The Open Trust Protocol (OTrP) v2

Hannes Tschofenig, Ming Pei, David Wheeler
Why is there a new document?

- WG decisions to
  - Remove support for security domain from the base protocol,
  - Align with SUIT for software updates,
  - Align with RATS for attestation,
  - Include CBOR serialization support (in addition to JSON),
  - Add support for multiple TEEs,
- Architecture draft made lots of text in the original OTrP draft redundant.
- Support for wider set of use cases introduced new features
- Terminology changes in the architecture draft required alignment.
OTrP and Backwards Compatibility

• With the previously introduced changes it is difficult (if not impossible) to maintain backwards compatibility.

• How important is it to maintain backwards compatibility with v1.0?

• Possible approaches:
  • New version number (approach taken in v2)
  • New name (suggested by Jeremy)
  • Something else?
Design Overview

• CDDL for describing the protocol messages
  • Description agnostic of the serialization (at least in theory)
  • Security mechanisms used with JSON and CBOR serialization will be different.

• 6 messages (TrustedAppInstall, TrustedAppDelete, Success, Error, QueryRequest, QueryResponse)

• TA software described via a SUIT manifest; same is true for personalization data. Can be signed and/or encrypted. TAs are identified with (vendor id, class id, device id).

• Common message type with TYPE, TOKEN, MSG style (with outer wrapper)

• Support for extension indication

• Attestation accomplished with EAT (with NONCE in QueryRequest for freshness guarantees)

• Tid&rid combined into a single field – NONCE.
Outer_Wrapper = {
    msg-authenc-wrapper => bstr.cbor
    Msg_AuthEnc_Wrapper / nil,
    otrp-message => (QueryRequest /
        QueryResponse /
        TrustedAppInstall /
        TrustedAppDelete /
        Error /
        Success ),
}

Msg_AuthEnc_Wrapper = [ * (COSE_Mac_Tagged /
    COSE_Sign_Tagged /
    COSE_Mac0_Tagged /
    COSE_Sign1_Tagged)]
QueryRequest = (  
  TYPE : int,  
  TOKEN : bstr,  
  REQUEST : [+data_items],  
  ? CIPHER_SUITE : [+suite],  
  ? NONCE : bstr,  
  ? VERSION : [+version],  
  ? OCSP_DATA : bstr,  
  * $$extensions  
)

suite = int

version = int

data_items = (  
  attestation: 1,  
  ta: 2,  
  ext: 3  
)
QueryResponse = (  
    TYPE : int,  
    TOKEN : bstr,  
    ? SELECTED_CIPHER_SUITE : suite,  
    ? SELECTED_VERSION : version,  
    ? EAT : bstr,  
    ? TA_LIST : [+ta_id],  
    ? EXT_LIST : [+ext_info],  
    * $$extensions
)

TrustedAppInstall = (  
    TYPE : int,  
    TOKEN : bstr,  
    ? TA : [+SUIT_Outer_Wrapper],  
    * $$extensions  
)
Success = ( 
    TYPE : int, 
    TOKEN : bstr, 
    ? MSG : tstr, 
    * $\$extensions 
)
Error = ( 
    TYPE : int,
    TOKEN : bstr,
    ERR_CODE : int,
    ? ERR_MSG : tstr,
    ? CIPHER_SUITE : [+suite],
    ? VERSION : [+version],
    * $$extensions
)

Open Issues

• How does the CDDL need to look like to support CBOR/JSON-agnostic serialization?
• Are additional fields in the message header needed for message routing by the broker?
• How is the OCSP_DATA formatted & encapsulated?
• Should the algorithm recommendation be in the spec or in a separate spec?
• Mapping to security wrappers and examples are missing.