GLOBALPLATFORM®
THE STANDARD FOR SECURE DIGITAL SERVICES AND DEVICES

GlobalPlatform Technology
TMF: Open Trust Protocol (OTrP) Mapping
Version 0.0.0.18

Public Review

May 2020

Document Reference: GPD_SPE_124

THIS SPECIFICATION OR OTHER WORK PRODUCT IS BEING OFFERED WITHOUT ANY WARRANTY WHATSOEVER, AND IN PARTICULAR, ANY WARRANTY OF NON-INFRINGEMENT IS EXPRESSLY DISCLAIMED. ANY IMPLEMENTATION OF THIS SPECIFICATION OR OTHER WORK PRODUCT SHALL BE MADE ENTIRELY AT THE IMPLEMENTER'S OWN RISK, AND NEITHER THE COMPANY, NOR ANY OF ITS MEMBERS OR SUBMITTERS, SHALL HAVE ANY LIABILITY WHATSOEVER TO ANY IMPLEMENTER OR THIRD PARTY FOR ANY DAMAGES OF ANY NATURE WHATSOEVER DIRECTLY OR INDIRECTLY ARISING FROM THE IMPLEMENTATION OF THIS SPECIFICATION OR OTHER WORK PRODUCT.

Contents

Figures

Tables

1 Introduction

GlobalPlatform has defined a security model for the administration of Trusted Execution Environments (TEEs) and the administration of Trusted Applications (TAs) and corresponding Security Domains (SDs), collectively referred to as the TEE Management Framework (TMF). TMF is defined in multiple specifications:

- *TEE Management Framework (TMF) including ASN.1 Profile* ([TMF ASN.1]) presents the roles and responsibilities of the stakeholders involved in the administration, the life cycle of administrated entities, and the mechanisms involved in administration operations. In addition, it describes an ASN.1 implementation (referred to as the ASN.1 Profile).
- *TMF: Open Trust Protocol (OTrP) Protocol* ([OTrP Profile]) describes an OTrP implementation (referred to as the OTrP Profile), including the OTrP Security Domain and associated security mechanisms, and specifies the JSON encoding for OTrP messages.
- Other profiles may be defined in separate specifications.

This document specifies a mapping between TMF OTrP Profile messages and TMF ASN.1 Profile commands. The document specifies how TMF OTrP request messages received by a TEE are mapped to TMF ASN.1 Profile commands and how TMF ASN.1 response output is mapped to TMF OTrP response messages. Direct mapping is not mandatory – that is, incoming request messages from the OTrP Profile are not required to be converted to the ASN.1 Profile – but this is a possible realization where appropriate.

The execution of a TMF OTrP request message SHALL provide the same result that the equivalent TMF ASN.1 Profile command would have achieved.

If you are implementing this specification and you think it is not clear on something:

1. **Check with a colleague.**

And if that fails:

2. **Contact GlobalPlatform at TEE-issues-GPD_SPE_124_v1.0@globalplatform.org**

1.1 Audience

This document is suitable for software developers implementing an OTrP Profile using the TEE Management Framework to manage Trusted Execution Environments (TEEs), as well as Trusted Applications (TAs) and their corresponding Security Domains (SDs).

This document is also intended for implementers of the TEE itself, its Trusted OS, the Trusted Core Framework, the TEE APIs, and the communications infrastructure required to access Trusted Applications.

1.2 IPR Disclaimer

Attention is drawn to the possibility that some of the elements of this GlobalPlatform specification or other work product may be the subject of intellectual property rights (IPR) held by GlobalPlatform members or others. For additional information regarding any such IPR that have been brought to the attention of GlobalPlatform, please visit https://globalplatform.org/specifications/ip-disclaimers/. GlobalPlatform shall not be held responsible for identifying any or all such IPR, and takes no position concerning the possible existence or the evidence, validity, or scope of any such IPR.

1.3 References

The table below lists references applicable to this specification. The latest version of each reference applies unless a publication date or version is explicitly stated.

Table 1-1: Normative References

Standard / Specification	Description	Ref
GPD_SPE_007	GlobalPlatform Technology TEE Client API Specification	[TEE Client]
GPD_SPE_010	GlobalPlatform Technology TEE Internal Core API Specification	[TEE Core]
GPD_SPE_120	GlobalPlatform Technology TEE Management Framework (TMF) including ASN.1 Profile [Initially published as TEE Management Framework]	[TMF ASN.1]
GPD_SPE_121	GlobalPlatform Technology TMF: Symmetric Cryptography Security Layer	[TMF Symmetric]
GPD_SPE_122	GlobalPlatform Technology TMF: Asymmetric Cryptography Security Layer	[TMF Asymmetric]
GPD_SPE_123	GlobalPlatform Technology TMF: Open Trust Protocol (OTrP) Protocol	[OTrP Profile]
RFC 2119	Key words for use in RFCs to Indicate Requirement Levels	[RFC 2119]
RFC 5280	Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile https://tools.ietf.org/html/rfc5280	[RFC 5280]
RFC 7515	JSON Web Signature https://tools.ietf.org/html/rfc7515	[RFC 7515]
RFC 7516	JSON Web Encryption https://tools.ietf.org/html/rfc7516	[RFC 7516]

1.4 Terminology and Definitions

The following meanings apply to SHALL, SHALL NOT, MUST, MUST NOT, SHOULD, SHOULD NOT, and MAY in this document (refer to [RFC 2119]):

- **SHALL** indicates an absolute requirement, as does MUST.
- **SHALL NOT** indicates an absolute prohibition, as does MUST NOT.
- **SHOULD** and **SHOULD NOT** indicate recommendations.
- **MAY** indicates an option.

Selected technical terms used in this document are defined in [TMF ASN.1] and [TEE Core].

Additional technical terms are defined in Table 1-2.

Table 1-2: Terminology and Definitions

Term	Definition
ASN.1 Profile	A specification of the TMF commands, written in ASN.1.
OTrP Mapping Implementation Layer (OMIL)	A trusted software module that can convert OTrP commands to ASN.1 and convert ASN.1 responses to OTrP.
OTrP Profile	A specification of TMF commands, written in JSON.
TEE Management Framework (TMF)	A security model for administration of Trusted Execution Environments (TEEs) and for administration and life cycle management of Trusted Applications (TAs) and corresponding Security Domains (SDs).

1.5 Abbreviations and Notations

Selected abbreviations and notations used in this document are defined in [TMF ASN.1] and [TEE Core].

Additional abbreviations and notations are included in Table 1-3.

Table 1-3: Abbreviations and Notations

Abbreviation / Notation	Meaning
DSI	Device State Information
OMIL	OTrP Mapping Implementation Layer
OTrP	Open Trust Protocol
OWE	Outside World Entity – In the TMF, this is usually the owner of the Security Domain.
rSD	Root Security Domain
SD	Security Domain
TA	Trusted Application
TEE	Trusted Execution Environment
TMF	TEE Management Framework

