WORKSPACE.PROTECTION:** This security objectives ensures that cryptographic algorithms and key generation algorithms for workspace protection conform to the [ANS14] [ANS12].
- **O::EUS.PROTECTION:** This security objectives ensures that cryptographic algorithms and key generation algorithms for the EUS protection conform to the [ANS14] [ANS12].
- **O::AUTHENTICATION:** This security objectives ensures that cryptographic algorithms and key generation algorithms for user authentication conform to the [ANS14] [ANS12].
- **O::PASSPHRASE.CSS.SM.PROTECTION:** This security objectives ensures that cryptographic algorithms and key generation algorithms for user passphrase protection conform to the [ANS14] [ANS12].
- **O::ACCOUNT.RECOVERY:** This security objectives ensures that cryptographic algorithms and key generation algorithms for user account recovery conform to the [ANS14] [ANS12].

- **O::CSS.SM.SUPERENCRYPTION:** This security objectives covers this threat by ensuring that cryptographic algorithms and key generation algorithms for CSS/SM super-encryption conform to the [ANS14] [ANS12].
- **O::COMPONENT.COMMUNICATION:** This security objectives ensures that cryptographic algorithms and key generation algorithms for client-server communication protection conform to the [ANS14] [ANS12].
- **O::ROLE.SEPARATION:** This security objectives ensures that cryptographic algorithms and key generation algorithms for role separation conform to the [ANS14] [ANS12].
- **O::AUDIT.PROTECTION:** This security objectives ensures that cryptographic algorithms and key generation algorithms for audit logs protection conform to the [ANS14] [ANS12].

OSP::DOC.PROTECTION

- **O::DATA.PROTECTION:** This security objective covers this OSP by ensuring confidentiality, integrity and authenticity protection for the data of the users' documents.

OSP::METADATA.PROTECTION

- **O::METADATA.PROTECTION:** This security objective covers this OSP by ensuring confidentiality, integrity and authenticity protection for the metadata of the users' documents.

OSP::PASSPHRASE.STRENGTH

- **O::PASSPHRASE.STRENGTH:** This security objectives ensures that users' passphrase are compliant with the quality metrics set by the administrators of the TOE.

OSP::ROLE

- **O::WORKSPACE.PROTECTION:** This security objective covers this OSP by ensuring roles separation for the users in the workspace definition.
- **O::ROLE.SEPARATION:** This security objective covers this OSP by ensuring role separation for the users of the TOE.

OSP::AUDIT

- **O::AUDIT.LOGS:** This security objective ensures this OSP by recording audit logs.
- **O::AUDIT.PROTECTION:** This security objective covers this OSP by protecting the sensitive contents of the audit logs.

7.1.3.3 Assumption covering

A::ROLE.ADMIN

- **OE::TRAINED.ADMIN:**This OE directly covers this assumption by defining the administrators behaviour

A::ROLE.USER

- **OE::TRAINED.USER:**This OE directly covers this assumption by defining the user behaviour

A::TLS

- **OE::TLS:**This OE covers this assumption by defining the software environment(TLS) of the client.

A::SERVER.SOFTWARE.ENVIRONMENT

- **OE::SOFTWARE.ENVIRONMENT:**This OE covers this assumption by defining the software environment of the CSS/SM and of the database.

A::SERVER.PHYSICAL.ENVIRONMENT

- **OE::PHYSICAL.ENVIRONMENT:**This OE covers this assumption by defining the physical environment of the CSS/SM and of the database.

A::UA.OPERATING.ENVIRONMENT

- **OE::SOFTWARE.ENVIRONMENT:**This OE covers this assumption by defining the software environment of the User Agents.

7.2 Security Function Requirements rationale

7.2.1 SFR dependencies

FCS_COP.1/aes.gcm

- FCS_CKM.1/random covers the dependency requirement "*[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]*"
- FCS_CKM.1/kdf.pbkdf covers the dependency requirement "*[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]*"
- FCS_CKM.4 covers the dependency requirement "*FCS_CKM.4 Cryptographic key destruction*"

FCS_COP.1/aes.cbc

- FCS_CKM.1/random covers the dependency requirement "*[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]*"
- FCS_CKM.4 covers the dependency requirement "*FCS_CKM.4 Cryptographic key destruction*"

FCS_COP.1/signature

- FCS_CKM.1/random covers the dependency requirement "*[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]*"
- FCS_CKM.4 covers the dependency requirement "*FCS_CKM.4 Cryptographic key destruction*"

FCS_CKM.3/user.key

- FCS_CKM.1/random covers the dependency requirement "*[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]*"
- FCS_CKM.4 covers the dependency requirement "*FCS_CKM.4 Cryptographic key destruction*"

FDP_ACC.1/workspace

- FDP_ACF.1/workspace covers the dependency requirement "*FDP_ACF.1 Security attribute based access control*"

FDP_ACF.1/workspace

- FDP_ACC.1/workspace covers the dependency requirement "*FDP_ACC.1 Subset access control*"
- FMT_MSA.3/workspace covers the dependency requirement "*FMT_MSA.3 Static attribute initialisation*"

