TLS authentication using ETSI TS 103 097 and IEEE 1609.2 certificates

IETF meeting 93 – Prague – TLS WG session

Wednesday, July 22, 2015

Objective & Motivations

- **Objective**: enable C/S authentication using C-ITS* certificates

- **Motivations**: C-ITS networks are highly mobile with a limited bandwidth
  - X.509 certificates are not optimized for bandwidth and delay-sensitive applications
  - Definition of new certificates for C-ITS
    - US: IEEE 1609.2
    - EU: ETSI TS 103 097

- **Use case**: secured communication between a vehicle and a server on the Internet:
  - e.g. vehicle data upload on a remote log server
  - e.g. vehicle software update
  - e.g. traffic light information via 3G/LTE communication (SPAT)

    - Authentication between an ITS-Station and a server should be possible using C-ITS certificates

*C-ITS: Co-operative Intelligent Transportation System

Modifications on TLS Handshake

```
Client                                          Server

ClientHello
/* cert_type extension */   ----------->    ServerHello
                                            /* cert_type extension */
                                            Certificate*
                                            ServerKeyExchange*
                                            CertificateRequest*
Certificate                 <-----------    ServerHelloDone
ClientKeyExchange
CertificateVerify*
[ChangeCipherSpec]
Finished                    ----------->    [ChangeCipherSpec]
                            <-----------    Finished
Application Data            <---------->    Application Data
```

* Indicates optional or situation-dependent messages that are not always sent.

Technical modifications

- `ClientHello` and `ServerHello` messages of the handshake protocol SHALL include the `cert_type` extension [RFC 6091]

```
enum {
   X.509(0), OpenPGP(1), RawPublicKey(2),
   IEEE(TBD), ETSI(TBD), (255)
}CertificateType;
```

Technical specifications

- `CertificateRequest` SHALL be filled with the following values:

CertificateRequest as defined in [RFC 5246]

```
enum {
    rsa_sign(1), dss_sign(2), rsa_fixed_dh(3), dss_fixed_dh(4), rsa_ephemeral_dh_RESERVED(5),
dss_ephemeral_dh_RESERVED(6), fortezza_dms_RESERVED(20), ECDSA_sign(64), (255)
}ClientCertificateType;

opaque DistinguishedName<1..2^16-1>;

struct {
    ClientCertificateType certificate_types<1..2^8-1>;
    SignatureAndHashAlgorithm supported_signature_algorithms<2^16-1>;
    DistinguishedName certificate_authorities<0..2^16-1>;
}CertificateRequest;
```

Filled values

```
ClientCertificateType ECDSA_sign(64)

SignatureAndHashAlgorithm {0x04,0x03} (ECDSA-SHA256)

DistinguishedName    List of HashedId8 [ETSI TS 103 097] (The server informs the client about
        the certificate authorities it trusts to help the client to select a
corresponding certificate)
```

Thank you!

Verification of ETSI certificate

ETSI TS 103 097 certificate format:

version	signer_info	subject_info	verification /encryption keys	assurance_level	ItsAid_ssp list	validity_restrictions	signature

1. Verify that the certificate content is conform to one of ETSI profiles.

2. Verify the certificate's signer identity:

```
IF    The certificate digest included in signer_info is known:
      Go to step 3.
ELSE:
      IF    It is a root certificate digest:
            Verification failed (error – untrusted root CA).
      ELSE:
            Pause the current certificate verification process and
            start verification of the next certificate in the chain recursively by
            restarting from step 1. Once verified, resume the certificate
            verification.
```

3. Verify that the certificate is not in the Certificate Revocation List (CRL).

4. Verify the signature of the certificate (see [10] for details).

5. Verify `subject_info`: `subject_name` shall be a 32 bytes hash of the server URL. Note that this step is only done by clients. Servers shall ignore this step.

6. Verify `validity_restrictions`: only time validity is checked, space validity (geographical region) is ignored.

7. Verifity `its_aid_ssp`: ITS-AID included in the certificate shall be consistent with those included in the signer's certificate (heritage).