1.6 Revision History

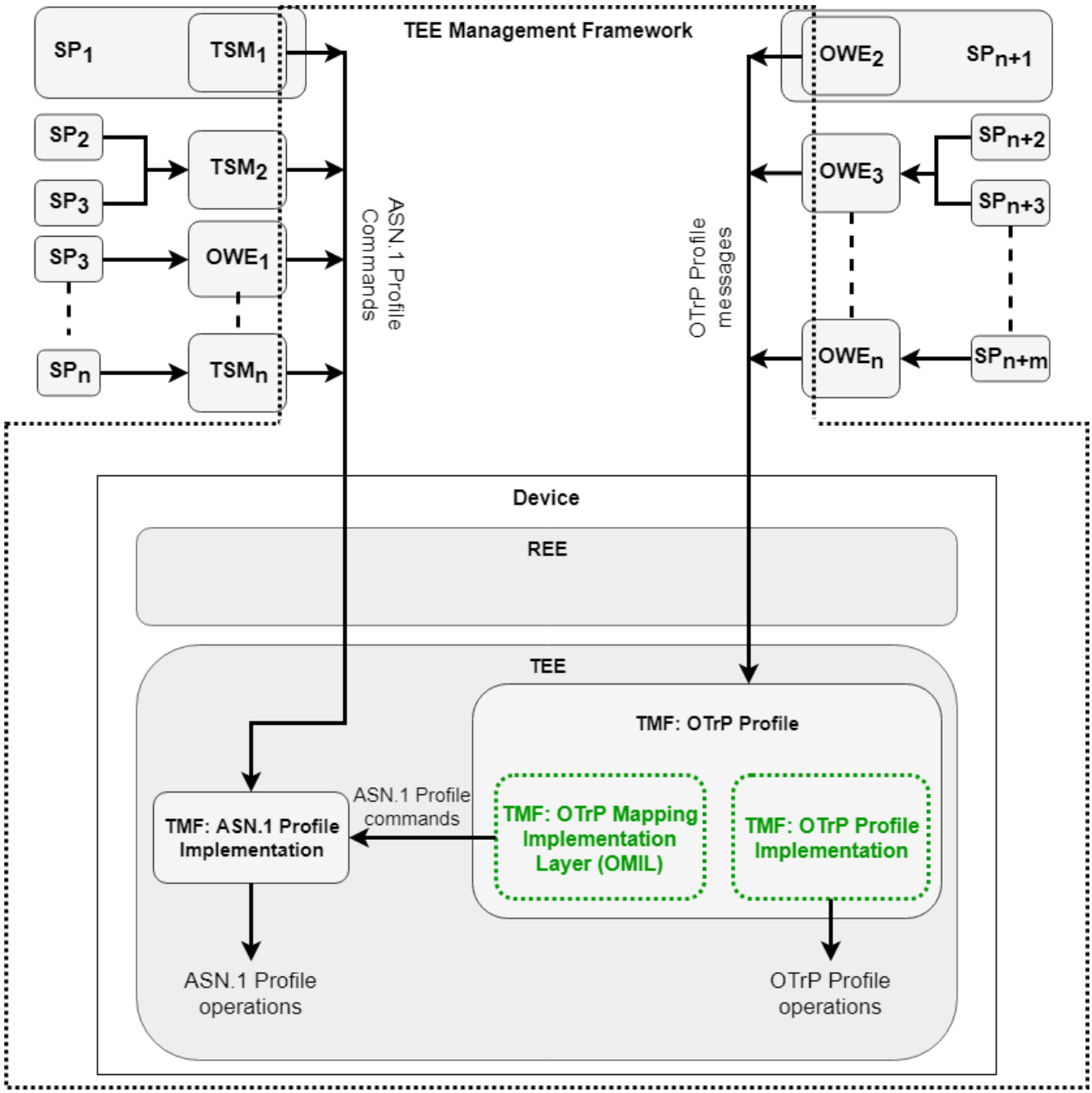
GlobalPlatform technical documents numbered *n*.0 are major releases. Those numbered *n*.1, *n*.2, etc., are minor releases where changes typically introduce supplementary items that do not impact backward compatibility or interoperability of the specifications. Those numbered *n.n*.1, *n.n*.2, etc., are maintenance releases that incorporate errata and precisions; all non-trivial changes are indicated, often with revision marks.

Table 1-4: Revision History

Date	Version	Description
November 2019	0.0.0.9	Committee Review
January 2020	0.0.0.14	Member Review
May 2020	0.0.0.18	Public Review **Question** Section 3.5, Processing OTrP Commands, says: • If an `rSD` is specified, verify that it exists and is a root SD for OTrP. Otherwise reject the request with the error `ERR_REQUEST_INVALID`. The author would like to receive comments on whether this is the correct error. Is a more specific error message needed? If so, the new error message will need to be added to [OTrP Profile].
TBD	1.0	Public Release

2 OTrP Profile Relationship with TMF: ASN.1 Profile

Figure 2-1: Using OMIL to Convert between OTrP and ASN.1



The TEE Management Framework (TMF) defines a security model for the administration of GlobalPlatform compliant Trusted Execution Environments (TEE), and the administration and life cycle management of Trusted Applications (TA) and their corresponding Security Domains (SD).

- *TEE Management Framework (TMF) including ASN.1 Profile* ([TMF ASN.1]) defines extensive commands for administration and life cycle management based on ASN.1 message format.
- *TMF: Open Trust Protocol (OTrP) Profile* ([OTrP Profile]) defines essential TEE management messages and essential TA and SD life cycle management messages based on JSON message format.

A TEE may support the ASN.1 Profile (as described in [TMF ASN.1]), the OTrP Profile (as described in [OTrP Profile]), or both.

- Trusted Service Managers (TSM) or Outside World Entities (OWE) that support the ASN.1 Profile use ASN.1 Profile commands to administer the TEEs on authorized devices.
- OWEs that support the OTrP Profile use OTrP messages to administer TAs and SDs on authorized devices.
- A Service Provider (SP) may choose between a TSM or an OWE for the life cycle management of its TAs in TEEs.

The execution of a TMF OTrP request message SHALL provide the same result that the equivalent TMF ASN.1 Profile command would have achieved.

Figure 2-1 depicts an overview of a TEE that supports both the ASN.1 Profile and the OTrP Profile.

A TEE that already supports the ASN.1 Profile may integrate OTrP Profile support using one of the following methods:

- Implementing OTrP Profile functionality directly into the TEE OS.
- Implementing an OTrP Mapping Implementation Layer (OMIL) that reuses the existing ASN.1 Profile support. OMIL needs to store secrets and state. It must therefore be implemented within the same Security Domain as the TEE. It may be implemented as a combination of Client Application and Trusted Application, but the Client Application must not have access to any data that may be used to compromise the system.

This specification focuses on the latter method and recommends the implementation details for OMIL, its responsibilities, as well as details on how to map OTrP request messages to ASN.1 Profile commands and ASN.1 Profile response output back to OTrP response messages. It is assumed that OMIL has no special access to the TEE – that is, it can only issue TEE Core API and TMF commands using the TEE Client API.

3 OTrP Mapping Implementation Layer (OMIL)

3.1 Authorizing Commands

In order to issue TMF commands, OMIL must be provisioned with the appropriate keys.

GlobalPlatform recommends that OMIL create an rSD under which all the OTrP administered SDs are then created.

OMIL should then provision this SD with a freshly generated key of the appropriate type.

This requires that OMIL is given appropriate permissions. This MAY be in the form of appropriately signed Authorization Tokens restricted to a fixed UUID.

Where OMIL runs as a TA, communication with the TEE is over an intrinsically secure channel. If the TEE's TMF implementation supports token-based authorization, there is no requirement for OMIL to set up a security layer. Therefore, it only needs to be provisioned with a key that can verify the authorization token. As we have an intrinsically secure channel to deliver that key, there is no advantage to using asymmetric cryptography; a symmetric key using HMAC is sufficient.

However, if OMIL is implemented in the REE, or if the commands pass through the REE, or if the TMF implementation does not support tokens – then OMIL would need to set up a security layer. If the key to set up the channel has to be delivered over an insecure channel, then OMIL should provision the SD with a public key and use the asymmetric security layer (as discussed in *TMF: Asymmetric Cryptography Security Layer* - [TMF Asymmetric]) for further commands.

By using a freshly generated key known only to the OMIL instance, we can guarantee that only OMIL can access the rSD and hence administer SDs on behalf of the OWE or SP.

OMIL needs to be able to manage all OTrP Security Domains. However, as it only has a single storage space, there is no advantage to OMIL using individual keys to manage each SD. An attacker that can access the single key would also be able to access individual keys. Therefore, OMIL should have SD Management permission covering the entire set of domains under its rSD.

