Internet Draft   P. Urien
Intended status: Experimental   Telecom Paris
Expires: April 2023   October 2022

Internet of Secure Elements
draft-urien-coinrg-iose-06.txt

Abstract

This draft defines an infrastructure for secure elements over internet, and features needed for their secure remote use.
It describes a network architecture based on the TLS 1.3 protocol, which enables remote calls of cryptographic procedures, identified by Unified Resource Identifier (URI) such as
schemeS://sen@server.com:443/?query
The Internet of Secure Element (IoSE) is a set of secure elements providing TLS servers, communication interfaces, and identified by their name (Secure Element Name, sen).

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

Status of this Memo

Copyright Notice

Table of Contents

1 Overview

This draft defines an infrastructure for the deployment of secure elements over internet, and features needed for their secure remote use.

Secure elements [ISO7816] are tamper resistant micro-controllers, whose security Evaluation Assurance Levels (EAL) are in the range EAL5+/EAL6+ according to Common Criteria standards [CC], which define up to 7 levels.

This draft describes a network architecture based on the TLS 1.3 [RFC8446] protocol, which enables remote calls of cryptographic procedures, identified by Unified Resource Identifier (URI) [RFC3986].

We believe that internet should provide to its users open computing resources, with high security and trust levels. Many applications, such as blockchain, require on-line trusted computing resources, running cryptographic algorithms.

```
                     TLS SNI                        SEN
                   +--------+                +-----------------+
                   | TCP/IP | Communication |   Application   |
server.com:443---+ Server +---------------+       TLS        |
SNI=SEN            |        |   Interface   | Secure Element |
                   +--------+                +-----------------+
                   server.com                     TLS Server

                    schemeS://sen@server.com:port/?query
```

The network architecture comprises the following elements:
- Secure elements, identified by their name (Secure Element Name, SEN) running embedded TLS servers and applications.
- TCP/IP servers, able to parse TLS ClientHello message, in order to extract SNI (Server Name Indication) extension [RFC6066]. If the SNI value matches the SEN value, the TLS packets are routed toward the selected secure element.

The secure element URI [RFC3986] is schemeS://sen@server.com:443/?query, in which:

- scheme indicates the application data interchange format,
- S means secured by TLS,
- sen is the secure element name included in the TLS SNI extension,
- server.com:port is a TCP/IP node and associated port
- query is the command to be executed by the secure element

TLS sessions MUST use mutual authentication between client and server, either based either on pre-shared-key (PSK) or X509 certificates.

The TCP/IP server MAY manage multiple secure elements. As an illustration, according to the IETF draft [RACS] a grid of Secure Elements (GoSE) is a server hosting a set of secure elements.

In summary the Internet of Secure Element (IoSE) is a set of secure elements providing TLS servers, communication interfaces, and identified by their SEN name.

2. About Secure Elements

Secure elements are defined according to [ISO7816] standards. Most of them use 8 bits Micro Controller Unit (MCU) and embedded cryptographic accelerator. Non volatile memory size is up to 100KB, and RAM size is up to 10KB. Open software can be written thanks to the JavaCard (JC) programming language, and associated API frameworks such as JC3.04, JC3.05, JC3.1.

Secure elements are dedicated to cryptographic procedures; they are available under multiples physical form factors, such as smartcard, NFC chip, embedded SIM (eSIM), or surface-mount devices.

Secure elements have no network resources. They exchange small messages (up to 256 bytes) over communication interfaces such as ISO7816 (5 wires) [ISO7816], I2C (Inter-Integrated Circuit), or SPI (Serial Peripheral Interface) [GP-SPI-I2C].

Nevertheless they are able to process the TLS 1.3 protocol. For example the IETF draft [TLS-SE] defines segmentation/reassembly mechanisms over ISO7816, which enable exchange of TLS packets with secure elements. The open project [TLS-SE-CODE] is an implementation of [TLS-SE] for javacards. The open project [KEYSTORE-CODE] is an implementation of secure element server. The open project [IOSE-CODE] is a demonstrator for Internet of Secure Elements.

Therefore secure element can be used as host, providing TLS server, and communication interface.

They are several ways to provide a host name for a secure element (i.e. a server name), which is referred as secure element name (SEN) by this draft,:

- The [TLS-SE] draft uses historical bytes (up to 15 bytes) inserted in the ISO7816 ATR (Answer To Reset), which is a response triggered by a physical reset. A javacard application may define the value of historical bytes.

- The [RACS] IETF draft describes Grid of Secure Elements (GoSE), and introduces Secure Element Identifier (SEID) as unique identifier indicating that a given SE is hosted by a GoSE. SEID also implicitly refers the physical slot (SlotID) to which the secure element is plugged. SEID MAY be used as SEN.

3. Network Architecture

The network architecture is based on TLS1.3 servers and future versions.

A TCP/IP node manages a server. According to [ESNI] TLS has two working modes, shared and split.

- In Shared Mode, the provider is the origin server for all the domains whose DNS records point to it. In this mode, the TLS connection is terminated by the provider
- In Split Mode, the provider is not the origin server for private domains. Rather, the DNS records for private domains point to the provider, and the provider's server relays the connection back to the origin server, who terminates the TLS connection with the client.