FMT_MSA.3/workspace

- FMT_MSA.1/sharing covers the dependency requirement "*FMT_MSA.1 Management of security attributes*"
- FMT_MSA.1/workspace covers the dependency requirement "*FMT_MSA.1 Management of security attributes*"
- FMT_SMR.1 covers the dependency requirement "*FMT_SMR.1 Security roles*"

FMT_MSA.1/sharing

- FDP_ACC.1/workspace covers the dependency requirement "*[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]*"
- FMT_SMR.1 covers the dependency requirement "*FMT_SMR.1 Security roles*"
- FMT_SMF.1/workspace covers the dependency requirement "*FMT_SMF.1 Specification of Management Functions*"

FMT_MSA.1/workspace

- FDP_ACC.1/workspace covers the dependency requirement "*[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]*"
- FMT_SMR.1 covers the dependency requirement "*FMT_SMR.1 Security roles*"
- FMT_SMF.1/workspace covers the dependency requirement "*FMT_SMF.1 Specification of Management Functions*"

FMT_SMF.1/workspace

- No dependency.

FMT_SMR.1

- FIA_UID.2 covers the dependency requirement "*FIA_UID.1 Timing of identification*"

FDP_ACC.1/server

- FDP_ACF.1/server covers the dependency requirement "*FDP_ACF.1 Security attribute based access control*"

FDP_ACF.1/server

- FDP_ACC.1/server covers the dependency requirement "*FDP_ACC.1 Subset access control*"
- FMT_MSA.3/server covers the dependency requirement "*FMT_MSA.3 Static attribute initialisation*"

FMT_MSA.3/server

- FMT_MSA.1/server covers the dependency requirement "*FMT_MSA.1 Management of security attributes*"
- FMT_SMR.1 covers the dependency requirement "*FMT_SMR.1 Security roles*"

FMT_MSA.1/server

- FDP_ACC.1/server covers the dependency requirement "*[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]*"
- FMT_SMR.1 covers the dependency requirement "*FMT_SMR.1 Security roles*"
- FMT_SMF.1/server covers the dependency requirement "*FMT_SMF.1 Specification of Management Functions*"

FMT_SMF.1/server

- No dependency.

FCS_COP.1/hmac

- FCS_CKM.1/kdf.sha256 covers the dependency requirement "*[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]*"
- FCS_CKM.4 covers the dependency requirement "*FCS_CKM.4 Cryptographic key destruction*"

FPT_ITT.1/communication

- No dependency.

FCS_COP.1/ecies

- FCS_CKM.1/random covers the dependency requirement "*[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]*"
- FCS_CKM.4 covers the dependency requirement "*FCS_CKM.4 Cryptographic key destruction*"

FCS_COP.1/spake2+

- FCS_CKM.1/kdf.pbkdf covers the dependency requirement "*[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]*"
- FCS_CKM.4 covers the dependency requirement "*FCS_CKM.4 Cryptographic key destruction*"

FCS_COP.1/sigma

- FCS_CKM.1/random covers the dependency requirement "*[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]*"
- FCS_CKM.4 covers the dependency requirement "*FCS_CKM.4 Cryptographic key destruction*"

FIA_AFL.1/failure

- FIA_UAU.2 covers the dependency requirement "*FIA_UAU.1 Timing of authentication*"

FIA_ATD.1

- No dependency.

FIA_SOS.1

- No dependency.

FIA_UAU.2

- FIA_UID.2 covers the dependency requirement "*FIA_UID.1 Timing of identification*"

FIA_UAU.3

- No dependency.

FCS_COP.1/zrtp

- FCS_CKM.1/random covers the dependency requirement "*[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]*"
- FCS_CKM.4 covers the dependency requirement "*FCS_CKM.4 Cryptographic key destruction*"

FDP_ACC.1/recovery

- FDP_ACF.1/recovery covers the dependency requirement "*FDP_ACF.1 Security attribute based access control*"

FDP_ACF.1/recovery

- FDP_ACC.1/recovery covers the dependency requirement "*FDP_ACC.1 Subset access control*"
- FMT_MSA.3/recovery covers the dependency requirement "*FMT_MSA.3 Static attribute initialisation*"

FMT_MSA.3/recovery

- FMT_MSA.1/recovery covers the dependency requirement "*FMT_MSA.1 Management of security attributes*"
- FMT_SMR.1 covers the dependency requirement "*FMT_SMR.1 Security roles*"

FMT_MSA.1/recovery

- FDP_ACC.1/recovery covers the dependency requirement "*[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]*"
- FMT_SMR.1 covers the dependency requirement "*FMT_SMR.1 Security roles*"
- FMT_SMF.1/recovery covers the dependency requirement "*FMT_SMF.1 Specification of Management Functions*"

FMT_SMF.1/recovery

- No dependency.

FAU_GEN.1

- FPT_STM.1 covers the dependency requirement "*FPT_STM.1 Reliable time stamps*"

FPT_STM.1

- No dependency.