If a Service Provider uses OTrP commands to create an SD but subsequently wants to use ASN.1 Profile commands to administer it, the Service Provider will need an individual key. Therefore, each individual key should be stored in the individual SD's PERSO storage area. (For more information, see section 5.)

3.2 Keys

3.2.1 TEE-Priv, TEE-Pub, and TEE-Cert

TEE-Priv is a private key that is unique per TEE instance. TEE-Priv is used to sign messages on behalf of the TEE.

TEE-Cert should be the Base64 encoding of a X.509 certificate on the private key representing the TEE instance. The certificate must be rooted in a key whose hash is known to the OTrP server.

Because OMIL generates OTrP messages on behalf of the TEE, it is permissible for OMIL to store TEE-Priv, TEE-Pub, and TEE-Cert in its PRIVATE or PERSO storage.

TEE-Priv must have sign permission as it signs transaction IDs (TID).

3.2.2 TFW-Priv, TFW-Pub, and TFW-Cert

TFW-Priv is a private key representing the trusted firmware underlying the TEE. All communication with TFW is implementation defined.

TFW is optional. Therefore, an OMIL implementation may always choose to return an empty structure for TFW in all responses.

3.2.3 OWE-Whitelist

OMIL must maintain two white lists of root hashes: one for Security Domain installation and one for TEE maintenance.

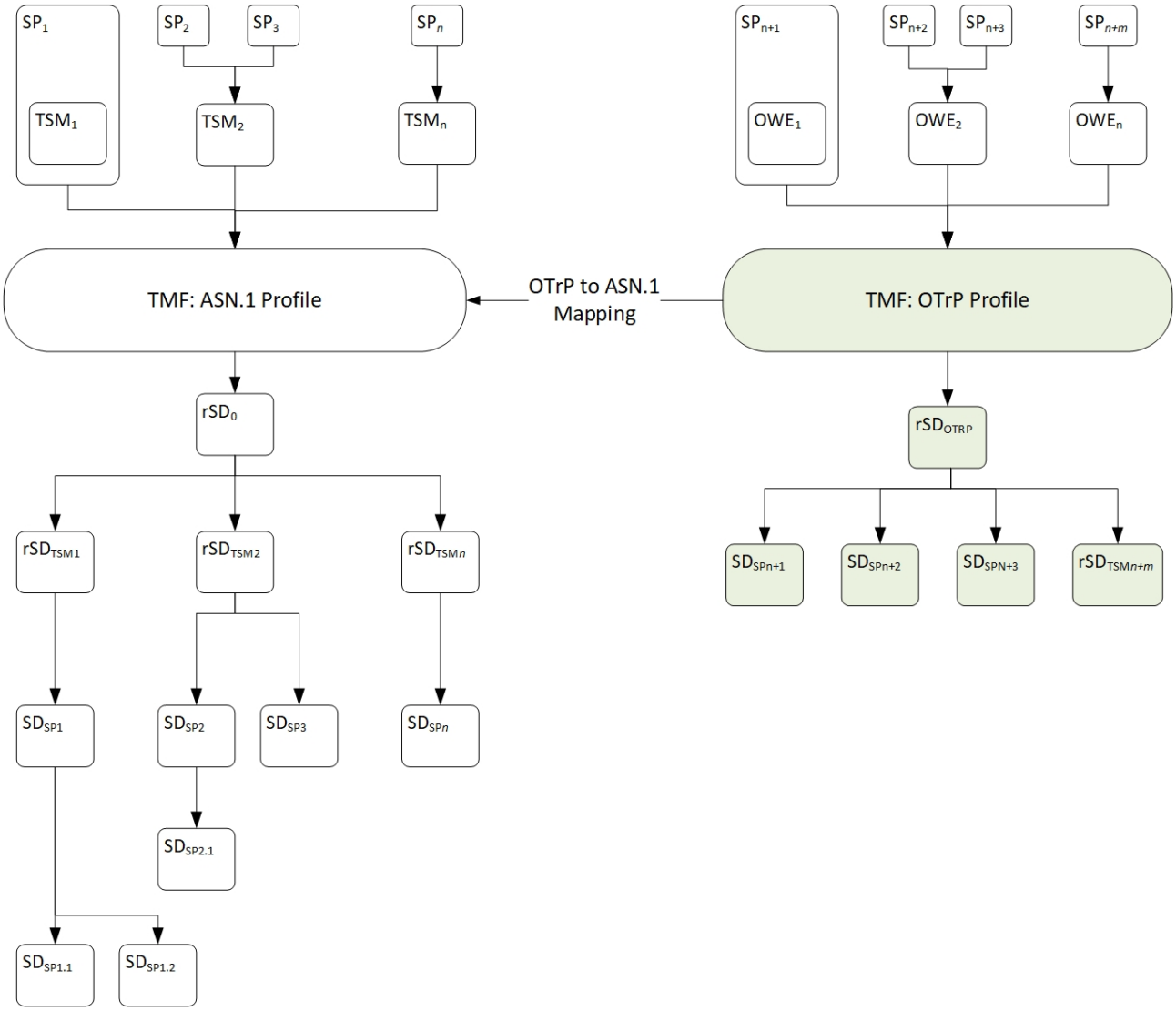
OTrP does not provide any mechanism for updating the white lists. Therefore, the mechanism for storing these lists is implementation defined. They can be encoded in the OMIL binary or stored in OMIL PERSO or PRIVATE storage.

3.3 Security Domain Mapping

OMIL should create an rSD as the root of an OTrP hierarchy, then create an individual Security Domain for each Service Provider under this rSD.

The OMIL TA should not reside in the OMIL hierarchy; it should be in an SD managed by the OMIL developer. This ensures that the OMIL TA can be upgraded without access to the OTrP-managed SDs or TAs.

Figure 3-1: Security Domains



OMIL MAY maintain a list of SDs that it has created on behalf of OWEs as OMIL-SD-List. The OMIL-SD-List MAY contain information regarding the UUIDs of the SDs, tsmid, spid, and SP-AIK.

3.4 OPERATION-RESPONSE-PRIMITIVE-TYPE

OMIL maintains an `OPERATION-RESPONSE-PRIMITIVE-TYPE` indicating the status of the OTrP operation based on the return code of the corresponding TMF ASN.1 Profile command. The following table recommends return code mapping to `OPERATION-RESPONSE-PRIMITIVE-TYPE`.

Table 3-1: OTrP Profile `OPERATION-RESPONSE-PRIMITIVE-TYPE` vs. TMF ASN.1 Return Codes

TMF ASN.1 Return Codes	OTrP Profile `OPERATION-RESPONSE-PRIMITIVE-TYPE`
`TEE_SUCCESS`	`OPERATION_SUCCESS`
`TEE_ERROR_ACCESS_DENIED`	`ERR_OWE_NOT_TRUSTED`
`TEE_ERROR_BAD_FORMAT`	`ERR_REQUEST_INVALID`
`TEE_ERROR_BAD_PARAMETERS`	`ERR_REQUEST_INVALID`
`TEE_ERROR_BAD_STATE`	`ERR_TEE_FAIL`
`TEE_ERROR_CORRUPT_OBJECT`	`ERR_TEE_FAIL`
`TEE_ERROR_EXCESS_DATA`	`ERR_REQUEST_INVALID`
`TEE_ERROR_INTERNAL`	`ERR_TEE_FAIL`
`TEE_ERROR_ITEM_NOT_FOUND`	`ERR_TA_NOT_FOUND`
`TEE_ERROR_LIMIT_EXCEEDED`	`ERR_TEE_RESOURCE_FULL`
`TEE_ERROR_NOT_SUPPORTED`	`ERR_REQUEST_INVALID` (if returned from the ASN.1 Profile `Store Data` command) `ERR_UNSUPPORTED_CRYPTO_ALG` (if returned from the ASN.1 Profile `Install SD` command)
`TEE_ERROR_OUT_OF_MEMORY`	`ERR_TEE_RESOURCE_FULL`
`TEE_ERROR_STORAGE_NO_SPACE`	`ERR_TEE_RESOURCE_FULL`
`TEE_ERROR_STORAGE_NOT_AVAILABLE`	`ERR_TEE_RESOURCE_FULL`

3.4.1 Handling Temporary Failure

If an OMIL implementation receives a temporary failure response to a TMF ASN.1 Profile command, it should attempt to resubmit the command rather than return the error to the client.

