MitSoft

Qualified Long-term Preservation Service

Preservation Profile With Storage

QLPS/PP-WST

Unique object ID (OID): **1.3.6.1.4.1.57890.1.5.1**

Document version 1.02

Valid since 2023-05-15

UAB „MIT-SOFT“, Kalvarijų st. 276-100, LT-08316 Vilnius, Tel. +370 233 3922, Fax +370 5 213 6191, www.mitsoft.lt, info@mitsoft.lt, Company ID 120792080, VAT payer code LT207920811

Approvals

Revision history

Version	Valid since	Description
1.00		First official version of the document
1.01		Fixed minor wording errors
1.02	2023-05-15	CMS signatures in PDF are permitted; clarification for non-valid digital signature preservation

Approval of the document

	Name	Date	Signature
Reviewed by	Adomas Birštunas	2023-04-04	
Approved by	Antanas Mitašiūnas	2023-05-15	

Table of content

1. Introduction

The joint stock company "MIT-SOFT" (further – the MitSoft) is a qualified long-term preservation service provider (further – PSP).

This document defines one of the preservation profiles supported by the MitSoft PSP. The name of the preservation profile defined in the current document is as follows:

- MitSoftQWST profile – MitSoft electronic document with storage preservation profile for qualified electronic signatures and seals.

MitSoftQWST profile should be used when qualified electronic signatures, advanced electronic signatures, qualified electronic seals or advanced electronic seals contained within electronic documents or containers should be preserved by the MitSoft PSP and stored within PSP storage.

Preservation storage model, preservation goals, operations supported by the MitSoftQWST profile and preservation protocol to be used are further described in this document.

2. References

[ADOC-V1.0] – Lietuvos vyriausiasis archyvaras. "Elektroniniu parašu pasirašyto elektroninio dokumento specifikacija ADOC-V1.0".

[ADOC-V2.0] – Lietuvos vyriausiasis archyvaras. "Elektroninio dokumento specifikacija ADOC-V2.0".

[EGAS-V1.0] – Valstybinio socialinio draudimo fondo valdybos prie Socialinės apsaugos ir darbo ministerijos direktoriaus įsakymas "Dėl elektroninės gyventojų aptarnavimo sistemos naudojimo taisyklių ir elektroninės gyventojų aptarnavimo sistemos elektroniniu parašu pasirašyto dokumento specifikacijos EGAS V1.0 patvirtinimo".

[eIDAS] – Regulation (EU) No 910/2014 of the European Parliament and of the Council on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC.

[EN 319 122-1] – ETSI EN 319 122-1: "Electronic Signatures and Infrastructures (ESI); CAdES digital signatures; Part 1: Building blocks and CAdES baseline signatures".

[EN 319 132-1] – ETSI EN 319 132-1: "Electronic Signatures and Infrastructures (ESI); XAdES digital signatures; Part 1: Building blocks and XAdES baseline signatures".

[EN 319 142-1] – ETSI EN 319 142-1: "Electronic Signatures and Infrastructures (ESI); PAdES digital signatures; Part 1: Building blocks and PAdES baseline signatures".

[EN 319 162-1] – ETSI EN 319 162-1: "Electronic Signatures and Infrastructures (ESI); Associated Signature Containers (ASiC); Part 1: Building blocks and ASiC baseline containers".

[MDOC-V1.0] – Lietuvos vyriausiasis archyvaras. "Elektroniniu parašu pasirašyto kompiuterio skaitomo elektroninio dokumento specifikacija MDOC-V1.0".

[PDF-LT-V1.0] – Lietuvos vyriausiasis archyvaras. "Elektroninio dokumento PDF-LT-V1.0 specifikacija".

[PDF-RC-V1.0] – Valstybės įmonė Registrų centras. „Elektroninio dokumento specifikacija PDF-RC-V1.0".

[DSS Core 2.0] - OASIS: "Digital Signature Service Core Protocols, Elements, and Bindings Version 2.0", Committee Specification 02, 11 December 2019.

[TS 101 733] – ETSI TS 101 733: "Electronic Signatures and Infrastructures (ESI); CMS Advanced Electronic Signatures (CAdES)".

[TS 101 903] – ETSI TS 101 903: "Electronic Signatures and Infrastructures (ESI); XML Advanced Electronic Signatures (XAdES)".

[TS 102 778-2] - ETSI TS 102 778-2: "Electronic Signatures and Infrastructures (ESI); PDF Advanced Electronic Signature Profiles; Part 2: PAdES Basic - Profile based on ISO 32000-1".

[TS 102 778-4] – ETSI TS 102 778-4: "Electronic Signatures and Infrastructures (ESI); PDF Advanced Electronic Signature Profiles; Part 4: PAdES Long Term - PAdES-LTV Profile".

[TS 103 171] – ETSI TS 103 171: "Electronic Signatures and Infrastructures (ESI); XAdES Baseline Profile".

[TS 103 172] – ETSI TS 103 172: "Electronic Signatures and Infrastructures (ESI); PAdES Baseline Profile".

[TS 103 173] – ETSI TS 103 173: "Electronic Signatures and Infrastructures (ESI); CAdES Baseline Profile".

[TS 103 174] – ETSI TS 103 174: "Electronic Signatures and Infrastructures (ESI); ASiC Baseline Profile".

[TS 119 512] – ETSI TS 119 512: "Electronic Signatures and Infrastructures (ESI); Protocols for trust service providers providing long-term data preservation services".

3. Definitions of terms and abbreviations

Compromise: a loss, theft, modification, illegal use, or any other security violation of the confidential data.

Container: data object, which contains a set of data objects and optional additional information, which describes the contained data objects and optionally its content and its interrelationships.

Data object: actual binary/octet data being operated on (e.g. transformed, digested, or signed) by an application and which may be associated with additional information like an identifier, the encoding, size or type.

Long-term: time period during which technological changes may be a concern.

Long-term preservation: extension of the validity status of a digital signature over long periods of time and/or extension of provision of proofs of existence of data over long periods of time, in spite of obsolescence of cryptographic technology such as crypto algorithms, key sizes or hash functions, key compromises or of the loss of the ability to check the validity status of public key certificates.

Preservation client: component or a piece of software which interacts with a preservation service via the preservation protocol.

Preservation evidence: evidence produced by the preservation service which can be used to demonstrate that one or more preservation goals are met for a given preservation object.

Preservation evidence policy: set of rules that specify the requirements and the internal process to generate or how to validate a preservation evidence.