FCS_CKM.1/random

- FCS_COP.1/aes.gcm covers the dependency requirement "*[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation]*"
- FCS_COP.1/aes.cbc covers the dependency requirement "*[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation]*"
- FCS_CKM.4 covers the dependency requirement "*FCS_CKM.4 Cryptographic key destruction*"

FCS_CKM.1/kdf.pbkdf

- FCS_COP.1/spake2+ covers the dependency requirement "*[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation]*"
- FCS_CKM.4 covers the dependency requirement "*FCS_CKM.4 Cryptographic key destruction*"

FCS_CKM.1/kdf.sha256

- FCS_COP.1/spake2+ covers the dependency requirement "*[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation]*"
- FCS_COP.1/sigma covers the dependency requirement "*[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation]*"
- FCS_CKM.4 covers the dependency requirement "*FCS_CKM.4 Cryptographic key destruction*"

FCS_CKM.4

- FCS_CKM.1/random covers the dependency requirement "*[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]*"

FIA_UID.2

- No dependency.

7.2.2 Coverage tables

Security objectives	SFR

SECURITY OBJECTIVES	SFR
O::DATA.PROTECTION	FCS_COP.1/aes.gcm FCS_COP.1/aes.cbc FCS_CKM.3/user.key FDP_ACC.1/workspace FDP_ACF.1/workspace FMT_MSA.3/workspace FMT_MSA.1/sharing FMT_SMF.1/workspace FMT_SMR.1 FCS_CKM.1/random FCS_CKM.4 FIA_UID.2
O::METADATA.PROTECTION	FCS_COP.1/aes.gcm FCS_COP.1/aes.cbc FCS_CKM.3/user.key FDP_ACC.1/workspace FDP_ACF.1/workspace FMT_MSA.3/workspace FMT_SMR.1 FCS_CKM.1/random FCS_CKM.4 FIA_UID.2
O::WORKSPACE.PROTECTION	FCS_COP.1/aes.gcm FCS_COP.1/aes.cbc FCS_COP.1/signature FDP_ACC.1/workspace FDP_ACF.1/workspace FMT_MSA.3/workspace FMT_MSA.1/workspace FMT_SMF.1/workspace FMT_SMR.1 FIA_UID.2
O::EUS.PROTECTION	FCS_COP.1/aes.gcm FCS_COP.1/aes.cbc FMT_SMR.1 FDP_ACC.1/server FDP_ACF.1/server FMT_MSA.3/server FMT_MSA.1/server FMT_SMF.1/server FIA_ATD.1 FCS_CKM.1/random FCS_CKM.1/kdf.pbkdf
O::AUTHENTICATION	FCS_COP.1/hmac FCS_COP.1/ecies FCS_COP.1/spake2+ FCS_COP.1/sigma FIA_AFL.1/failure FIA_ATD.1 FIA_SOS.1 FIA_UAU.2 FIA_UAU.3 FCS_CKM.1/kdf.pbkdf FCS_CKM.1/kdf.sha256 FIA_UID.2
O::PASSPHRASE.STRENGTH	FIA_SOS.1
O::PASSPHRASE.CSS.SM.PROTECTION	FCS_COP.1/spake2+
O::ACCOUNT.RECOVERY	FMT_SMR.1 FCS_COP.1/ecies FCS_COP.1/spake2+ FIA_ATD.1 FCS_COP.1/zrtp FDP_ACC.1/recovery FDP_ACF.1/recovery FMT_MSA.3/recovery FMT_MSA.1/recovery FMT_SMF.1/recovery
O::CSS.SM.SUPERENCRYPTION	FCS_COP.1/aes.gcm FCS_COP.1/aes.cbc FCS_CKM.1/random
O::COMPONENT.COMMUNICATION	FCS_COP.1/hmac FPT_ITT.1/communication FCS_COP.1/spake2+ FCS_COP.1/sigma
O::ROLE.SEPARATION	FCS_COP.1/signature FDP_ACC.1/workspace FDP_ACF.1/workspace FMT_MSA.3/workspace FMT_MSA.1/workspace FMT_SMF.1/workspace FMT_SMR.1 FDP_ACF.1/server
O::AUDIT.LOGS	FAU_GEN.1 FPT_STM.1
O::AUDIT.PROTECTION	FCS_COP.1/aes.gcm