If it encounters a `TEE_ERROR_SHORT_BUFFER` response, it should determine the correct length of buffer required and if there is enough memory to allocate this buffer, it should resubmit the command. Only if it is not possible to allocate sufficient memory to submit the command successfully should OMIL return an OTrP error.

3.4.2 Handling Errors after Multiple Commands

If processing a single OTrP command requires OMIL to submit multiple TMF ASN.1 Profile commands, it is possible that an initial command may succeed but a later command may fail.

In this case, OMIL SHALL take steps to reverse the effect of the successful TMF ASN.1 Profile command, to leave the state of the device unchanged.

3.5 Processing OTrP Commands

All JSON fragments in this section are informative only. See [OTrP Profile] for the normative reference.

All OTrP commands have the general format:

```
{
   "payload":COMMAND-PAYLOAD,
   "protected":{
      "alg":"PRINTABLE-STRING-PRIMITIVE-TYPE",
      "rSD":"PRINTABLE-STRING-PRIMITIVE-TYPE",
      "tee":"PRINTABLE-STRING-PRIMITIVE-TYPE"
   }
   "header":{
      "x5c":["CERT-PRIMITIVE-TYPE"],
      "kid":"PRINTABLE-STRING-PRIMITIVE-TYPE"
   }
   "signature":"PRINTABLE-STRING-PRIMITIVE-TYPE"
}
```

Where:

- `payload:` The `COMMAND-PAYLOAD` used as a payload to generate a signature.
- `alg:` A cryptographic algorithm used to sign a message.
- `rSD:` (OPTIONAL) The UUID of the rSD that is supposed to receive the request message.
- `tee:` (OPTIONAL) A zero-terminated string that describes the TEE to connect to.
- `x5c:` An X.509 Certificate Chain (as described in [RFC 5280]) represented as a `CERT-PRIMITIVE-TYPE` array.
- `kid:` (OPTIONAL) A string indicating the key used in the JWS scheme for signing data.
- `signature:` The base64url encoded signature.

All command payloads include:

```
{
   "ver":"GPD-VERSION-TYPE",
   "tid":"PRINTABLE-STRING-PRIMITIVE-TYPE",
   "rid":"PRINTABLE-STRING-PRIMITIVE-TYPE",
   "tee":"PRINTABLE-STRING-PRIMITIVE-TYPE",
   "nextdsi":BOOLEAN,
   "dsihash":"PRINTABLE-STRING-PRIMITIVE-TYPE",
   "nonce":"PRINTABLE-STRING-PRIMITIVE-TYPE",
   "content":CONTENT-ENCRYPTION-TYPE
}
```

Where:

- `ver`: The version of the OTrP message, structured as `GPD-VERSION-TYPE` defined in [OTrP Profile] section 5.4.
- `tid`: A unique value for the ongoing transaction. The `tid` value is initially received in the `GetDeviceTEEStateTBSRequest` message and remains unchanged during an OTrP session.
- `rid`: A unique value that identifies the OTrP request. The response SHALL contain the same `rid` value as the corresponding request.
- `tee`: A zero-terminated string that describes the TEE to connect to. Its value matches the parameter `name` used to connect to a TEE while initializing a context using the `TEEC_InitializeContext` command ([TEE Client] section 4.5.2). When this element is not supplied, the OTrP request SHALL be sent to the default TEE on the device.
- `nextdsi`: A Boolean value indicating whether a newly calculated `DSI-TYPE` SHALL be returned in the corresponding response message.
- `dsihash`: The base64 encoded SHA-256 hash of the `DSI-TYPE` sent along with the OTrP request. `dsihash` received SHALL be compared with the SHA-256 hash of the internal `DSI-TYPE`.
- `nonce`: The `nonce` value SHALL match the value of the `nextnonce` sent to the OWE in the immediately previous response.
- `content`: Encrypted data structured as a `CONTENT-ENCRYPTION-TYPE`. The input to the encryption function is specific to the request message type as detailed within the request descriptions.

OMIL must:

- Verify that it supports the algorithm used for the signature. Otherwise reject the request with the error `ERR_UNSUPPORTED_CRYPTO_ALG`.
- Retrieve the key to verify the signature, `kid`. If OMIL cannot open the key, or it is not the correct type for the algorithm, reject the request with the error `ERR_REQUEST_INVALID`.
- If a `tee` is specified in the `protected` section, verify that the description matches that for the current TEE. Otherwise reject the request with the error `ERR_TEE_UNKNOWN`.
- If the `tee` value in the command payload is not `Null`, verify that the description matches that for the current TEE. Otherwise reject the request with the error `ERR_TEE_UNKNOWN`.

- If an `rSD` is specified, verify that it exists and is a root SD for OTrP. Otherwise reject the request with the error `ERR_REQUEST_INVALID`.
- Validate the JSON web signature associated with the request. Otherwise reject the message with the error `ERR_REQUEST_INVALID`.
- Verify that the OWE-Cert chains to a root CA certificate in the OWE-Whitelist. Otherwise reject the message with the error `ERR_REQUEST_INVALID`.
- Validate the OCSP data. If this has expired, reject the command with the error `ERR_OCSP_INVALID`.
- Verify that the `tid` is the value expected for the current transaction. Otherwise reject the request with the error `ERR_REQUEST_INVALID`.
- Verify that the `nonce` supplied in the command matches the value of `nextnonce` sent to the OWE in the previous response. Otherwise reject the request with the error `ERR_REQUEST_INVALID`.
- If the request is valid, store the request identifier (`rid`) and store a copy to be returned in the response, then attempt to process the command. Otherwise return a response containing the relevant error with all other fields empty.

3.5.1 Use of Nonces

OTrP uses nonces to enforce the sequence of commands.

Within each session, nonces must be statistically unique; however, it is permissible to use a counter. OMIL may use a single counter shared between all sessions.

OMIL SHALL store the current `nextnonce` for each open session and compare its value to that returned in the next command from that OWE.

However, there is no requirement to store this value between sessions; therefore, it can be stored in volatile memory.

3.5.2 Device State Information

The Device State Information (DSI) contains the current configuration information for all Security Domains managed by a particular OWE. OMIL SHALL maintain the DSI for each OWE that has created one or more SDs on the device using OTrP Profile messages. OMIL is also responsible for providing the DSI to the OWE at the beginning of the OTrP session and in OTrP response messages if indicated by the OWE in the preceding request.

The DSI is represented as `DSI-TYPE` ([OTrP Profile] section 4.14), which contains a `DSI-CONTENT-TYPE` ([OTrP Profile] section 4.15).

The first element of `DSI-CONTENT-TYPE`, `tfwdata`, is generated according to `TRUSTED-FIRMWARE-TYPE` as defined in [OTrP Profile] section 4.17. This is optional. For OMIL this element can be omitted.

The second element, `tee`, is generated according to `TEE-DESCRIPTION-TYPE` as defined in [OTrP Profile] section 4.18.