Preservation goal: one of the following objectives achieved during the preservation time frame: extending over long periods of time the validity status of digital signatures, providing proofs of existence of data over long periods of time, or augmenting externally provided preservation evidences.

Preservation interface: component implementing the preservation protocol on the side of the preservation service.

Preservation object: typed data object which is submitted to, processed by or retrieved from a preservation service.

NOTE: This covers submission data objects, preservation object containers and preservation evidences.

Preservation object container: container which contains a set of data objects and optionally related metadata providing information about the data objects and optionally preservation manifest(s) specifying its content and relationships.

Preservation object identifier: unique identifier of a (set of) preservation object(s) submitted to a preservation service.

Preservation period: for a preservation service with storage, duration during which the preservation service preserves the submitted preservation objects and the associated evidences.

Preservation profile: uniquely identified set of implementation details pertinent to a preservation storage model and one or more preservation goals which specifies how preservation evidences are generated and validated.

Preservation protocol: protocol to communicate between the preservation service and a preservation client.

Preservation scheme: generic set of procedures and rules pertinent to a preservation storage model and one or more preservation goals which outlines how preservation evidences are created and validated.

NOTE: Different preservation profiles can implement the same preservation scheme.

Preservation service: service capable of extending the validity status of a digital signature over long periods of time and/or of providing proofs of existence of data over long periods of time.

Preservation service provider: trust service provider providing a preservation service.

Preservation storage model: one of the following ways of implementing a preservation service: with storage, with temporary storage, without storage.

Proof of existence: evidence that proves that an object existed at a specific date/time.

Signature augmentation: process of incorporating to an electronic signature or electronic seal information aiming to maintain the validity of that signature/seal over the near term and/or the long term.

Signature validation constraint: technical criteria against which an electronic signature or electronic seal can be validated.

Signature policy: signature creation policy, signature augmentation policy, signature validation policy or any combination thereof, applicable to the same signature or set of signatures.

Submission data object: original data object provided by the client.

Subscriber: legal or natural person bound by agreement with a preservation trust service provider to any subscriber obligations.

Other definitions are used as in Regulation (EU) 910/2014 [eIDAS].

ETSI	–	European Telecommunications Standards Institute
OID	–	Object identifier
PO	–	Preservation object
POC	–	Preservation object container
PSP	–	Preservation service provider
SubDO	–	Submission data object

4. MitSoft electronic document with storage preservation profile for qualified signatures and seals

4.1. Description

MitSoftQWST profile defines operational details for the MitSoft preservation service for the case it is used with integrated electronic document storage. Preservation objects are stored in PSP storage. The goal of the preservation service using MitSoftQWST profile is preservation over long periods of time of the ability to validate electronic signatures and electronic seals, maintenance their validity status and getting proofs of existence of the associated signed data.

Preservation objects to be preserved by preservation service are qualified electronic signatures and qualified electronic seals within signed/sealed electronic documents or containers. Preservation service preserves advanced electronic signatures and advanced electronic seals (that are not qualified) only if such a requirement is listed in the Subscriber agreement. Further in this document they are called electronic signatures or seals, or just digital signatures.

Preservation service preserves valid electronic signatures and valid electronic seals. Non-valid electronic signatures and non-valid electronic seals are preserved only if such a requirement is listed in the Subscriber agreement and it is allowed according signature policy.

Preservation service does not accept electronic signatures or electronic seals that are not part of some submitted electronic document or container. Every submitted electronic document or container shall contain actual data signed by the preserved electronic signature or sealed by the preserved electronic seal (subscribers are not allowed to provide only hash values of the signed data). The supported electronic document specifications and container standards are listed in the section 4.12. Preservation objects to be preserved by this profile are defined in the section 4.8.

This profile should be used when signed/sealed electronic documents/containers are preserved and stored by the MitSoft preservation service. Preservation service is responsible for maintenance and preservation of the electronic signatures and electronic seals present within stored electronic documents/containers during preservation period.

MitSoftQWST profile offers operations for:

- electronic document/container uploading to MitSoft preservation service and storing it within MitSoft preservation service storage (see for Store, PreservePO operations),
- electronic document/container downloading from MitSoft preservation service storage (see for Download, RetrievePO operations),
- electronic document/container removing from MitSoft preservation service storage (see for Remove, DeletePO operations),
- retrieving preserving electronic document/container current status information (see for Status and RetrieveTrace operation).

Preservation of the electronic signatures and electronic seals within stored electronic documents/containers are performed by the MitSoft preservation service automatically in a timely manner.

4.2. Identification

The unique identifier (OID) of the MitSoftQWST profile is as follows:

- identifier presented as OID is
  - 1.3.6.1.4.1.57890.1.5.1;

  the values of its fields are given in the Table 1.

- Identifier presented as URI is
  - `http://uri.mitsoft.lt/preservation/profile/qwst/1`

Table 1. The values of the fields of the unique identifier of the MitSoftQWST profile

Name	Value
ISO	1
Organization recognized by ISO	3
U.S. Department of Defence	6
Internet	1
Private company	4
Private company registered by IANA	1
Joint stock company "MIT-SOFT"	57890
Subdivision MitSoft	1
Document type (preservation profile)	5
Preservation profile identifier	1

4.3. Validity period

The present preservation profile becomes active from 2022-12-01. The same preservation profile is applied during the whole preservation period.

4.4. Preservation scheme

The MitSoftQWST profile implements the following preservation scheme:

- preservation scheme with signature augmentation and with storage, which is defined in Annex F.3 of ETSI TS 119 512 [TS 119 512] and indicated by the identifier:
  - `http://uri.etsi.org/19512/scheme/pds+wst+aug`

4.5. Preservation goal

The MitSoftQWST profile supports the following preservation goal:

- PDS - extending over long periods of time the validity status of digital signatures ETSI TS 119 512 [TS 119 512], which is indicated by the URI:
  - `http://uri.etsi.org/19512/goal/pds`

4.6. Preservation storage model

The MitSoftQWST profile supports preservation service with storage – WTS according to ETSI TS 119 512 [TS 119 512].

4.7. Supported operations

The present preservation profile supports the following operations:

- RetrieveInfo,
- Store,
- Status,

- Download,
- Remove,
- PreservePO,
- RetrievePO,
- DeletePO,
- RetrieveTrace,
- Search.

4.8. Submission data objects and preservation objects

MitSoftQWST preserves electronic signatures and electronic seals contained in some electronic document or container.

