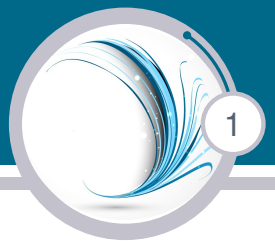
Transport Layer Security (TLS) Authentication using ITS ETSI and IEEE Certificates

IETF-103/IPWAVE Group

Télécom ParisTech & CISCO

6th Of November 2018

Plan



1. Motivations & Objective
2. Use Cases
3. Extension Overview
4. TLS Extension

Motivations

- Motivations:
  - C-ITS[1] networks are highly mobile with a limited bandwidth.

That is why size-optimized certificates were standardized by ETSI and IEEE to secure data exchange in highly dynamic vehicular environments in Intelligent Transportation System (ITS).

[1] Cooperative Intelligent Transportation System

Objective

- Objective:
  - We need an authentication method more optimized for bandwidth and processing time to support delay-sensitive applications.
  - Enable Client/Server authentication using C-ITS certificates

Use cases

- 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 [2])
  e.g. connected cloud services
  e.g. connected infotainment

[2] Signal Phase and Timing adapter

Use cases

- Authentication between an ITS-Station and a server should be possible using C- ITS certificates:
  e.g. rent company
  e.g. car manufacturer
  e.g. wireless electric vehicle charging

Extension Overview


```
/* Managed by IANA */
 enum {
     X509(0),
     RawPublicKey(2),
     1609Dot2(?), /* Number 3 will be requested for 1609.2 */
     (255)
 } CertificateType;

 struct {
     select (certificate_type) {

         /* certificate type defined in this document.*/
          case 1609Dot2:
          opaque cert_data<1..2^24-1>;

          /* RawPublicKey defined in RFC 7250*/
         case RawPublicKey:
         opaque ASN.1_subjectPublicKeyInfo<1..2^24-1>;

         /* X.509 certificate defined in RFC 5246*/
         case X.509:
         opaque cert_data<1..2^24-1>;

          };

        Extension extensions<0..2^16-1>;
    } CertificateEntry;
```

TLS Extension

Client		Server
ClientHello,		
client_certificate_type*=(1609Dot2),		
server_certificate_type*=(X509,		
RawPublicKey,16099Dot),	------->	ServerHello,
		{EncryptedExtensions}
		{client_certificate_type*=1609Dot2}
		{server_certificate_type*=X509}
		{Certificate*}
		{CertificateVerify*}
		{Finished}
	<---------	[Application Data*]
{Finished}	--------->	
[Application Data]	<-------->	[Application Data]

One new value referring the IEEE certificate is added to the client-certificate-type and the server-certificate-type as defined in RFC 8446.

Thank You!
https://tools.ietf.org/html/draft-tls-certieee1609-02

Server Hello

- The server supports the extension. It selects a certificate type from the client certificate type field in the extended Client Hello and must take into account the client authentication list priority.
- The server does not support the proposed certificate type and terminates the session with a fatal alert of type unsupported certificate.
- The server does not support the extension defined in this document. In this case, the server returns the server hello without the extensions defined in this document in case of TLS 1.2.
- The server supports the extension defined in this document, but it does not have any certificate type in common with the client. Then, the server terminates the session with a fatal alert of type unsupported certificate.
- The server supports the extensions defined in this document and has at least one certificate type in common with the client. In this case, the server MUST include the client certificate type extension in the Server Hello for TLS 1.2 or in Encrypted Extension for TLS 1.3.