Therefore, for OMIL, a DSI has the format:

```
"dsi":{
   "tee":{
      "name":"PRINTABLE-STRING-PRIMITIVE-TYPE",
      "teever":"GPD-VERSION-TYPE",
      "cert":"CERT-PRIMITIVE-TYPE",
      "cacert":["CERT-PRIMITIVE-TYPE"],
      "sdlist":[{
         "sdid":"PRINTABLE-STRING-PRIMITIVE-TYPE",
         "spid":"PRINTABLE-STRING-PRIMITIVE-TYPE",
         "protocol":"PRINTABLE-STRING-PRIMITIVE-TYPE",
         "talist":[{
            "taid":"PRINTABLE-STRING-PRIMITIVE-TYPE",
            "taver":"PRINTABLE-STRING-PRIMITIVE-TYPE
            "} … ]
         }],
         "teeaiklist":[{
            "spaik":[PUB-KEY-ROLE-ARRAY-TYPE],
            "spid":"PRINTABLE-STRING-PRIMITIVE-TYPE"
            }],
         "isaset":ISA-TYPE,
         "teeImplementationProperty":[
            "gpd.tee.tmf.resetpreserved.entities" [BASE64_UUID,…]
            ]
      }
}
```

Where:

- `name:` The parameter name used to connect to a TEE while initializing a context using the `TEEC_InitializeContext` command ([TEE Client] section 4.5.2).
- `teever:` The version of the TEE, structured as the string `"GPD.TEE.[Major].[Minor].[Maintenance].[RFU]"`
- `cert:` The certificate on TEE-Pub from OMIL storage.
- `cacert:` The remaining certificates in the chain from OMIL storage.
- `sdlist:` The list of Security Domains known to the `spid`. OMIL only permits a single Security Domain per spid, tsmid pair so this will be zero or one Security Domains.
  - o `sdid:` The Security Domain identifier.

310 - o `spid:` The Service Provider identifier.
311 - o `protocol:` From `Get SD Definition` `SecurityDomain.protocols` ([TMF ASN.1]
312 section 9.2.2).
313 • `talist:` The list of Trusted Applications in a Security Domain can be obtained with the ASN.1 Profile
314 command `Get List of TA` ([TMF ASN.1] section 8.8.3).
315 - o `taid:` The UUID for the TA.
316 - o `taver:` The version of the TA – this can be obtained using the ASN.1 Profile command
317 `Get TA Definition 1` ([TMF ASN.1] section 8.8.5). OMIL must submit a separate
318 `Get TA Definition 1` command for each TA in the list.
319 • `teeaiklist:` A list of the keys used by the Security Domain. These can be retrieved using the ASN.1
320 Profile command `List Objects` ([TMF ASN.1] section 8.6.3).
321 - o `spaik:` The public keys for the Security Domain.
322 - o `spid:` The Service Provider identifier.
323 • `isaset:` The ISA set. While this can be retrieved using the ASN.1 Profile command
324 `Get TEE Definition` ([TMF ASN.1] section 8.8.1), OMIL MAY cache this information.
325 • `teeImplementationProperty:` The list of Security Domains and Trusted Applications belonging to
326 this caller that are marked as to be preserved on Factory Reset.
327

4 OTrP Messages – ASN.1 Profile Commands Mapping

All JSON fragments in this chapter are informative only. See [OTrP Profile] for the normative reference.

When processing a command, if OMIL discovers an error in the request, it should stop and return a response containing that error. If it encounters an error during the processing, it should attempt to revert the state of the TEE to that before the command was issued.

4.1 GET-TA-INFORMATION

The OTrP `GET_TA_INFORMATION` request ([OTrP Profile] section 5.5) is analogous to the ASN.1 Profile command `Get TA Definition 1` ([TMF ASN.1] section 8.8.5).

Request

```
{
   "GetTAInformationRequest":{
      "ver":"GPD-VERSION-TYPE",
      "taid":"PRINTABLE-STRING-PRIMITIVE-TYPE",
      "spid":"PRINTABLE-STRING-PRIMITIVE-TYPE"
   }
}
```

Where:

- `ver:` The version of the command – to be returned in the response.
- `taid:` The base64 encoded UUID.
- `spid:` The Service Provider identifier.

Processing

OMIL must:

- Verify that the OTrP request is valid for this device.
- Obtain the `tsmid` from the `x5c` used to validate the OTrP message.
- Decode the `taid` and issue a `Get TA Definition 1` command.
  - On `TEE_ERROR_ITEM_NOT_FOUND`, return `ERR_TA_NOT_FOUND`.
  - On other errors, return the appropriate conversion as described in section 3.4.
- Calculate the expected Security Domain using the `spid` in the command and the `tsmid`.
- Verify that the expected Security Domain matches the `parent` field from the `Get TA Definition 1` response. Otherwise return `ERR_TA_NOT_FOUND`.
- Set `taver` to the version field of the Trusted Application from the `Get TA Definition 1` response.
- Return the response.

Response

```
{
   "GetTAInformationResponse":{
      "ver":"GPD-VERSION-TYPE",
      "status":"OPERATION-RESPONSE-PRIMITIVE-TYPE",
      "taid":"PRINTABLE-STRING-PRIMITIVE-TYPE",
      "taver":"PRINTABLE-STRING-PRIMITIVE-TYPE",
      "sdid":"PRINTABLE-STRING-PRIMITIVE-TYPE",
      "spid":"PRINTABLE-STRING-PRIMITIVE-TYPE",
      "tsmid":"PRINTABLE-STRING-PRIMITIVE-TYPE"
   }
}
```

Where:

- `ver:` The version of the command – from the request.
- `status:` `OPERATION_SUCCESS`; otherwise its value SHALL be an error string in section 3.4.
- `taid:` The base64 encoded UUID – from the request.
- `taver:`
  - On success: The version of the TA returned by the `Get TA Definition 1`.
  - On error: Empty.
- `sdid:` The calculated Security Domain identifier.
- `spid:` The Service Provider identifier – from the request.
- `tsmid:` The TSM identifier, as determined from the OWE CERT.

4.2 GET-DEVICE-TEE-STATE

The Get Device TEE State command starts an OTrP transaction.

Request

```
{
   "GetDeviceTEEStateTBSRequest":{
      "ver":"GPD-VERSION-TYPE",
      "tid":"PRINTABLE-STRING-PRIMITIVE-TYPE",
      "rid":"PRINTABLE-STRING-PRIMITIVE-TYPE",
      "ocspdat":OCSP-ARRAY-TYPE,
      "supportedsigalgs":[SIGNATURE-PRIMITIVE-TYPE]
   }
}
```

Where:

- `ver:` The version of the OTrP message, structured as `GPD-VERSION-TYPE`.
- `tid:` A unique value for the ongoing transaction – returned in the reply.
- `rid:` A unique value for this message – returned in the reply.
- `ocspdat:` `OCSP-ARRAY-TYPE` as described in [OTrP Profile] section 4.5.1. The first element of the array is the OCSP stapling for validating the OWE-Cert, followed by OCSP stapling for verifying each subsequent intermediate CA in the certificate chain.
- `supportedsigalgs:` (OPTIONAL) A list of signature algorithms supported by the OWE. Its value is an array of `SIGNATURE-PRIMITIVE-TYPE`. If this element is absent, the TEE SHALL use any signature algorithm defined by the `SIGNATURE-PRIMITIVE-TYPE`.

Processing

OMIL must:

- Verify that the OTrP request is valid for this device.
- Verify that the `tid` value is not currently in use for another transaction.
  - If the `tid` is in use, reject the request with the error `ERR_REQUEST_INVALID`.
- Check the revocation status of the OWE-Cert and its intermediate CA certificates in the chain, using the OCSP stapling.
- Cache the OCSP stapling for subsequent command checking. The TEE MAY use its own clock for OCSP stapling validation.
- Calculate the `dsi` value.
- Return `OPERATION_SUCCESS`.