Table 7.6: Security objective coverage table

SFR	Security objective
FCS_COP.1/aes.gcm	**O::DATA.PROTECTION O::METADATA.PROTECTION O::WORKSPACE.PROTECTION O::EUS.PROTECTION O::CSS.SM.SUPERENCRYPTION O::AUDIT.PROTECTION**
FCS_COP.1/aes.cbc	**O::DATA.PROTECTION O::METADATA.PROTECTION O::WORKSPACE.PROTECTION O::EUS.PROTECTION O::CSS.SM.SUPERENCRYPTION**
FCS_COP.1/signature	**O::WORKSPACE.PROTECTION O::ROLE.SEPARATION**
FCS_CKM.3/user.key	**O::DATA.PROTECTION O::METADATA.PROTECTION**
FDP_ACC.1/workspace	**O::WORKSPACE.PROTECTION O::DATA.PROTECTION O::METADATA.PROTECTION O::ROLE.SEPARATION**
FDP_ACF.1/workspace	**O::WORKSPACE.PROTECTION O::DATA.PROTECTION O::METADATA.PROTECTION O::ROLE.SEPARATION**
FMT_MSA.3/workspace	**O::WORKSPACE.PROTECTION O::DATA.PROTECTION O::METADATA.PROTECTION O::ROLE.SEPARATION**
FMT_MSA.1/sharing	**O::DATA.PROTECTION**
FMT_MSA.1/workspace	**O::WORKSPACE.PROTECTION O::ROLE.SEPARATION**
FMT_SMF.1/workspace	**O::WORKSPACE.PROTECTION O::DATA.PROTECTION O::ROLE.SEPARATION**
FMT_SMR.1	**O::DATA.PROTECTION O::METADATA.PROTECTION O::WORKSPACE.PROTECTION O::EUS.PROTECTION O::ROLE.SEPARATION O::ACCOUNT.RECOVERY**
FDP_ACC.1/server	**O::EUS.PROTECTION**
FDP_ACF.1/server	**O::EUS.PROTECTION O::ROLE.SEPARATION**
FMT_MSA.3/server	**O::EUS.PROTECTION**
FMT_MSA.1/server	**O::EUS.PROTECTION**
FMT_SMF.1/server	**O::EUS.PROTECTION**
FCS_COP.1/hmac	**O::AUTHENTICATION O::COMPONENT.COMMUNICATION**
FPT_ITT.1/communication	**O::COMPONENT.COMMUNICATION**
FCS_COP.1/ecies	**O::AUTHENTICATION O::ACCOUNT.RECOVERY**
FCS_COP.1/spake2+	**O::COMPONENT.COMMUNICATION O::AUTHENTICATION O::PASSPHRASE.CSS.SM.PROTECTION O::ACCOUNT.RECOVERY**

SFR	Security objective
FCS_COP.1/sigma	**O::COMPONENT.COMMUNICATION O::AUTHENTICATION**
FIA_AFL.1/failure	**O::AUTHENTICATION**
FIA_ATD.1	**O::AUTHENTICATION O::EUS.PROTECTION O::ACCOUNT.RECOVERY**
FIA_SOS.1	**O::PASSPHRASE.STRENGTH O::AUTHENTICATION**
FIA_UAU.2	**O::AUTHENTICATION**
FIA_UAU.3	**O::AUTHENTICATION**
FCS_COP.1/zrtp	**O::ACCOUNT.RECOVERY**
FDP_ACC.1/recovery	**O::ACCOUNT.RECOVERY**
FDP_ACF.1/recovery	**O::ACCOUNT.RECOVERY**
FMT_MSA.3/recovery	**O::ACCOUNT.RECOVERY**
FMT_MSA.1/recovery	**O::ACCOUNT.RECOVERY**
FMT_SMF.1/recovery	**O::ACCOUNT.RECOVERY**
FAU_GEN.1	**O::AUDIT.LOGS**
FPT_STM.1	**O::AUDIT.LOGS**
FCS_CKM.1/random	**O::DATA.PROTECTION O::METADATA.PROTECTION O::EUS.PROTECTION O::CSS.SM.SUPERENCRYPTION**
FCS_CKM.1/kdf.pbkdf	**O::EUS.PROTECTION O::AUTHENTICATION**
FCS_CKM.1/kdf.sha256	**O::AUTHENTICATION**
FCS_CKM.4	**O::DATA.PROTECTION O::METADATA.PROTECTION**
FIA_UID.2	**O::AUTHENTICATION O::DATA.PROTECTION O::METADATA.PROTECTION O::WORKSPACE.PROTECTION**

Table 7.7: SFR reverse coverage table

7.2.3 SFR COVERAGE RATIONALE

O::DATA.PROTECTION

- **FCS_COP.1/aes.gcm:** This SFR ensures confidentiality and integrity protection of file content.
- **FCS_COP.1/aes.cbc:** This SFR ensures confidentiality of the chunk key wrapped by the workspace user key.

- **FCS_CKM.3/user.key:**This SFR enforces the user security plan by specifying how the workspace user key may be accessed by legitimate users.
- **FDP_ACC.1/workspace:**This SFR enforces the enterprise security plan by defining the access control rules to documents.
- **FDP_ACF.1/workspace:**This SFR enforces the enterprise security plan by specifying the access control rules to documents operations enforced by the server.
- **FMT_MSA.3/workspace:**This SFR enforces the enterprise security plan by specifying the initializations of attributes used for access control to document operations enforced by the server.
- **FMT_MSA.1/sharing:**This SFR defines the ability to share files to external entity.
- **FMT_SMF.1/workspace:**This SFR defines the file sharing management function.
- **FMT_SMR.1:**This SFR defines the security roles of the TOE.
- **FCS_CKM.1/random:**This SFR specifies how keys used for protecting files are generated.
- **FCS_CKM.4:**This SFR ensures that keys used for protecting files are adequately destroyed.
- **FIA_UID.2:**This SFR ensures that no access to files data is possible without proper user authentication.