Submission data object (SubDO) is an electronic document/container, presented as one file, which:

- contains at least one electronic signature or electronic seal,
- contains every data object signed by electronic signature or sealed by electronic seal, and
- optionally contains its own metadata.

Supported SubDOs are defined in the section 4.12.

Preservation object (PO) preserved by the MitSoft PSP is an electronic signature or an electronic seal of the B-LTA signature level (or corresponding archival signature format for non-baseline signatures).

Preservation service preserves qualified electronic signatures and qualified electronic seals. Preservation service preserves advanced electronic signatures and advanced electronic seals (that are not qualified) only if such a requirement is listed in the Subscriber agreement.

Preservation service preserves valid electronic signatures and valid electronic seals (having validation status PASSED). Non-valid electronic signatures and non-valid electronic seals (having validation status other than PASSED) are preserved only if validation result does not contain any validation error, which prevents digital signature from the augmentation according signature policy, and if such a requirement is listed in the Subscriber agreement. The precise list of validation error subindications, that do not prevent digital signature from the augmentation, is listed in the signature policy.

Preservation object preserved by the MitSoft PSP is derived from the submission data object by augmenting electronic signatures and electronic seals presented within SubDO. Augmentation is performed to reach B-LTA signature level (or corresponding archival signature format for non-baseline signatures). Augmentation involves getting validation data and time stamps, and their inclusion into digital signature. Preservation of the electronic signature or electronic seal is not performed in the case B-LTA signature level (or corresponding archival signature format for non-baseline signatures) cannot be reached (caused by its invalidity, or by impossibility to get reliable missing validation data).

Accepted submission data objects that do not contain any electronic signature or electronic seal to be preserved are stored within PSP storage, but no preservation actions are performed with it. Such a SubDOs may be removed from the MitSoft PSP document storage by the client request using corresponding operations (Remove, DeletePO).

Preservation of electronic signatures and electronic seals is assured by the new archival time stamp inclusion. PSP preservation mechanism ensures, that new preservation evidence (archival time stamp) will be included in the case the last archival time stamp signing certificate to be expired, or in the case some cryptographic algorithm become not enough secure to ensure preservation evidence reliability in the near future.

New version of the preservation evidence policy may be issued and applied if current preservation event policy can no longer ensure required level of reliability.

In the case if preservation of electronic signatures and electronic seals cannot be successfully performed due to unavailable validation data and/or time stamps, augmentation attempts will be repeated by the preservation service for a reasonable period of time.

In the context of electronic documents, digital signature usage (and preservation) without electronic document/container it belongs to (and actually signed) is meaningless. Therefore, every operation deals with preservation object container instead of separate preservation object preserved by the MitSoft PSP.

Preservation object container (POC) is a submitted electronic document or container, which contains electronic signatures and/or seals (preservation objects) with preservation evidences (time stamps) included by the preservation service. POC is derived from the SubDO, by augmenting preserving digital signatures and preservation evidences (time stamps) inclusion. In the special case, when there are no preserving digital signatures in the SubDO, POC may be the same unmodified SubDO.

Operations defined by this profile accepts and returns preservation object, which is a whole preservation object container - electronic document or container (together with all preserving digital signatures and preservation evidences within it), or submission data object. Separate electronic signature or seal cannot be accessed using this profile operations.

Preservation object identifier identifies one electronic document/container (POC) and every preserving digital signature within it.

4.9. Preservation protocol

The present section describes semantics and syntax of the supported operations. Operations are implemented as REST web services. Web service caller authentication shall be used and communication encryption (Secure Sockets Layer) shall be applied.

Operations originally defined in the ETSI TS 119 512 are shortly introduced in this section using references to the standard and specifying specific issues only. Other specific operations semantics and syntax are defined in this section. Only the main fields are described, field names are unformal, types and field cardinalities are omitted. The exact preservation interface - requests and responses fields - is described separately using JSON notation (see section 4.9.11).

MitSoftQWST profile supports the following operations:

4.9.1. RetrieveInfo

RetrieveInfo operation is used to get the list of the supported preservation profiles.

RetrieveInfo operation syntax and semantics are fully described in the ETSI TS 119 512 section 5.3.2. Current profile supports this operation only used with JSON syntax.

4.9.2. Store

Store operation is an extended analogue of the PreservePO operation defined in the ETSI TS 119 512. Store operation is used to submit submission data object (SubDO) to the preservation service. For this operation submission data object shall be provided. Optional additional information may be also provided for this operation. Preservation service is responsible to store provided submission data object and to preserve preservation objects derived from the SubDO by electronic signatures/seals augmentation. Details on the preservation objects and how they are derived from the SubDO are described in the section 4.8. Preservation service stores additional data as is

and preservation service is not responsible to ensure provision of proofs of existence for such a data.

This operation supports submission data objects of the formats listed in the section 4.12.

Current profile supports this operation used with JSON syntax. JSON schema file and OpenAPI Document are presented in the section 4.9.11.

4.9.2.1 Store Request

The *Store* request shall accept the submission data object content and additional parameters.

The request shall include the following fields:

- The *client ID*. It shall contain the identifier of the preservation client which performs the call to this operation.
- The *document file*. It shall contain the content of SubDO submitted as a file. The content shall be provided in its original format without encoding or transformation.

The request may contain the following fields:

- The optional *document specification*. If present, it shall contain the identifier of document specification defining the type of SubDO content. The identifier represents one of specifications listed in section 4.12. If this parameter is omitted, then the SubDO type shall be determined from its file name extension or content.
- The optional *external identifier*. If present, it shall contain identifier associated with SubDO by preservation client. The identifier value shall be unique among SubDOs of this preservation client.

  If another PO with provided identifier is already stored in preservation service for this client, then it is replaced by submitted SubDO.
- The optional *preserve till*. If present, it shall contain date till which the preservation object derived from the SubDO should be preserved in preservation service. If it is omitted or its value is *null*, then preservation object will be preserved indefinitely.

The request may contain additional fields providing metadata associated to the submission data object and stored in preservation service (preservation object title, sort, date, number, etc.). See JSON syntax file for details (section 4.9.11).

The fields above shall be implemented as the following HTTP multi-part POST request parameters:

Fields	Description	HTTP request parameters
Client ID	Identifier of the preservation client	cid
Document file	The content of submission data object	docFile
Document specification	Identifier of specification defining the type of preserver object content	docSpecId
External identifier	Identifier used to identify SubDO in preservation client system	extCode
Preserve till	The date till which preservation object should be preserved	maintainableUntil

4.9.2.2 Store Response

The Store response contains status of the operation execution, whether the operation was successful or not, and errors in the case of failure.