Response

```
{
   "GetDeviceTEEStateTBSResponse":{
      "ver":"GPD-VERSION-TYPE",
      "status":"OPERATION-RESPONSE-PRIMITIVE-TYPE",
      "rid":"PRINTABLE-STRING-PRIMITIVE-TYPE",
      "tid":"PRINTABLE-STRING-PRIMITIVE-TYPE",
      "signerreq":BOOLEAN,
      "content":{
         "protected":"ENCRYPTION-PRIMITIVE-TYPE",
         "recipients":[{
            "header":{
               "alg":"KEYWRAP-PRIMITIVE-TYPE",
               "kid":"PRINTABLE-STRING-PRIMITIVE-TYPE"
               }],
            "encrypted_key":"PRINTABLE-STRING-PRIMITIVE-TYPE"
            }
         "iv":"PRINTABLE-STRING-PRIMITIVE-TYPE",
         "ciphertext":"{
            "dsi":DSI-CONTENT-TYPE,
            "nextnonce":"PRINTABLE-STRING-PRIMITIVE-TYPE"
            } ",
         "tag":"PRINTABLE-STRING-PRIMITIVE-TYPE"
      }
   }
}
```

Where:

- `ver:` The version of the OTrP message, structured as `GPD-VERSION-TYPE`.
- `status:` `OPERATION_SUCCESS`; otherwise its value SHALL be an error string in section 3.4.
- `rid:` A unique value for this message – returned in the reply.
- `tid:` A unique value for the ongoing transaction – returned in the reply.
- `signerreq:` A Boolean asking for OCSP data to be resent – recommend OMIL sets this to FALSE.
- `dsi:` See section 3.5.2 for details of the Device State Information.
- Other fields are as described in [OTrP Profile].

4.3 CREATE-SD

An OWE issues a `CreateSDTBSRequest` message to create a new Security Domain on a device.

Request

```
{
   "CreateSDTBSRequest":{
      "spid":"PRINTABLE-STRING-PRIMITIVE-TYPE",
      "sdid":"PRINTABLE-STRING-PRIMITIVE-TYPE",
      "spcert":"CERT-PRIMITIVE-TYPE",
      "tsmid":"PRINTABLE-STRING-PRIMITIVE-TYPE",
      "did":"PRINTABLE-STRING-PRIMITIVE-TYPE",
      "sd_data":"PRINTABLE-STRING-PRIMITIVE-TYPE"
   }
}
```

Where:

- `spid:` The Service Provider identifier.
- `sdid:` The Security Domain identifier.
- `spcert:` The Service Provider certificate.
- `tsmid:` The TSM identifier.
- `did:` The Device identifier.
- `sd_data:` The Security Domain data.

Processing

OMIL must:

- Validate that the command is valid for this device.
- Verify that the `did` is correct for this device.
- Verify that the `tsmid` matches the certificate chain.
- Verify that the `sdid` matches the `spid` and `tsmid`.
- Verify that the Security Domain `sdid` does not already exist on the device. If it does, reject the request with the error `ERR_SDID_ALREADY_USED`.
- Verify that the UUID is either UUID type 1 or type 4; if not, reject the command with the error `ERR_INVALID_UUID`.
- If all is correct, issue two separate ASN.1 Profile commands sequentially: `Install SD` ([TMF ASN.1] section 8.5.1), then `Store Data` ([TMF ASN.1] section 8.6.1).
- The `Install SD` command takes the parameters:
  - `SDLifecycleState` set to `SdActiveState`.

484 o `SDPrivileges` set to `gpd.privilege.sdPersonalization`, `gpd.privilege.taManagement`
485 and `gpd.privilege.taPersonalization`

486 • The `Store Data` command takes the parameter:

487 o `storedDataObject:` The `sd_data` field.

488 • As the OTrP command is expected to be monotonic, if the `Store Data` command fails, OMIL SHALL
489 issue an ASN.1 Profile command `Uninstall SD` ([TMF ASN.1] section 8.5.2) to ensure that the state
490 of the TEE is unchanged.

491 • Store the `spcert` in the OMIL private storage.

492 • Issue an ASN.1 Profile command `Fetch Object` ([TMF ASN.1] section 8.6.4) to obtain the public key
493 or keys.

494 • Calculate the new `dsi` value.

495 • Generate a new nonce.

496 • Issue the OTrP response.

497 **Response**

```
498 {
499    "CreateSDTBSResponse":{
500       "status":"OPERATION-RESPONSE-PRIMITIVE-TYPE",
501       "did":"PRINTABLE-STRING-PRIMITIVE-TYPE",
502       "sdid":"PRINTABLE-STRING-PRIMITIVE-TYPE",
503       "spaik":PUB-KEY-ROLE-ARRAY-TYPE,
504       "dsi":DSI-CONTENT-TYPE,
505       "nextnonce":"PRINTABLE-STRING-PRIMITIVE-TYPE"
506    }
507 }
```

508 Where:

509 • `status: OPERATION_SUCCESS`; otherwise its value SHALL be an error string in section 3.4.

510 • All other fields are as per [OTrP Profile].

4.4 UPDATE-SD

An OWE issues an `UpdateSDTBSRequest` message to update SD metadata with the given parameters. OMIL SHALL issue the ASN.1 Profile command `Store Data` ([TMF ASN.1] section 8.6.1) while mapping `UpdateSDTBSRequest` to the ASN.1 command using the following convention.

Request

```
{
   "tsmid":"PRINTABLE-STRING-PRIMITIVE-TYPE",
   "spid":"PRINTABLE-STRING-PRIMITIVE-TYPE",
   "did":"PRINTABLE-STRING-PRIMITIVE-TYPE",
   "sdid":"PRINTABLE-STRING-PRIMITIVE-TYPE",
   "changes":{
      "spcert":["CERT-PRIMITIVE-TYPE"],
      "deloldspcert":["PRINTABLE-STRING-PRIMITIVE-TYPE"],
      "sd_data":"PRINTABLE-STRING-PRIMITIVE-TYPE"
   }
}
```

Where:

- `tsmid:` The TSM identifier.
- `spid:` The Service Provider identifier.
- `did:` The Device identifier.
- `sdid:` The Security Domain identifier.
- `spcert:` The updated Service Provider certificate.
- `deloldspcert:` The current Service Provider certificate to delete.
- `sd_data:` The updated Security Domain data.

Processing

OMIL must:

- Verify that the OTrP request is valid for this device.
- Verify that the `did` is correct for this device.
- If the request contains a `newspid` element, deprecated in OTrP v1.1, reject the command with the error `ERR_UNSUPPORTED_MSG_VERSION` as TMF does not support renaming Security Domains.
- If the command includes a `deloldspcert` element, determine that a matching certificate exists.
  - If it does, open the corresponding persistent object with `TEE_OpenPersistentObject` ([TEE Core] section 5.7.1), but do not delete it until other processing has completed.
  - If no matching certificate exists, reject the command with the error `ERR_REQUEST_INVALID`.

- Store a new SP certificate in OMIL private storage using `TEE_CreatePersistentObject` ([TEE Core] section 5.7.2).
- Store any `sd_data` in the SD private storage using a `Store Data` command.
- If OMIL cannot store all the data, revert any previous `TEE_CreatePersistentObject` or `Store Data` command.
- Delete any old certificates using a `TEE_CloseAndDeletePersistentObject1` command ([TEE Core] section 5.7.4).
    - OMIL should not delete any certificates until it is certain it can store the new certificates supplied in the command. This ensures that the command can be rolled back. However, it does mean that some commands may fail that would have succeeded if the deletion was performed first.
- If successful, calculate a new DSI and nonce.
- Extract the public keys for the Security Domain using ASN.1 Profile `Fetch Object` commands ([TMF ASN.1] v1.1 section 8.6.4).
- Return the response.

