Technical Specification

Title	Shamirs Webservice
Author	Christof Reichardt Paul-Ehrlich-Weg 1 D-63110 Rodgau E-Mail: projektstudien@christofreichardt.de
Created on	2021-03-05
Version	1.0.0

1 Introduction

The software intends to provide several cryptographic services – such as the generation of symmetric and assymetric keys, the distribution of secret shares or the digital signature and encryption of electronic documents – by exposing an appropriate REST API. The cryptographic keys used for signing and encrypting remain on the server within PKCS12 keystores. Key entries within such PKCS12 keystores are typically encrypted as well, e.g. via password based encryption algorithms.

1.1 Terminology

ECDSA
: Elliptic Curve Digital Signature Algorithm.

Participant
: A user which has subscribed to a secret sharing sheme.

POST
: A method defined by the HTTP protocol.

PKCS12
: Defines a file format used to store cryptographical objects, e.g. private keys, within a single file.

REST
: Representational state transfer denotes an architectural style related to interactive applications using web services.

Secret Sharing
: Methods for distributing shares of a secret, e.g. a password, between certain participants. The participants need to combine a subset of the shares in able to recover the original secret.

2 Security Goals

2.1 Multiple-Eye Principle

It should be possible to enforce the multiple-eye principle when granting access to one of the keystores comprising the cryptographic keys. Consider, for example, a company signing key within such a keystore and an electronic payment order which must be reviewed and digitally signed. This can be achieved by applying a secret sharing algorithm in addition to the password based encryption of the keystores.

2.2 System Administration

Ideally, the system administration of the cryptographic services should not be able to gain access to the cryptographic keys even with root access to the server. This means that the passwords – or rather password shares – of the keystores should not remain on the server after their initial generation. Either the password shares will be distributed, e.g. via (encrypted) E-Mail messages, immediately after their creation or the clients fetch them via the REST API in due course. In order to open a session related to a certain keystore the appropriate shares must be again transmitted back to the server by the participants. After the closure of the session the key entries of the involved keystore will be re-encrypted and new fitting password shares will be distributed. That means that a seizure of the server alone should normally not lead to a security breach. Encrypted documents or private keys required for digital signatures remain safe because the related keystores cannot be loaded without the appropriate password shares.

3 Use Cases

3.1 Generation of Keystores

Participants may trigger the generation of PKCS12 keystores together with the desired key entries. Such key entries are always related to certain algorithms, e.g. ECDSA. The password needed to access the keystore will be splitted into shares and made available for the distribution between the denoted participants.

Preconditions

(1) The actor POSTing a set of instructions must be a registered user with the necessary authorizations.

(2) The denoted participants must be known to the system.

Postconditions

(1) The keystore together with the required key entries has been created.

(2) The related password shares have been computed and (optionally) distributed.

(3) A session with state PROVISIONED has been assigned to the keystore.

3.2 Import of the Participants

Appropriate information about the participants and actors must be imported to system. It is out of scope to maintain an independent identity management system. Instead the system relies on a separete OpenId Connect system for this purpose. In order to relate the participants referenced by the instructions to generate a certain keystore the system must maintain or buffer some information about the approved participants.

Preconditions

The participants together with their roles and corresponding permissions must be known to a OpenID Connect Provider.

Postconditions

The system has created the participant entities and is able to relate them to the users maintained by the OpenId Connect provider.

3.3 Session Activation

In order to encrypt or digitally sign electronic documents a session belonging to a certain keystore must be activated.

Preconditions

(1) The present session belonging to the given keystore has the state PROVIDED.

(2) A subset of password shares required to recover the password of the keystore is available.

(3) The actor that opens the session owns the required authorizations.

Postconditions

(1) The session belonging to the keystore has got the state ACTIVE.

(2) The session is ready to encrypt/sign documents by applying key entries of the related keystore.

3.4 Session Closure

3.4.1 Automatic Closure

During the session activation a maximum idle period had been transmitted. A concurrent thread checks periodically for idle sessions and closes them.

Preconditions

One ore more sessions have been inactive for longer than the denoted idle time.

Postconditions

(1) The idle sessions have been asssigned the state CLOSED and won't process documents anymore.

(2) The related keystores have been assigned new sessions with state PROVIDED.

(3)

3.4.2 On Demand

3.5 Digital Signature