O::METADATA.PROTECTION

- **FCS_COP.1/aes.gcm:**This SFR ensures confidentiality and integrity protection of file metadata.
- **FCS_COP.1/aes.cbc:**This SFR ensures confidentiality of the chunk key wrapped by the workspace user key.
- **FCS_CKM.3/user.key:**This SFR enforces the user security plan by specifying how the workspace user key may be accessed by legitimate users.
- **FDP_ACC.1/workspace:**This SFR enforces the enterprise security plan by defining the access control rules to documents metadata.
- **FDP_ACF.1/workspace:**This SFR enforces the enterprise security plan by specifying the access control rules to document's metadata operations enforced by the server.
- **FMT_MSA.3/workspace:**This SFR enforces the enterprise security plan by specifying the initializations of attributes used for access control to document operations enforced by the server.
- **FMT_SMR.1:**This SFR defines the security roles of the TOE.
- **FCS_CKM.1/random:**This SFR specifies how keys used for protecting metadata of the files are generated.
- **FCS_CKM.4:**This SFR ensures that keys used for protecting files metadata are adequately destroyed.
- **FIA_UID.2:**This SFR ensures that no access to files metadata is possible without proper user authentication.

O::WORKSPACE.PROTECTION

- **FCS_COP.1/aes.gcm:**This SFR ensures confidentiality and integrity protection of workspace metadata.
- **FCS_COP.1/aes.cbc:**This SFR ensures confidentiality protection of workspace user lists.
- **FCS_COP.1/signature:**This SFR ensures integrity and authenticity protection of workspace definitions. In particular this covers the user security plan by allowing end users to verify that all workspace definitions have been issued by a valid owner.
- **FDP_ACC.1/workspace:**This SFR enforces the enterprise security plan by defining the access control rules to workspace operations.
- **FDP_ACF.1/workspace:**This SFR enforces the enterprise security plan by specifying the access control rules to workspace operations enforced by the server.
- **FMT_MSA.3/workspace:**This SFR enforces the enterprise security plan by specifying the initializations of attributes used for access control to workspace operations enforced by the server.
- **FMT_MSA.1/workspace:**This SFR defines which user can change the access properties of a workspace.
- **FMT_SMF.1/workspace:**This SFR defines the workspace management function.
- **FMT_SMR.1:**This SFR defines the security roles of the TOE.
- **FIA_UID.2:**This SFR ensures that no access to workspace is possible without proper user authentication.

O::EUS.PROTECTION

- **FCS_COP.1/aes.gcm:**This SFR ensures confidentiality and integrity protection of the EUS content.
- **FCS_COP.1/aes.cbc:**This SFR ensures confidentiality protection of the user-encrypted EUS.
- **FMT_SMR.1:**This SFR defines the security roles of the TOE.
- **FDP_ACC.1/server:**This SFR covers this security objective by ensuring access control on the server.
- **FDP_ACF.1/server:**This SFR covers this security objective by defining the rules of the access control on the server.
- **FMT_MSA.3/server:**This SFR covers this security objective by defining the management rules of the access control on the server.
- **FMT_MSA.1/server:**This SFR covers this security objective by defining the management rules of the access control on the server.
- **FMT_SMF.1/server:**This SFR covers this security objective by defining the management rules of the access control on the server.
- **FIA_ATD.1:**This SFR covers this security objective by describing the user's authorization status.
- **FCS_CKM.1/random:**This SFR specifies how the key used for EUS Enterprise protection is generated.
- **FCS_CKM.1/kdf.pbkdf:**This SFR specifies how the passphrase derived key used for EUS protection is generated.

O::AUTHENTICATION

- **FCS_COP.1/hmac:**This SFR ensures that the client and the server have the same secret key at the end of each step of the authentication.
- **FCS_COP.1/ecies:**This SFR covers this security objective by describing a key wrapping during user registration.
- **FCS_COP.1/spake2+:**This SFR covers this security objective by describing a passphrase-based authentication.
- **FCS_COP.1/sigma:**This SFR covers this security objective by describing a certificate-based authentication.
- **FIA_AFL.1/failure:**This SFR covers this security objective by describing failure handling for passphrase-based authentication
- **FIA_ATD.1:**This SFR covers this security objective by describing the associated user security attributes.
- **FIA_SOS.1:**This SFR covers this security objective by setting passphrase quality metrics for the users.
- **FIA_UAU.2:**This SFR covers this security objective by describing requirements on the user authentication.
- **FIA_UAU.3:**This SFR covers this security objective by describing requirements on the user authentication.
- **FCS_CKM.1/kdf.pbkdf:**This SFR covers this security objective by describing a passphrase-based authentication
- **FCS_CKM.1/kdf.sha256:**This SFR ensures that the client and the server have the same secret key at the end of each step of the authentication.
- **FIA_UID.2:**This SFR covers this security objective by describing requirements on the user identification

O::PASSPHRASE.STRENGTH

- **FIA_SOS.1:**This SFR covers this security objective by setting passphrase quality metrics for the users.