Response

```
{
   "status":"OPERATION-RESPONSE-PRIMITIVE-TYPE",
   "did":"PRINTABLE-STRING-PRIMITIVE-TYPE",
   "spaik":PUB-KEY-ROLE-ARRAY-TYPE,
   "dsi":DSI-CONTENT-TYPE,
   "nextnonce":"PRINTABLE-STRING-PRIMITIVE-TYPE"
}
```

Where:

- `status:` `OPERATION_SUCCESS`; otherwise its value SHALL be an error string in section 3.4.
- `spaik:` The public keys for the Security Domain.
- All other fields are as per [OTrP Profile].

4.5 DELETE-SD

An OWE issues a `DeleteSDTBSRequest` message to update SD metadata with the given parameters. OMIL SHALL issue an ASN.1 Profile command `Uninstall SD` ([TMF ASN.1] section 8.5.2) while mapping `DeleteSDTBSRequest` to the ASN.1 Profile using the following convention.

Request

```
{
   "tsmid":"PRINTABLE-STRING-PRIMITIVE-TYPE",
   "did":"PRINTABLE-STRING-PRIMITIVE-TYPE",
   "sdid":"PRINTABLE-STRING-PRIMITIVE-TYPE",
   "deletetas":BOOLEAN
}
```

Where:

- `tsmid:` The TSM identifier.
- `did:` The Device identifier.
- `sdid:` The Security Domain identifier.
- `deletetas:` A Boolean value indicating whether the TAs within the Security Domain shall be deleted.

Processing

OMIL must:

- Verify that the OTrP request is valid for this device.
- Verify that the `did` is correct for this device.
- Verify that the Security Domain exists, for instance by issuing an ASN.1 Profile command `Get SD Definition` ([TMF ASN.1] section 8.8.2).
- If the Security Domain is the parent of another SD (created with TMF), reject the request with the error `ERR_SD_NOT_EMPTY`.
- Use the ASN.1 Profile command `Get List of TA` ([TMF ASN.1] section 8.8.3) to retrieve a list of all TAs in the SD.
- If the SD contains any TAs and `deletetas` is set to FALSE, reject the request with the error `ERR_SD_NOT_EMPTY`.
- If `DeleteTAs` is TRUE, delete all TAs in the SD by issuing an ASN.1 Profile command `Uninstall TA` ([TMF ASN.1] section 8.4.2) for each TA.
- Delete the SD with an ASN.1 Profile command `Uninstall SD` ([TMF ASN.1] section 8.5.2).
- Calculate the new DSI and nonce.
- Return the appropriate response.

Response

```
{
   "status":"OPERATION-RESPONSE-PRIMITIVE-TYPE",
   "did":"PRINTABLE-STRING-PRIMITIVE-TYPE",
   "dsi":DSI-CONTENT-TYPE,
   "nextnonce":"PRINTABLE-STRING-PRIMITIVE-TYPE"
}
```

Where:

- `status: OPERATION_SUCCESS`; otherwise its value SHALL be an error string in section 3.4.
- All other fields are as per [OTrP Profile].

4.6 INSTALL-TA

An OWE issues an `InstallTATBSRequest` message to install a new TA on a device.

Request

```
{
   "tsmid":"PRINTABLE-STRING-PRIMITIVE-TYPE",
   "did":"PRINTABLE-STRING-PRIMITIVE-TYPE",
   "spid":"PRINTABLE-STRING-PRIMITIVE-TYPE",
   "sdid":"PRINTABLE-STRING-PRIMITIVE-TYPE",
   "spcert":"CERT-PRIMITIVE-TYPE",
   "taid":"PRINTABLE-STRING-PRIMITIVE-TYPE",
   "taver":"PRINTABLE-STRING-PRIMITIVE-TYPE",
   "pop_data":POP-TYPE
}
```

Where:

- `tsmid:` The TSM identifier.
- `did:` The Device identifier.
- `spid:` The Service Provider identifier.
- `sdid:` The Security Domain identifier.
- `spcert:` The Service Provider certificate.
- `taid:` The base64 encoded UUID for the TA.
- `taver:` The version of the TA.
- `pop_data:` `POP-TYPE` value SHALL be included when the given `taid` is a UUID version 5. It is used to perform a verification of proof of possession of a UUID version 5 as defined in [TMF ASN.1] section 8.3.3.7. (See details in [OTrP Profile] Annex D.)

Processing

OMIL must:

- Verify that the OTrP request is valid for this device.
- Verify that the `did` is correct for this device.
- Verify that the Security Domain exists and is the correct Security Domain for this OWE.
- Decrypt the `encrypted_ta_bin` using the `spaik`.
- Install the TA using an ASN.1 Profile command `Install TA` ([TMF ASN.1] section 8.4.1).
- Set the version number using the ASN.1 Profile command `Store TEE Property` ([TMF ASN.1] section 8.7.3).
- Decrypt the `encrypted_ta_data` using the `spaik`.

- Store the data associated with the TA using an ASN.1 Profile command `Store Data` ([TMF ASN.1] section 8.6.1).
- If the `Store Data` command fails, delete the TA to return to the previous state.
- Calculate the new DSI and Nonce.
- Return the response.

Response

```
{
   "status":"OPERATION-RESPONSE-PRIMITIVE-TYPE",
   "did":"PRINTABLE-STRING-PRIMITIVE-TYPE",
   "dsi":DSI-CONTENT-TYPE,
   "nextnonce":"PRINTABLE-STRING-PRIMITIVE-TYPE"
}
```

Where:

- `status: OPERATION_SUCCESS`; otherwise its value SHALL be an error string in section 3.4.
- All other fields are as per [OTrP Profile].

4.7 UPDATE-TA

An OWE issues an `UpdateTATBSRequest` message to update previously installed TA or TA data on a device. The `UpdateTATBSRequest` message may contain updates for TA binary (`encrypted_ta_bin`) only, TA data (`encrypted_ta_data`) only, or both.

Request

```
{
   "tsmid":"PRINTABLE-STRING-PRIMITIVE-TYPE",
   "did":"PRINTABLE-STRING-PRIMITIVE-TYPE",
   "spid":"PRINTABLE-STRING-PRIMITIVE-TYPE",
   "sdid":"PRINTABLE-STRING-PRIMITIVE-TYPE",
   "spcert":"PRINTABLE-STRING-PRIMITIVE-TYPE",
   "taid":"PRINTABLE-STRING-PRIMITIVE-TYPE",
   "newtaver":"PRINTABLE-STRING-PRIMITIVE-TYPE",
   "pop_data":POP-TYPE
}
```

Where:

- `tsmid:` The TSM identifier.
- `did:` The Device identifier.
- `spid:` The Service Provider identifier.
- `sdid:` The Security Domain identifier.
- `spcert:` The Service Provider certificate.
- `taid:` The base64 encoded UUID for the TA.
- `newtaver:` The string containing the TA version information that is to be updated.
- `pop_data: POP-TYPE` value SHALL be included when the given `taid` is a UUID version 5. It is used to perform a verification of proof of possession of a UUID version 5 as defined in [TMF ASN.1] section 8.3.3.7. (See details in [OTrP Profile] Annex D.)

Processing

OMIL must:

- Verify that the OTrP request is valid for this device.
- Verify that the `did` is correct for this device.
- Verify that the Security Domain exists and is the correct Security Domain for this OWE.
- Retrieve the version number of the current TA using the `TEE_GetPropertyAsString` command ([TEE Core] section 4.4.1).
- Ensure that the version specified in `newtaver` is higher than the current version – this will be TEE specific.

