TEEP BOF
Architecture

Mingliang Pei
28th March 2017 -- IETF 98th, Chicago
OTrp Design Choices

- **Uses asymmetric keys and PKI**
  - Manufacturer-provided keys and trust anchors
  - Enables attestation between TAM and TEE-device

- **JSON-based messaging between TAM and TEE**
  - Messages for attestation
  - Messages for security domain management and TA management
  - Use JOSE (JSON signing and encryption specifications) – CBOR alternative spec available.

- **OTrp Agent in REE relays message exchanges between a TAM and TEE**

- **Device has a single TEE only**
Open Trust Protocol (OTrP) Overview

- CAs issue certificates to OTrP actors (TEE, TAM, SP)
- TAM and TEE exchange messages
- An OTrP Agent relays the OTrP message between TAM and TEE.
  - The communication between Rich App to TAM is up to SP and TAM.
OTrP Agent

- Responsible for routing OTrP Messages to the appropriate TEE
- Most commonly developed and distributed by TEE vendor
- Implements an interface as a service, SDK, etc.
* **AIK: Attestation Identity Key, TFW: Trusted Firmware**

### Keys

- **Certificate Authority**
  - CA Certificate

- **Service Provider**
  - SP Key pair and Certificate

- **TAM**
  - TAM Key pair and Certificate
  - Trust Anchors: trusted Root CA list of TEE certificates

- **Device TEE**
  - TEE Key pair and Certificate
  - TFW Key pair and Certificate (optional)

### Trust Anchors

- SP AIK in runtime for use by SP (encrypt TA data / verify)

### Usage

- * Key pair and Certificate: used to issue certificate
- * Key pair and Certificate: used to sign a TA
- * Key pair and Certificate: sign OTrP requests to be verified by TEE
- * Key pair and Certificate: device attestation to remote TAM and SP.
- * Key pair and Certificate: evidence of secure boot and trustworthy firmware
Entity Relationships

- TAM
  - managed by
  - manages
- Device
  - has > 1
- TEE
  - < is built into 1
- Service Provider
  - < developed by 1
- Trusted App
  - < produced by 1..n
- Security Domain
  - < is associated with 1..n
  - contains
  - belongs to
  - contains
  - belongs
- TAM
  - 1..m
Security of the Operation Protocol is enhanced by applying the following three Measures:

- [✓] Verifies validity of Message Sender’s Certificate
- [✓] Verifies signature of Message Sender to check immutability
- [✓] Encrypted to guard against exposure of Sensitive data

## Protocol Flow

### Phase#1
**“Device Attestation”**
Operation request triggered and verify Device state information

- Request to TSM for TA installation
- Send `[GetDeviceState]` to TEE
- Return DSI as a response to `[GetDeviceState]`

### Phase#2
Prerequisite operation (if Security domain doesn’t exist where the TA should be installed)

- Send `[CreateSD]` to create SD where the TA will be installed
- Send other prerequisite commands (if necessary)

### Phase#3
Perform Operation requested by SP or Client Application

- Send `[installTA]` with encrypted TA binary and its data

<table>
<thead>
<tr>
<th>TAM</th>
<th>Client App</th>
<th>TEE</th>
</tr>
</thead>
</table>

- Decrypt TA binary and its personal data.
- Install TA into target SD.
- Store personal data in TA’s private storage.
Discussion

Thank you!
JSON Message Security and Crypto Algorithms

• Use JSON signing and encryption RFCs
  – RFC 7515, JSON Web Signature (JWS)
  – RFC 7516, JSON Web Encryption (JWE)
  – RFC 7517, JSON Web Key (JWK)
  – RFC 7518, JSON Web Algorithms (JWA)