On failure, HTTP response with status different from 200 and content with error message is returned.

On success, HTTP response with status code 200 is returned. The response shall contain the following elements:

- The *operation status* element. It shall contain the value "success" indicating the success of the operation.
- The *external identifier* element. It shall contain the preservation object identifier in preservation service. If *external identifier* was provided in operation request by preservation client, then the same value will be returned. If external identifier was omitted or its value was *null*, then a new identifier value created by preservation service will be returned. It shall be used by preservation client to reference preservation object in future calls to other preservation service operations.

The Store response JSON object shall be defined as in JSON Schema file (signa-arch-api-schema.json) provided in section 4.9.11.1. The elements of JSON Schema shall implement elements of Store response mapped by names as show in the following table:

Element	Description	JSON member name
Operation status	The status of store operation execution	status
External identifier	Preservation object identifier stored in preservation service	extCode

4.9.3. Status

Status operation is an extended analogue of the RetrieveTrace operation defined in the ETSI TS 119 512. Status operation is used to retrieve information about the stored preservation object (electronic document/container), but not the preservation object itself. For this operation preservation object identifier shall be provided. Operation returns current status of the previously submitted electronic document/container, main data objects within it and also validity status of the electronic signatures and electronic seals within it. Operation shall return status information only for preservation objects that are accessible by the preservation client.

Current profile supports this operation used with JSON syntax. JSON schema file and OpenAPI Document are presented in the section 4.9.11.

4.9.3.1 Status Request

The Status request shall include the following fields:

- The *client ID*. It shall contain the identifier of the preservation client which performs the call to this operation.
- The *identifier*. It shall reference the preserved object of preservation client previously stored in preservation service (see *store* operation and its *external identifier* field).

The request may include the following fields:

- The optional *mode*. If present, it shall indicate the amount of data preservation client needs to receive about preservation object. It may

indicate only to check preservation object presence in preservation service, get more details on its state in preservation service, or retrieve information on preserved object content and digital signature validation. See JSON syntax file for details (section 4.9.11).

The fields above shall be implemented as the following HTTP GET request parameters:

Field	Description	JSON member name
Client Id	Identifier of the client	cid
Identifier	Preservation object identifier	extCode
Mode	Mode, which describes the amount of the data to be returned	mode

4.9.3.2 Status Response

Response contains status of the operation execution, whether the operation was successful or not, and errors in the case of failure.

On failure, HTTP response with status different from 200 and content with error message is returned.

On success, HTTP response with status code 200 is returned and it contains status information about the preservation object.

The response shall contain the following elements:

- The *status* element. It shall indicate if preservation object is present in preservation service, it is valid or has errors.
- The *document identifier* element. It shall contain the preservation object identifier.

Depending on the *mode* indicated in Status request, the response may contain the following additional elements:

- The *document specification* element. If present, it shall contain the identifier of document specification defining the type of submission data object content. The identifier represents one of specifications listed in section 4.12.
- The *maintainable until* element. If present, it shall indicate the date until which the preservation object is preserved by the preservation service.
- The *next update date* element. If present, it shall contain date and time of next moment when preservation system will perform the analysis of preservation object and its digital signatures.
- The *last augment date* element. If present, it shall contain date and time when augmentation of preservation object digital signatures was performed in preservation service for the last time.
- The *last augment status* element. If present, it shall indicate the status of the last operation, that augmented preservation object's digital signatures.
- The *next augment date* element. If present, it shall contain date and time when the next augmentation or preservation object and its digital signatures is planned to be performed by preservation service.
- The *events* element. If present, it shall contain the list of actions that have been performed on preservation object by preservation client or preservation service.
- The *structure* element. If present, it shall contain the preservation object structure validation status and list of error messages, if detected.

- The *packaging* element. If present, it shall contain the preservation object (ex., electronic document file) package validation status and list of error messages, if detected.
- The *signature* element. If present, it shall contain the list of digital signatures of preservation object. An element in the list shall contain a digital signature validation status, validation errors (if detected) and information on digital signature and signer certificate.
- The *content* element. If present, it shall contain a list of POC content entries (signed and/or unsigned data files stored within it), their validation status and error messages, if detected.
- The *metadata* element. If present, it shall contain a list of SubDO metadata, their validation status and error messages, if detected.

The Status response JSON object shall be defined as in JSON Schema file (signa-arch-api-schema.json) provided in section 4.9.11.1. The elements of JSON Schema shall implement elements of Status response mapped as show in the following table:

Field	Description	JSON member name
Status	The general status of the preservation object in preservation service.	status
Document identifier	Preservation object identifier in preservation service	extCode
Document specification	Preservation object type identifier	docSpecId
Maintainable until	The date until which preservation object is preserved by preservation service	maintainableUntil
Next update date	Time when preservation service will update information on preservation object	nextUpdateDate
Last augment date	The date and time when last time preservation object was augmented by preservation service	lastAugmentDate
Last augment status	The code indicating the status of last augmentation of preservation object	lastAugmentStatus
Next augment date	Time when preservation service will augment preservation object	nextAugmentDate
Events	List of actions performed on preservation object by preservation service or preservation client	events
Structure	Validation status of preservation object structure	structure
Packaging	Validation status of preservation object packaging	packaging
Signatures	The list of preservation object digital signatures with digital signature information and validation status	signatures

Content	The list of POC content (files) entries and validation status	content
Metadata	The list of POC metadata and validation status	metadata

4.9.4. Download

Download operation is an extended analogue of the RetrievePO operation defined in the ETSI TS 119 512. Download operation is used to retrieve stored preservation object together with collected preservation evidences within it. For this operation preservation object identifier shall be provided. Returned preservation object format should be one of the formats listed in the section 4.12. Operation shall return only preservation objects that are accessible by the preservation client.

Current profile supports this operation used with JSON syntax. JSON schema file and OpenAPI Document are presented in the section 4.9.11.

4.9.4.1 Download Request

The Download request shall include the following fields:

- The *client ID*. It shall contain the identifier of the preservation client that performs the call to this operation.
- The *identifier*. It shall reference the preserved object in preservation service. This preservation object is previously stored in preservation service by the same preservation client.

The fields above shall be implemented as the following HTTP GET request parameters:

Field	Description	JSON member name
Client ID	Identifier of the preservation client	cid
Identifier	Preservation object identifier in preservation service	extCode

4.9.4.2 Download Response

The Download operation response is returned as HTTP response with status code 200 and the content of preservation object. The content of preservation object is returned without encoding or transformation.