- Decrypt the `encrypted_ta_bin` and `encrypted_ta_data` fields.
- If the request only contains a Binary and no data, update the TA using the ASN.1 Profile command `Update TA` ([TMF ASN.1] section 8.4.3).
- If the request only contains data and no binary, update the data with an ASN.1 Profile command `Store Data` ([TMF ASN.1] section 8.6.1).
- If the request contains both TA and data, atomically update both using the ASN.1 Profile command `Update TA and Data` ([TMF ASN.1] v1.1 section 8.4.6).
- The new state of the TA SHALL always be `taExecutableState`.
- Set the version number using the ASN.1 Profile command `Store TEE Property` ([TMF ASN.1] section 8.7.3).
- Store the SP Certificate (`spcert`) provided with the `InstallTATBSRequest` in OMIL private storage and associate the `spcert` with the SD where the TA is being installed.
- Calculate the new DSI and Nonce.
- Return the response.
- OMIL SHALL use the following convention while mapping `UpdateTATBSRequest`.

Response

```
{
   "status":"OPERATION-REASON-PRIMITIVE-TYPE",
   "did":"PRINTABLE-STRING-PRIMITIVE-TYPE",
   "dsi":DSI-CONTENT-TYPE,
   "nextnonce":"PRINTABLE-STRING-PRIMITIVE-TYPE"
}
```

Where:

- `status: OPERATION_SUCCESS`; otherwise its value SHALL be an error string in section 3.4.
- All other fields are as per [OTrP Profile].

4.8 DELETE-TA

An OWE issues a `DeleteTATBSRequest` message to delete an existing TA on a device.

Request

```
{
   "tsmid":"PRINTABLE-STRING-PRIMITIVE-TYPE",
   "did":"PRINTABLE-STRING-PRIMITIVE-TYPE",
   "sdid":"PRINTABLE-STRING-PRIMITIVE-TYPE",
   "taid":"PRINTABLE-STRING-PRIMITIVE-TYPE"
}
```

Where:

- `tsmid:` The TSM identifier.
- `did:` The Device identifier.
- `sdid:` The Security Domain identifier.
- `taid:` The base64 encoded UUID for the TA.

Processing

OMIL must:

- Verify that the OTrP request is valid for this device.
- Verify that the `did` is correct for this device.
- Verify that the Security Domain exists and is the correct Security Domain for this OWE.
- Verify that the TA exists.
- Issue an ASN.1 Profile command `Uninstall TA` ([TMF ASN.1] section 8.4.2) to delete the TA.
- Calculate the new DSI and Nonce.
- Return the response.

Response

```
{
   "status":"OPERATION-RESPONSE-PRIMITIVE-TYPE",
   "did":"PRINTABLE-STRING-PRIMITIVE-TYPE",
   "dsi":DSI-CONTENT-TYPE,
   "nextnonce":"PRINTABLE-STRING-PRIMITIVE-TYPE"
}
```

Where:

- `status:` `OPERATION_SUCCESS`; otherwise its value SHALL be an error string in section 3.4.

- All other fields are as per [OTrP Profile].

4.9 STORE-TEE-PROPERTY

An OWE issues a `StoreTEEPropertyTBSRequest` message to store, update, or delete TEE properties. TEE properties are described in [TMF ASN.1] section A.5. The OTrP Profile supports only the TEE property `gpd.tee.tmf.resetpreserved.entities`, which is used to indicate entities as UUIDs to be preserved across a Factory Reset operation on TEE.

Request

```
{
   "tsmid":"PRINTABLE-STRING-PRIMITIVE-TYPE",
   "did":"PRINTABLE-STRING-PRIMITIVE-TYPE",
   "property":"gpd.tee.tmf.resetpreserved.entities",
   "value":{
         "taids":UUID-ARRAY-TYPE,
         "sdids":UUID-ARRAY-TYPE
      }
}
```

Where:

- `tsmid:` The TSM identifier.
- `did:` The Device identifier.
- `property:` The TEE property to store: `gpd.tee.tmf.resetpreserved.entities`
- `value:`
  - `taids:` UUIDS of TAs structured as `UUID-ARRAY-TYPE` that SHALL be preserved across a Factory Reset operation on TEE.
  - `sdids:` UUIDS of SDs structured as `UUID-ARRAY-TYPE` that SHALL be preserved across a Factory Reset operation on TEE.

Processing

OMIL must:

- Verify that the OTrP request is valid for this device.
- Verify that the `did` is correct for this device.
- Verify the certificate chains to a hash in the TEE maintenance whitelist.
- Verify that `property` equals "`gpd.tee.tmf.resetpreserved.entities`"; if not, reject the request with the error `ERR_REQUEST_INVALID`.
- Decode the base64 encoding of the value field; if the UUIDS cannot be decoded, reject the request with the error `ERR_REQUEST_INVALID`.
- Use the ASN.1 Profile command `Store TEE Property` ([TMF ASN.1] section 8.7.3) to update the `gpd.tee.tmf.resetpreserved.entities` property.
- Calculate the new DSI and Nonce.

- Return the response.

Response

```
{
   "status":"OPERATION-RESPONSE-PRIMITIVE-TYPE",
   "did":"PRINTABLE-STRING-PRIMITIVE-TYPE",
   "dsi":DSI-CONTENT-TYPE,
   "nextnonce":"PRINTABLE-STRING-PRIMITIVE-TYPE"
}
```

Where:

- `status: OPERATION_SUCCESS`; otherwise its value SHALL be an error string in section 3.4.
- All other fields are as per [OTrP Profile].

4.10 FACTORY-RESET

The Factory Reset command moves the TEE to a notional “factory” state

Request

```
{
   "tsmid":"PRINTABLE-STRING-PRIMITIVE-TYPE",
   "did":"PRINTABLE-STRING-PRIMITIVE-TYPE"
}
```

Where:

- `tsmid:` The TSM identifier.
- `did:` The Device identifier.

Processing

OMIL must:

- Verify that the OTrP request is valid for this device.
- Verify that the `did` is correct for this device.
- Verify the certificate chains to a hash in the TEE maintenance whitelist.
- Store sufficient data in OMIL private storage to indicate that a Factory Reset has been requested and by whom, so that when OMIL is restarted it can determine that a reset has happened.
- Use the ASN.1 Profile command `Factory Reset` ([TMF ASN.1] section 8.7.4) to reset the device.

Important: As OMIL is a TA, it will be terminated by the factory reset. Therefore, either the TEE or the OTrP agent in the REE must ensure that OMIL is restarted.

- Once OMIL has restarted, verify that only the expected Security Domains and Trusted Applications are present. If not, call `Factory Reset` again.
- Calculate the `dsi`.
- Generate a new nonce value.
- Return the response.

Response

```
{
   "status":"OPERATION-RESPONSE-PRIMITIVE-TYPE",
   "did":"PRINTABLE-STRING-PRIMITIVE-TYPE",
   "dsi":DSI-CONTENT-TYPE,
   "nextnonce":"PRINTABLE-STRING-PRIMITIVE-TYPE"
}
```

Where:

- `status: OPERATION_SUCCESS`; otherwise its value SHALL be an error string in section 3.4.
- Other fields are as per [OTrP Profile].

5 Enabling OTrP SD with TMF ASN.1 Profile Capability

To convert a Security Domain that has been created using the OTrP Profile into one that can be managed using the ASN.1 Profile, the OWE needs to inject a new key for use with a TMF Security Layer or for the verification of Authorization Tokens.

Depending on the chosen method, the following keys are required:

Table 5-1: Keys Required to Enable OTrP Security Domain with ASN.1 Profile Capability

Authorization Methods	Cryptographic Primitives	Permissions
Symmetric Security Layer	AES or Triple Des key	Derive permission
Asymmetric Security Layer	RSA or ECDSA private key	Sign permission
	RSA or ECDSA public key	Verify permission
Authorization Token	RSA or ECDSA private key	Verify permission
	HMAC	Verify permission

To inject the necessary keys, the OWE submits one or more `Update SD` commands (section 4.4).

The TMF Security Layers are defined in:

- TMF: Symmetric Cryptography Security Layer, [TMF Symmetric]
- TMF: Asymmetric Cryptography Security Layer, [TMF Asymmetric]