O::PASSPHRASE.CSS.SM.PROTECTION

- **FCS_COP.1/spake2+:**This SFR covers this security objective by describing a passphrase-based authentication variant that does not store the passphrase in clear on the server.

O::ACCOUNT.RECOVERY

- **FMT_SMR.1:**This SFR defines the security roles of the TOE.
- **FCS_COP.1/ecies:**This SFR enforces the user security plan by specifying how the user passphrase key is wrapped for account recovery.
- **FCS_COP.1/spake2+:**This SFR ensures confidentiality and integrity protection of the account recovery deposit transmitted by the client to the server.
- **FIA_ATD.1:**This SFR covers this security objective by describing the associated user security attributes related to account recovery.
- **FCS_COP.1/zrtp:**This SFR enforces the user security plan by specifying how the user passphrase key is recovered.
- **FDP_ACC.1/recovery:**This SFR covers this security objective by ensuring access control on the user deposit for account recovery.
- **FDP_ACF.1/recovery:**This SFR covers this security objective by ensuring access control on the user deposit for account recovery.
- **FMT_MSA.3/recovery:**This SFR covers this security objective by defining the management rules of the access control on the user deposit for account recovery.
- **FMT_MSA.1/recovery:**This SFR covers this security objective by defining the management rules of the access control on the user deposit for account recovery.
- **FMT_SMF.1/recovery:**This SFR covers this security objective by defining the management rules of the access control on the user deposit for account recovery.

O::CSS.SM.SUPERENCRYPTION

- **FCS_COP.1/aes.gcm:**This SFR ensures confidentiality, authenticity and integrity protection of the data of the CSS/SM that are stored in its database.
- **FCS_COP.1/aes.cbc:**This SFR ensures confidentiality protection of the data of the CSS/SM that are stored in its database.
- **FCS_CKM.1/random:**This SFR ensures confidentiality, authenticity and integrity protection of the data of the CSS/SM that are stored in its database.

O::COMPONENT.COMMUNICATION

- **FCS_COP.1/hmac:**This SFR ensures that the client and the server have the same secret key at the end of each step of the authentication.
- **FPT_ITT.1/communication:**This SFR ensures confidentiality and integrity protection of the TSF data transmitted between the client and the server.
- **FCS_COP.1/spake2+:**This SFR ensures confidentiality protection of the TSF data transmitted between the client and the server.
- **FCS_COP.1/sigma:**This SFR ensures confidentiality protection of the TSF data transmitted between the client and the server.

O::ROLE.SEPARATION

- **FCS_COP.1/signature:**This SFR ensures integrity and authenticity protection of workspace definitions. In particular this covers the user security plan by allowing end users to verify that all workspace definitions and the roles in that workspace have been issued by a valid owner.
- **FDP_ACC.1/workspace:**This SFR enforces the enterprise security plan by defining the access control rules to workspace operations.
- **FDP_ACF.1/workspace:**This SFR enforces the enterprise security plan by defining the access control rules to workspace operations.
- **FMT_MSA.3/workspace:**This SFR enforces the enterprise security plan by specifying the initializations of attributes used for access control to document operations enforced by the server.

- **FMT_MSA.1/workspace:**This SFR defines which user can change the access properties of a workspace.
- **FMT_SMF.1/workspace:**This SFR defines the file sharing management function.
- **FMT_SMR.1:**This SFR defines the security roles of the TOE.
- **FDP_ACF.1/server:**This SFR covers this security objective by defining the rules of the administration interface access control on the server.

O::AUDIT.LOGS

- **FAU_GEN.1:**This SFR covers this security objective by defining the recording of security events reported by the CSS/SM.
- **FPT_STM.1:**This SFR covers this security objective by defining time stamps for auditing.

O::AUDIT.PROTECTION

- **FCS_COP.1/aes.gcm:**This SFR ensures confidentiality, authenticity and integrity protection of the audit logs sensitive data of the CSS/SM that are stored in its database.

7.3 Security Assurance Requirement rationale

The aimed assurance level for this security target is EAL3 augmented with ALC_FLR.3 and AVA_VAN.3. This level has been chosen to be conform to the French “Qualification Standard” package defined in [ANS].

The ALC_FLR.3 component has no dependencies.

The AVA_VAN.3 component has the following dependencies:

- ADV_ARC.1 Security architecture description,
- ADV_FSP.4 Complete 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,
- ATE_DPT.1 Testing: basic design.

The dependencies of the AVA_VAN.3 component with ADV_FSP.4, ADV_IMP.1 and ADV_TDS.3 are not satisfied.

This security target claims conformance with the assurance package defined by the standard qualification process. So all unsatisfied dependencies of [ANS] are left unchanged.