In case of a failure, the HTTP response has status code different from 200 and content with error message.

4.9.5. Remove

Remove operation is an extended analogue of the DeletePO operation defined in the ETSI TS 119 512. Remove operation is used to delete preservation object form the PSP's storage. For this operation preservation object identifier shall be provided.

When preservation object is removed, the corresponding SubDO and preservation evidences are deleted as well. The traces of the performed preservation actions will not be deleted. Operation shall allow to delete only preservation objects that are accessible by the preservation client.

Current profile supports this operation used with JSON syntax. JSON schema file and OpenAPI Document are presented in the section 4.9.11.

4.9.5.1 Remove Request

The Remove request shall include the following fields:

- The *client ID*. It shall contain the identifier of the preservation client that performs the call to this operation.
- The *identifier*. It shall reference the preserved object to be removed in preservation service.
- The *actor name*. It indicates the name of the claimed requestor of the removal of this preservation object. It is mandatory if the removal of the preservation object is requested before the end of its preservation period.
- The *reason*. It indicates the reason of the removal of this preservation object. It is mandatory if the removal of the preservation object is requested before the end of its preservation period.

The fields above shall be implemented as the following HTTP GET request parameters:

Field	Description	JSON member name
Client ID	Identifier of the preservation client	cid
Identifier	Preservation object identifier to be removed	extCode
Actor name	Claimed name of the requestor of this operation	actorName
Reason	Reason for the removal of the preservation object	reason

4.9.5.2 Remove Response

The Remove response contains status of the operation execution, whether the operation was successful or not, and errors in the case of failure.

On failure, the HTTP response has status code different from 200 and content with error message.

On success, HTTP response with status code 200 is returned.

The response shall contain the following elements:

- The *operation status* element. It shall contain the value "success" indicating the success of the operation.
- The *external identifier* element. It shall contain the identifier of removed preservation object.

The Remove response JSON object shall be defined as in JSON Schema file (signa-arch-api-schema.json) provided in section 4.9.11.1. The elements of JSON Schema shall implement elements of Remove response mapped by names as show in the following table:

Element	Description	JSON member name
Operation status	The status of remove operation execution	status
External identifier	Identifier of the removed preservation object	extCode

4.9.6. PreservePO

PreservePO operation is used to submit submission data object (SubDO) to the preservation service. For this operation submission data object shall be provided. Preservation service is responsible to store provided submission data object and to preserve preservation object derived from the SubDO by electronic signatures/seals augmentation. Details on the preservation objects and how they are derived from the SubDO are described in the section 4.8.

This operation supports submission data objects of the formats listed in the section 4.12.

PreservePO operation syntax and semantics are described in the ETSI TS 119 512 section 5.3.3. Current profile supports this operation only used with JSON syntax.

PreservePO operation shall accept the following elements from *OptionalInputs* element, as defined in section 4.2.8 of OASIS DSS-X Core 2.0 [DSS Core 2.0]:

- The optional *lang* element. If provided, it shall contain the code of the language to be used for preservation service messages.
- The *other* element. It shall contain an array with single element with attributes:
  - The *ID* attribute. It shall have fixed value "cid".
  - The *value* attribute. It shall have Base64 encoded preservation client identifier value.

PreservePO operation shall be available to preservation clients that can be identified by the preservation service. To allow for the preservation client identification, PreservePO operation request shall include *client identifier* with the help of *other* element, as described above.

4.9.7. RetrievePO

RetrievePO operation is used to retrieve stored preservation object together with collected preservation evidences within it. For this operation preservation object identifier shall be provided.

Returned preservation object format should be one of the formats listed in the section 4.12. Operation shall return only preservation objects that are accessible by the client.

RetrievePO operation syntax and semantics are described in the ETSI TS 119 512 section 5.3.4. Current profile supports this operation only used with JSON syntax.

RetrievePO operation shall accept the following elements from *OptionalInputs* element, as defined in section 4.2.8 of OASIS DSS-X Core 2.0 [DSS Core 2.0]:

- The optional *lang* element. If provided, it shall contain the code of the language to be used for preservation service messages.
- The *other* element. It shall contain an array with single element with attributes:
  - The *ID* attribute. It shall have fixed value "cid".
  - The *value* attribute. It shall have Base64 encoded preservation client identifier value.

RetrievePO operation supports the following elements as defined in section 5.3.4 of ETSI TS 119 512 with the following considerations:

- The POID element. It shall be present and it shall contain the identifier of preservation object to be retrieved.
- The SubjectOfRetrieval element. If present, it shall contain the value "POwithEmbeddedEvidence".
- The POFormat element. If present, it shall contain the URI identifying the preservation object format and will correspond to effective format of the

preservation object preserved in the preservation service and retrieved with this operation.

- The EvidenceFormat element. It shall be omitted. If it is included, then the operation shall return the error message indicating unsupported element.

RetrievePO operation shall be available to preservation clients that can be identified by the preservation service. To allow for the preservation client identification, RetrievePO operation request shall include *client identifier* with the help of *other* element, as described above.

4.9.8. DeletePO

DeletePO operation is used to delete preservation object form the PSP's storage. For this operation preservation object identifier shall be provided. Preservation object will be deleted together with corresponding submission data object, preservation evidences and all corresponding metadata. The traces of the performed preservation actions will not be deleted. Operation shall allow to delete only preservation objects that are accessible by the client.

DeletePO operation syntax and semantics are described in the ETSI TS 119 512 section 5.3.5. Current profile supports this operation only used with JSON syntax.

DeletePO operation shall accept the following elements from *OptionalInputs* element, as defined in section 4.2.8 of OASIS DSS-X Core 2.0 [DSS Core 2.0]:

- The optional *lang* element. If provided, it shall contain the code of the language to be used for preservation service messages.
- The *other* element. It shall contain an array with single element with attributes:
  - The *ID* attribute. It shall have fixed value "cid".
  - The *value* attribute. It shall have Base64 encoded preservation client identifier value.

DeletePO operation shall be available to preservation clients that can be identified by the preservation service. To allow for the preservation client identification, DeletePO operation request shall include *client identifier* with the help of *other* element, as described above.

4.9.9. RetrieveTrace

RetrieveTrace operation is used to retrieve preservation actions performed with the stored preservation object. For this operation preservation object identifier shall be provided.