• Supported encryption algorithms
  – A128CBC-HS256
  – A256CBC-HS512

• Supported signing algorithms
  – RS256 (RSA 2048-bit key)
  – ES256 (ECC P-256)
interface IOTrPAgentService {
    String processMessage(String tsmInMsg) throws OTrPAgentException;
    String getTAInformation(String spid, String taid, byte[] nonce);
}

public class OTrPAgentException extends Throwable {
    private int errCode;
}
OTrP Operations and Messages

✓ Remote Device Attestation

<table>
<thead>
<tr>
<th>Command</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetDeviceState</td>
<td>• Retrieve information of TEE device state including SD and TA associated to a TAM</td>
</tr>
</tbody>
</table>

✓ Security Domain Management

<table>
<thead>
<tr>
<th>Command</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreateSD</td>
<td>• Create SD in the TEE associated to a TAM</td>
</tr>
<tr>
<td>UpdateSD</td>
<td>• Update sub-SD within SD or SP related information</td>
</tr>
<tr>
<td>DeleteSD</td>
<td>• Delete SD or SD related information in the TEE associated to a TAM</td>
</tr>
</tbody>
</table>

✓ Trusted Application Management

<table>
<thead>
<tr>
<th>Command</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>InstallTA</td>
<td>• Install TA in the SD associated to a TAM</td>
</tr>
<tr>
<td>UpdateTA</td>
<td>• Update TA in the SD associated to a TAM</td>
</tr>
<tr>
<td>DeleteTA</td>
<td>• Delete TA in the SD associated to a TAM</td>
</tr>
</tbody>
</table>
OTrP JSON Message Format and Convention

{
    "<name>[Request | Response]": {
        "payload": "<payload contents of <name>TBS[Request | Response]>",
        "protected": "<integrity-protected header contents>",
        "header": "<non-integrity-protected header contents>",
        "signature": "<signature contents>"
    }
}

For example:

- CreateSDRequest
- CreateSDResponse
OTrP JSON Sample Message: GetDeviceState

```
{
  "GetDeviceStateTBSRequest": {
    "ver": "1.0",
    "rid": "<Unique request ID>",
    "tid": "<transaction ID>",
    "ocspdat": "<OCSP stapling data of TSM certificate>",
    "icaocspdat": "<OCSP stapling data for TSM CA certificates>",
    "supportedsigalgs": "<comma separated signing algorithms>"
  }
}

{
  "GetDeviceStateRequest": {
    "payload": "<BASE64URL encoding of the GetDeviceStateTBSRequest JSON above>",
    "protected": "<BASE64URL encoded signing algorithm>",
    "header": {
      "x5c": "<BASE64 encoded TSM certificate chain up to the root CA certificate>"
    },
    "signature": "<signature contents signed by TSM private key>"
  }
}
```
OTrP Sample Message: CreateSD Request

```json
{
    "CreateSDTBSRequest": {
        "ver": "1.0",
        "rid": "<unique request ID>",
        "tid": "<transaction ID>", // this may be from prior message
        "tee": "<TEE routing name from the DSI for the SD's target>",
        "nextdsi": "true | false",
        "dsikey": "<hash of DSI returned in the prior query>",
        "content": ENCRYPTED { // this piece of JSON data will be encrypted
            "spid": "<SP ID value>",
            "sdname": "<SD name for the domain to be created>",
            "spcert": "<BASE64 encoded SP certificate>",
            "tismid": "<An identifiable attribute of the TSM certificate>",
            "did": "<SHA256 hash of the TEE cert>
        }
    }
}
```
OTrP Sample Message: CreateSD Response

{
  "CreateSDTBSResponse": {
    "ver": "1.0",
    "status": "<operation result>",
    "rid": "<the request ID received>",
    "tid": "<the transaction ID received>",
    "content": ENCRIPTED {
      "reason": "<failure reason detail>", // optional
      "did": "<the device id received from the request>",
      "sdname": "<SD name for the domain created>",
      "teespaik": "<TEE SP AIK public key, BASE64 encoded>",
      "dsi": "<Updated TEE state, including all SD owned by this TSM>"
    }
  }
}