7.4 Security Function rationale

7.4.1 Coverage tables

SFR	Security function
FCS_COP.1/aes.gcm	**SF::Cryptographic.Operations**
FCS_COP.1/aes.cbc	**SF::Cryptographic.Operations**
FCS_COP.1/signature	**SF::Cryptographic.Operations**

SFR	Security function
FCS_CKM.3/user.key	**SF::Cryptographic.Operations**
FDP_ACC.1/workspace	**SF::Workspace.Access.Control**
FDP_ACF.1/workspace	**SF::Workspace.Access.Control**
FMT_MSA.3/workspace	**SF::Workspace.Upload SF::Documents.Sharing**
FMT_MSA.1/sharing	**SF::Workspace.Access.Control**
FMT_MSA.1/workspace	**SF::Workspace.Access.Control**
FMT_SMF.1/workspace	**SF::Workspace.Upload SF::Documents.Sharing**
FMT_SMR.1	**SF::Workspace.Upload SF::Documents.Sharing SF::User.Login SF::Account.Backup SF::Administration**
FDP_ACC.1/server	**SF::Server.Access.Control SF::Administration**
FDP_ACF.1/server	**SF::Server.Access.Control SF::Administration**
FMT_MSA.3/server	**SF::Administration**
FMT_MSA.1/server	**SF::Server.Access.Control**
FMT_SMF.1/server	**SF::Administration**
FCS_COP.1/hmac	**SF::Cryptographic.Operations**
FPT_ITT.1/communication	**SF::Component.Communication**
FCS_COP.1/ecies	**SF::Cryptographic.Operations**
FCS_COP.1/spake2+	**SF::Cryptographic.Operations**
FCS_COP.1/sigma	**SF::Cryptographic.Operations**
FIA_AFL.1/failure	**SF::Server.Access.Control**
FIA_ATD.1	**SF::User.Registration SF::User.Login SF::Account.Backup**
FIA_SOS.1	**SF::User.Registration**
FIA_UAU.2	**SF::Server.Access.Control**
FIA_UAU.3	**SF::User.Registration SF::User.Login**

SFR	Security function
FCS_COP.1/zrtp	**SF::Cryptographic.Operations**
FDP_ACC.1/recovery	**SF::Account.Recovery**
FDP_ACF.1/recovery	**SF::Account.Recovery**
FMT_MSA.3/recovery	**SF::Account.Backup**
FMT_MSA.1/recovery	**SF::Account.Recovery**
FMT_SMF.1/recovery	**SF::Account.Backup**
FAU_GEN.1	**SF::Audit**
FPT_STM.1	**SF::Audit**
FCS_CKM.1/random	**SF::Cryptographic.Operations**
FCS_CKM.1/kdf.pbkdf	**SF::Cryptographic.Operations**
FCS_CKM.1/kdf.sha256	**SF::Cryptographic.Operations**
FCS_CKM.4	**SF::Cryptographic.Operations**
FIA_UID.2	**SF::Server.Access.Control**

Table 7.8: Security Function Requirements coverage table

Security function	SFR
SF::Cryptographic.Operations	FCS_COP.1/aes.gcm FCS_COP.1/aes.cbc FCS_COP.1/signature FCS_CKM.4 FCS_COP.1/hmac FCS_CKM.1/random FCS_CKM.3/user.key FCS_CKM.1/kdf.pbkdf FCS_CKM.1/kdf.sha256 FCS_COP.1/spake2+ FCS_COP.1/sigma FCS_COP.1/ecies FCS_COP.1/zrtp
SF::Workspace.Upload	FMT_SMR.1 FMT_SMF.1/workspace FMT_MSA.3/workspace
SF::Workspace.Access.Control	FDP_ACC.1/workspace FDP_ACF.1/workspace FMT_MSA.1/workspace FMT_MSA.1/sharing
SF::Documents.Sharing	FMT_MSA.3/workspace FMT_SMF.1/workspace FMT_SMR.1
SF::Component.Communication	FPT_ITT.1/communication
SF::User.Registration	FIA_ATD.1 FIA_SOS.1 FIA_UAU.3

Security function	SFR
SF::User.Login	FIA_ATD.1 FIA_UAU.3 FMT_SMR.1
SF::Server.Access.Control	FIA_AFL.1/failure FIA_UID.2 FIA_UAU.2 FDP_ACC.1/server FDP_ACF.1/server FMT_MSA.1/server
SF::Account.Backup	FIA_ATD.1 FMT_SMR.1 FMT_SMF.1/recovery FMT_MSA.3/recovery
SF::Account.Recovery	FDP_ACC.1/recovery FDP_ACF.1/recovery FMT_MSA.1/recovery
SF::Administration	FDP_ACC.1/server FDP_ACF.1/server FMT_SMR.1 FMT_MSA.3/server FMT_SMF.1/server
SF::Audit	FAU_GEN.1 FPT_STM.1

Table 7.9: Security Function reverse coverage table

7.4.2 Security Function coverage rationale

FCS_COP.1/aes.gcm

- **SF::Cryptographic.Operations:**This security function implements the AES scheme in GCM mode.

FCS_COP.1/aes.cbc

- **SF::Cryptographic.Operations:**This security function implements the AES scheme in CBC mode.

FCS_COP.1/signature

- **SF::Cryptographic.Operations:**This security function implements ECDSA with SHA256 signature and verification.