Operation shall return information only about the preservation objects that are accessible by the client.

RetrieveTrace operation syntax and semantics are described in the ETSI TS 119 512 section 5.3.7. Current profile supports this operation only used with JSON syntax.

RetrieveTrace operation shall accept the following elements from *OptionalInputs* element, as defined in section 4.2.8 of OASIS DSS-X Core 2.0 [DSS Core 2.0]:

- The optional *lang* element. If provided, it shall contain the code of the language to be used for preservation service messages.
- The *other* element. It shall contain an array with single element with attributes:
  - The *ID* attribute. It shall have fixed value "cid".
  - The *value* attribute. It shall have Base64 encoded preservation client identifier value.

RetrieveTrace operation shall be available to preservation clients that can be identified by the preservation service. To allow for the preservation client identification, RetrieveTrace operation request shall include *client identifier* with the help of *other* element, as described above.

4.9.10. Search

Search operation is used to search among the set of the preservation objects stored in the PSP's storage. For this operation search filters for the preservation objects should be provided. Operation returns list of identifiers of the preservation objects.

Operation shall return identifiers only for preservation objects that are accessible by the client.

Search operation syntax and semantics are described in the ETSI TS 119 512 section 5.3.9. Current profile supports this operation only used with JSON syntax.

Search operation shall accept the following elements from *OptionalInputs* element, as defined in section 4.2.8 of OASIS DSS-X Core 2.0 [DSS Core 2.0]:

- The optional *lang* element. If provided, it shall contain the code of the language to be used for preservation service messages.
- The *other* element. It shall contain an array with single element with attributes:
  - The *ID* attribute. It shall have fixed value "cid".
  - The *value* attribute. It shall have Base64 encoded preservation client identifier value.

Search operation shall be available to preservation clients that can be identified by the preservation service. To allow for the preservation client identification, Search operation request shall include *client identifier* with the help of *other* element, as described above.

4.9.11. JSON Schemas and OpenAPI Documents

4.9.11.1 JSON Schema files

The JSON schema definitions for the Store, Status, Download and Remove operations are presented at:

- https://qtsp.mitsoft.lt/repo/qlps/arch-protocol/v1/signa-arch-api-schema.json

The JSON schema definitions for the RetrieveInfo, PreservePO, RetrievePO, DeletePO, RetrieveTrace and Search operations is presented at:

- https://qtsp.mitsoft.lt/repo/qlps/preserv-protocol/v1/signa-19512-preservation-api-schema.json

4.9.11.2 OpenAPI specifications

The OpenAPI specification for the Store, Status, Download and Remove operations is presented at:

- https://qtsp.mitsoft.lt/repo/qlps/arch-protocol/v1/signa-arch-api-openapi.json

The OpenAPI specification for the RetrieveInfo, PreservePO, RetrievePO, DeletePO, RetrieveTrace and Search operations is presented at:

- https://qtsp.mitsoft.lt/repo/qlps/preserv-protocol/v1/signa-19512-preservation-api-openapi.json

4.10. Preservation protocol usage guidelines

Preservation client could choose to remove submission data object (SubDO) from its own storage once it is stored to the preservation service with storage. In order to avoid any loses of data it is recommended for the preservation client to keep SubDO in its storage for some period of time after submission to preservation service, before it is safely stored to the preservation service storage.

Subscriber shall use the preservation operations according to the following guidelines:

- Preservation client submits the SubDO to the preservation service (*store*, *PreservePO* operations).
  - If preservation client submits SubDO using *store* operation, then it can indicate its preservation period.
  - If no preservation period is indicated or SubDO is submitted using *PreservePO* operation, then preservation period is unlimited.
  - Preservation of SubDO and augmentation of electronic signatures and electronic seals is not performed after the preservation period. Preservation client should indicate the preservation period only if is known that the preservation of SubDO is not necessary after the indicated period.
- Preservation client keeps the original copy of SubDO in its storage for some period, which is indicated in subscriber agreement.
- After this period, preservation client checks the status of the SubDO in preservation service (*status*, *RetrieveTrace* operations). If the SubDO is present in the preservation service, then preservation client can remove it from its own storage.
- If preservation client submits SubDO with identifier corresponding to a SubDO previously submitted to preservation service, then it replaces the previous SubDO. Previous SubDO with collected evidence data is lost. To avoid loss of data during SubDO update, preservation client should:
  - Retrieve SubDO from preservation service, before updating it on its system (for example, adding an electronic signature).
  - Store the updated SubDO to preservation service with the same identifier.

4.11. Applicable policies

The MitSoftQWST profile supports the following preservation evidence policy:

- MitSoft Qualified Long-term Preservation Service Preservation Evidence Policy, which is indicated by the unique object identifier (OID):
  - 1.3.6.1.4.1.57890.1.7.1.X

  where x stands for the latest version. All versions are available for the subscribers on the repository of MitSoft PSP. Each version contains the point in time from which on this version of the policy has become or will become active. The validity period of the version ends when new version becomes active.

The MitSoftQWST profile supports the following signature policy:

- MitSoft Qualified Long-term Preservation Service Signature Policy, which is indicated by the unique object identifier (OID):
  - 1.3.6.1.4.1.57890.1.6.1.X

  where x stands for the latest version. All versions are available for the subscribers on the repository of MitSoft PSP. Each version contains the point in time from which on this version of the policy has become or will become

active. The validity period of the version ends when new version becomes active.

4.12. Supported submission data objects

Preservation service accepts electronic documents or containers that are signed with electronic signatures or sealed with electronic seals. Every submitted electronic document or container shall contain actual data signed by the electronic signature or sealed by the electronic seal.

Every submission data object shall contain at least one electronic signature or electronic seal.

Submission data object formats supported by this preservation profile are defined in the following table:

Submission data object format (electronic document specification or container standard)	Electronic signature/seal format	Submission data object (specification) identifier
ADOC-V1.0 electronic document	XAdES	ADOC-V1.0
ADOC-V2.0 electronic document	XAdES baseline	ADOC-V2.0
EGAS-V1.0 electronic document	XAdES	EGAS-V1.0
MDOC-V1.0 electronic document	XAdES	MDOC-V1.0
PDF-LT-V1.0 electronic document	PAdES baseline	PDF-LT-V1.0
PDF-RC-V1.0 electronic document	PAdES baseline	PDF-RC-V1.0
ASiC-E container according to ETSI TS 103 174	XAdES baseline	ASiC-E-XAdES-TS
ASiC-E container according to ETSI EN 319 162-1	XAdES baseline	ASiC-E-XAdES-EN
ASiC-S container according to ETSI TS 103 174	XAdES baseline	ASiC-S-XAdES-TS
	CAdES baseline	ASiC-S-CAdES-TS
ASiC-S container according to ETSI EN 319 162-1	XAdES baseline	ASiC-S-XAdES-EN
	CAdES baseline	ASiC-S-CAdES-EN
PDF document with PAdES signatures according to ETSI TS 103 172	PAdES baseline	PDF-PAdES-TS
PDF document with PAdES signatures according to ETSI EN 319 142-1	PAdES baseline	PDF-PAdES-EN
PDF document with CMS signatures	PAdES	PDF-PAdES-CMS

This set of supported submission data object formats for particular subscriber may be reduced by the Subscriber agreement between this preservation service subscriber and MitSoft PSP.

4.12.1. ADOC-V1.0 electronic documents

ADOC-V1.0 electronic document [ADOC-V1.0], which contains at least one XAdES electronic signature or electronic seal conformant to XAdES standard ETSI TS 101 903 [TS 101 903].

This submission data object is identified by the following identifier:

- ADOC-V1.0

This submission data object is identified by the following URI identifier:

- http://uri.mitsoft.lt/specification/type/adoc-v1.0

The MIME type of this submission data object file is as follows:

- application/vnd.lt.archyvai.adoc-2008

4.12.2. ADOC-V2.0 electronic documents

ADOC-V2.0 electronic document [ADOC-V2.0], which contains at least one XAdES electronic signature or electronic seal conformant to XAdES baseline profile standard ETSI TS 103 171 [TS 103 171].

This submission data object is identified by the following identifier:

- ADOC-V2.0

This submission data object is identified by the following URI identifier:

- http://uri.mitsoft.lt/specification/type/adoc-v2.0

The MIME type of this submission data object file is as follows:

- application/vnd.etsi.asic-e+zip

4.12.3. EGAS-V1.0 electronic documents

EGAS-V1.0 electronic document [EGAS-V1.0], which contains at least one XAdES electronic signature or electronic seal conformant to XAdES standard ETSI TS 101 903 [TS 101 903].

This submission data object is identified by the following identifier:

- EGAS-V1.0

This submission data object is identified by the following URI identifier:

- http://uri.mitsoft.lt/specification/type/egas-v1.0

The MIME type of this submission data object file is as follows:

- application/vnd.lt.sodra.egas-2009

4.12.4. MDOC-V1.0 electronic documents

MDOC-V1.0 electronic document [MDOC-V1.0], which contains at least one XAdES electronic signature or electronic seal conformant to XAdES standard ETSI TS 101 903 [TS 101 903].

This submission data object is identified by the following identifier:

- MDOC-V1.0

This submission data object is identified by the following URI identifier:

- http://uri.mitsoft.lt/specification/type/mdoc-v1.0

The MIME type of this submission data object file is as follows:

- application/vnd.lt.archyvai.mdoc-2010

4.12.5. PDF-LT-V1.0 electronic documents

PDF-LT-V1.0 electronic document [PDF-LT-V1.0], which contains at least one PAdES electronic signature or electronic seal conformant to PAdES baseline profile standard ETSI TS 103 172 [TS 103 172].

This submission data object is identified by the following identifier:

- PDF-LT-V1.0

This submission data object is identified by the following URI identifier:

- `http://uri.mitsoft.lt/specification/type/pdf-lt-v1.0`

The MIME type of this submission data object file is as follows:

- `application/pdf`

4.12.6. PDF-RC-V1.0 electronic documents

PDF-RC-V1.0 electronic document [PDF-RC-V1.0], which contains at least one PAdES electronic signature or electronic seal conformant to PAdES baseline profile standard ETSI EN 319 142-1 [EN 319 142-1].

This submission data object is identified by the following identifier:

- `PDF-RC-V1.0`

This submission data object is identified by the following URI identifier:

- `http://uri.mitsoft.lt/specification/type/pdf-rc-v1.0`

The MIME type of this submission data object file is as follows:

- `application/pdf`

4.12.7. ASiC-E container according to ETSI TS 103 174

ASiC-E container which is conformant to ASiC baseline profile standard [TS 103 174] and which contains at least one XAdES electronic signature or electronic seal conformant to XAdES baseline profile standard ETSI TS 103 171 [TS 103 171].

This submission data object is identified by the following identifier:

- `ASiC-E-XAdES-TS`

This submission data object is identified by the following URI identifier:

- `http://uri.mitsoft.lt/ades/type/asic-e/xades/ts`

The MIME type of this submission data object file is as follows:

- `application/vnd.etsi.asic-e+zip`

4.12.8. ASiC-E container according to ETSI EN 319 162-1

ASiC-E container which is conformant to Building blocks and ASiC baseline container standard ETSI EN 319 162-1 [EN 319 162-1] and which contains at least one XAdES electronic signature or electronic seal conformant to Building blocks and XAdES baseline signatures standard ETSI EN 319132-1 [EN 319 132-1].

This submission data object is identified by the following identifier:

- `ASiC-E-XAdES-EN`

This submission data object is identified by the following URI identifier:

- `http://uri.mitsoft.lt/ades/type/asic-e/xades/en`

The MIME type of this submission data object file is as follows:

- `application/vnd.etsi.asic-e+zip`

4.12.9. ASiC-S container with XAdES signatures according to ETSI TS 103 174

ASiC-S container which is conformant to ASiC baseline profile standard ETSI TS 103 174 [TS 103 174] and which contains electronic signature or electronic seal conformant to XAdES baseline profile according ETSI TS 103 171 [TS 103 171].

This submission data object is identified by the following identifier:

- `ASiC-S-XAdES-TS`

This submission data object is identified by the following URI identifier:

- `http://uri.mitsoft.lt/ades/type/asic-s/xades/ts`

The MIME type of this submission data object file is as follows:

- `application/vnd.etsi.asic-s+zip`

4.12.10. ASiC-S container with CAdES signatures according to ETSI TS 103 174

ASiC-S container which is conformant to ASiC baseline profile standard ETSI TS 103 174 [TS 103 174] and which contains electronic signature or electronic seal conformant to CAdES baseline profile according ETSI TS 103 173 [TS 103 173].