FCS_CKM.3/user.key

- **SF::Cryptographic.Operations:**This security function implements user key wrapping with elliptic curve encryption.

FDP_ACC.1/workspace

- **SF::Workspace.Access.Control:**This security function implements access control on the workspaces, enforced by the server.

FDP_ACF.1/workspace

- **SF::Workspace.Access.Control:**This security function implements access control on the workspaces, enforced by the server.

FMT_MSA.3/workspace

- **SF::Workspace.Upload:**This security function allows workspaces owners to specify workspaces users' roles.
- **SF::Documents.Sharing:**This security function allows workspaces members to specify the document sharing attributes.

FMT_MSA.1/sharing

- **SF::Workspace.Access.Control:**This security function implements access control on the workspaces, enforced by the server, to modify the security attributes of a file sharing.

FMT_MSA.1/workspace

- **SF::Workspace.Access.Control:**This security function implements access control on the workspaces, enforced by the server to modify the security attributes of the workspace users.

FMT_SMF.1/workspace

- **SF::Workspace.Upload:**This security function handles workspaces users' roles management.
- **SF::Documents.Sharing:**This security function allows workspaces members to specify the document sharing attributes.

FMT_SMR.1

- **SF::Workspace.Upload:**This security function handles workspaces users' roles management.
- **SF::Documents.Sharing:**This security function associates user with the TOE role Guest for file sharing.
- **SF::User.Login:**This security function associates users with their TOE roles.
- **SF::Account.Backup:**This security function defines the user's *trustee*.
- **SF::Administration:**This security function associates the user with the role TOE administrator.

FDP_ACC.1/server

- **SF::Server.Access.Control:**This security function implements access control on the TOE services.
- **SF::Administration:**This security function implements access control on the TOE services.

FDP_ACF.1/server

- **SF::Server.Access.Control:**This security function implements access control on the TOE services.
- **SF::Administration:**This security function implements access control on the TOE services.

FMT_MSA.3/server

- **SF::Administration:**This security function allows TOE administrators to delete a user.

FMT_MSA.1/server

- **SF::Server.Access.Control:**This security function implements access control on the TOE administration function.

FMT_SMF.1/server

- **SF::Administration:**This security function allows TOE administrators to delete a user.

FCS_COP.1/hmac

- **SF::Cryptographic.Operations:**This security function implements HMAC-based key derivation.

FPT_ITT.1/communication

- **SF::Component.Communication:**This security function handles secure communications between the clients and the secure module(SM) of the server.

FCS_COP.1/ecies

- **SF::Cryptographic.Operations:**This security function implements elliptic curve encryption.

FCS_COP.1/spake2+

- **SF::Cryptographic.Operations:**This security function implements passphrase-based authentication.

FCS_COP.1/sigma

- **SF::Cryptographic.Operations:**This security function implements certificate-based authentication.

FIA_AFL.1/failure

- **SF::Server.Access.Control:**This security function handles authentication failures.

FIA_ATD.1

- **SF::User.Registration:**This security function handles the initiation of some users' attributes.
- **SF::User.Login:**This security function handles the initiation of some users' attributes.
- **SF::Account.Backup:**This security function handles the initiation of the user's attributes related to account recovery.

FIA_SOS.1

- **SF::User.Registration:**This security function handles the verification of the quality of the passphrase during users' registration.

FIA_UAU.2

- **SF::Server.Access.Control:**This security function implements access control on the TOE services.

FIA_UAU.3

- **SF::User.Registration:**This security function implements user authentication that is unforgeable.
- **SF::User.Login:**This security function implements user authentication that is unforgeable.

FCS_COP.1/zrtp

- **SF::Cryptographic.Operations:**This security function implements key agreement over Real-time Transport Protocol.

FDP_ACC.1/recovery

- **SF::Account.Recovery:**This security function handles access control on the user deposit for account recovery.

FDP_ACF.1/recovery

- **SF::Account.Recovery:**This security function handles access control on the user deposit for account recovery.

FMT_MSA.3/recovery

- **SF::Account.Backup:**This security function defines the corresponding user's *trustee*.

FMT_MSA.1/recovery

- **SF::Account.Recovery:**This security function implements the management rules of the access control on the user deposit for account recovery.

FMT_SMF.1/recovery

- **SF::Account.Backup:**This security function defines the user's *trustee*.

FAU_GEN.1

- **SF::Audit:**This security function handles the audit needs for the TOE.

FPT_STM.1

- **SF::Audit:**The audit logs use time stamps from the TOE.

FCS_CKM.1/random

- **SF::Cryptographic.Operations:**This security function implements cryptographic key generation.

FCS_CKM.1/kdf.pbkdf

- **SF::Cryptographic.Operations:**This security function implements cryptographic key derivation.

FCS_CKM.1/kdf.sha256

- **SF::Cryptographic.Operations:**This security function implements cryptographic key derivation.

FCS_CKM.4

- **SF::Cryptographic.Operations:**This security function implements cryptographic key destruction.

FIA_UID.2

- **SF::Server.Access.Control:**This security function implements access control on the TOE services.

INDEX

A

O

S

T