This submission data object is identified by the following identifier:

- `ASiC-S-CAdES-TS`

This submission data object is identified by the following URI identifier:

- `http://uri.mitsoft.lt/ades/type/asic-s/cades/ts`

The MIME type of this submission data object file is as follows:

- `application/vnd.etsi.asic-s+zip`

4.12.11. ASiC-S container with XAdES signatures according to ETSI EN 319 162-1

ASiC-S container which is conformant to Building blocks and ASiC baseline container standard ETSI EN 319 162-1 [EN 319 162-1] and which contains electronic signature or electronic seal conformant to Building blocks and XAdES baseline signatures standard ETSI EN 319 132-1 [EN 319 132-1].

This submission data object is identified by the following identifier:

- `ASiC-S-XAdES-EN`

This submission data object is identified by the following URI identifier:

- `http://uri.mitsoft.lt/ades/type/asic-s/xades/en`

The MIME type of this submission data object file is as follows:

- `application/vnd.etsi.asic-s+zip`

4.12.12. ASiC-S container with CAdES signatures according to ETSI EN 319 162-1

ASiC-S container which is conformant to Building blocks and ASiC baseline container standard ETSI EN 319 162-1 [EN 319 162-1] and which contains electronic signature or electronic seal conformant to Building blocks and CAdES baseline signatures standard ETSI EN 319 122-1 [EN 319 122-1].

This submission data object is identified by the following identifier:

- `ASiC-S-CAdES-EN`

This submission data object is identified by the following URI identifier:

- `http://uri.mitsoft.lt/ades/type/asic-s/cades/en`

The MIME type of this submission data object file is as follows:

- `application/vnd.etsi.asic-s+zip`

4.12.13. PDF documents with PAdES signatures according to ETSI TS 103 172

PDF electronic document with PAdES electronic signature or electronic seal conformant to PAdES baseline profile standard ETSI TS 103 172 [TS 103 172].

This submission data object is identified by the following identifier:

- `PDF-PAdES-TS`

This submission data object is identified by the following URI identifier:

- `http://uri.mitsoft.lt/ades/type/pdf/pades/ts`

The MIME type of this submission data object file is as follows:

- `application/pdf`

4.12.14. PDF document with PAdES signatures according to ETSI EN 319 142-1

PDF electronic document with PAdES electronic signature or electronic seal conformant to Building blocks and PAdES baseline signatures standard ETSI EN 319 142-1 [EN 319 142-1].

This submission data object is identified by the following identifier:

- `PDF-PAdES-EN`

This submission data object is identified by the following URI identifier:

- `http://uri.mitsoft.lt/ades/type/pdf/pades/en`

The MIME type of this submission data object file is as follows:

- `application/pdf`

4.12.15. PDF documents with CMS signatures

PDF electronic document with PAdES electronic signature or electronic seal conformant to Profile for CMS Signatures in PDF defined in the standard ETSI TS 102 778-2 "PDF Advanced Electronic Signature Profiles; Part 2: PAdES Basic - Profile based on ISO 32000-1" [TS 102 778-2]. Long-term form for this profile is defined in the standard ETSI TS 102 778-4 "PDF Advanced Electronic Signature Profiles; Part 4: PAdES Long Term - PAdES-LTV Profile" [TS 102 778-4].

This submission data object is identified by the following identifier:

- `PDF-PAdES-CMS`

This submission data object is identified by the following URI identifier:

- `http://uri.mitsoft.lt/ades/type/pdf/pades/cms`

The MIME type of this submission data object file is as follows:

- `application/pdf`

4.13. Supported preservation evidence formats

The following preservation evidence formats are supported by this preservation profile: XAdES Archive Time Stamp, CAdES Archive Time Stamp V3, PAdES Document Time Stamp.

4.13.1. XAdES Archive Time Stamp

The XML-based Archive Time Stamp property according ETSI TS 101 903 [TS 101 903] and ETSI EN 319 162-1 [EN 319 162-1] is used for preserving XAdES digital signatures that are part of ADOC-V1.0, ADOC-V2.0, EGAS-V1.0, MDOC-V1.0 electronic documents or ASiC-E, ASiC-S containers with XAdES digital signatures.

4.13.2. CAdES Archive Time Stamp V3

The ASN.1-based Archive Time Stamp V3 attribute according ETSI TS 101 733 [TS 101 733] and ETSI EN 319 122-1 [EN 319 122-1] is used for preserving CAdES digital signatures that are part of ASiC-S containers with CAdES digital signatures.

4.13.3. PAdES Document Time Stamp

The Document Time-Stamp attribute according ETSI TS 102 778-4 [TS 102 778-4] and ETSI EN 319 142-1 [EN 319 142-1] is used for preserving PAdES digital signatures that are part of PDF-LT-V1.0, PDF-RC-V1.0 electronic documents or PDF documents with PAdES digital signatures.

4.14. Cryptographic monitoring

Cryptographic monitoring for algorithms used inside submission data objects is based on the current algorithm reliability status and is used only during digital signature validation (performed on preservation object submission or by the request).

Cryptographic monitoring for algorithms used for digital signature augmentation is based on the expected algorithm reliability status and is applied for the newly created preservation evidences (time stamps) only.

MitSoft PSP Cryptographic monitoring implementation is based on the Cryptographic algorithm registry and digital signature preservation metadata.

Cryptographic algorithm registry stores information about supported cryptographic algorithms and their reliability status. The stored data cover the following:

- Name of the algorithm.
- Algorithm identifiers (OID and URI).
- Algorithm type (hash function, signature algorithm, canonicalization algorithm).
- Key lengths (for signature algorithms only).
- Algorithm reliability status – is it reliable at the current time and may be used in the submitted data, is it reliable to be used for preservation evidences collected during digital signature augmentation.
- Expected algorithm reliability time – the time it is thought, that algorithm will be still reliable at least at that time. Expected algorithm reliability time may be set and also may be moved forward during Cryptographic algorithm registry revision, if new information about algorithm reliability becomes available.

MitSoft PSP Cryptographic algorithm registry revision and update is performed regularly and reflects the recommendations presented in the standard ETSI TS 119 312 "Electronic Signatures and Infrastructures (ESI); Cryptographic Suites" standard [TS 119 312]. The expected resistance period according to ETSI TS 119 312 of the cryptographic algorithms (their parameters and key sizes) used for new preservation evidences should be 3 years or more at a time of augmentation.

Cryptographic algorithm registry revision and update is performed by the system administrator of the MitSoft PSP. In the case cryptographic algorithm defined in the preservation evidence policy was identified to become no more secure enough for new preservation evidence creation, the new version of the Preservation evidence policy is issued.