According to this terminology the secure element is the backend server, identified by a server name (referred as SEN).

The client-facing server finds in the ClientHello message required secure element name. Thereafter it performs segmentation/reassembly operations in order to shuttle TLS packet over the communication interface.

The client-facing server MAY also use encrypted server name indication (ESNI) features in order to protect secure elements name.

The application-layer protocol negotiation extension (ALPN) [RFC7301] MAY be used by secure element to select an internal application.

TLS protocol MUST be used with mutual authentication between client and secure element. PSK is a symmetric cryptographic scheme for one client-to-one-secure-element, while PKI is an asymmetric cryptographic scheme adapted to multiple-clients-to-one-secure-element.

Nevertheless it should be noticed that secure elements have not clock and therefore are not able to check validity date or certificate revocation.

4 Unified Resource Identifier (URI)

According to [RFC3986] the URI comprises a scheme name ended by the 'S' character, the secure element name, the client-facing name and port (server.com:port), and a query.

```
URI= schemeS://sen@server.com:port/?query
```

A client software entity able to process this URI, MUST retrieves the PSK or the certificate chain to be used within the TLS protocol.

The secure element name MUST be included in the SNI extension.
The used scheme used by the query, MAY be included in the ALPN extension.

For PSK it is possible, but not recommended for security reasons, to include the PSK value in the URI:

```
schemeS://sen:psk@server.com:port/?query
```

5 URI Example

A secure element implements a keystore, of which keys are identified by an index. The secure element name is mykeystore

The secure element name is found in the historical bytes of the ISO7816 ATR.

The client-facing server is server.com:443

The scheme used by the secure element is a shell, i.e. ASCII command lines ended by line feed and carriage return characters.

The query s0101020304050607080900A0B0C0D0E0F101112131415161718191A1B1C1D1E1F20%0D%0A computes a signature command ('s' prefix) with key of index 01, over the 32 bytes value 0102...1920

The URI is:

```
shellS://mykeystore@server.com:443/?s0101020304050607080900A0B0C0D0E0
F101112131415161718191A1B1C1D1E1F20%0D%0A
```

The software client opens a TLS session with the server server.com:443, with the name "mykeystore" inserted the SNI extension. Upon success a TLS secure channel is established with the secure element. The client sends the query, the secure element computes the signature and returns its value encoded in hexadecimal text.

6 Overview of Internet Of Secure Elements Framework

```
     +----------------------------------+
     |         User Application         |
     +----------------------------------+
     |              APIs                |
     +----------------------------------+
     |          TCP/IP Client           |
     +----------------------------------+   +-----------------+
     |          TCP/IP Server           |<--|                 |
     +----------------------------------+   | Administration |
     |   Secure Element Application     |<--|     (RACS)      |
     +----------------------------------+   +-----------------+
     |     Secure Element Hardware      |
     +----------------------------------+
```

The goal of IOSE is to provide to internet users open computing resources, with high security and trust levels. In order to reach this objective, the IOSE framework comprises seven layers.

- The User Application layer uses secure resources hosted in the internet
- The APIs layer provides software interface to virtual resources. It SHOULD provide secure storage of credentials required by TLS sessions.
- The TCP/IP client layer manages TLS session, according to profiles compatible with secure element computing capacities.
- The TCP/IP server layer manages one or several secure elements. It MAY provide privacy features such as server name encryption.
- The secure element application layer defines data interchange format and available procedures
- The secure element hardware layer defines security profile (according to Common Criteria standards) and communication interfaces
- The administration layer is in charge of secure elements application deployment and lifetime. These operations are performed locally or remotely (through the internet).

7 Functional Entities

```
           +---------------------------+    +-------------------+
           |                           |    |                   |
           |  Infrastructure Provider  +----+  SE-App Provider  |
           |  (Secure Element Server)  |    |                   |
           |                           |    +------+-----+------+
           +------------+--------------+           |     |
                        |                          |     |
                        |                          |     |
           +------------+--------------+           |     |
           |                           +-----------+     |
           |      Service Provider     |                 |
           |   (Cloud Infrastructure)  |                 |
           |                           +-------+  +------+
           +------------+--------------+       |  |
                        |                      |  |
                        |                      |  |
           +------------+--------------+    +--+--+--+
           |                           |    |        |
           |    Application Provider   +----+  User  +
           |                           |    |        |
           +---------------------------+    +--------+
```

The functional entities COULD involve five elements.

- The User is equipped with a connected device, executing an application using IOSE services.

- The Application Provider (AP) designs software, using IOSE infrastructure.

- The Service Provider (SP) manages a cloud infrastructure, and all facilities needed to setup secure element applications. An attestation mechanism MUST be available in order to prove SE application authenticity.

- The Infrastructure Provider (IP) provides secure element servers.

- The SE-App Provider (SE-AppP) designs secure and trusted software (SE-App) for secure elements.

8 Attestation Procedure

The goal of the attestation procedure is to allocate a secure element, and to prove to its user the exclusive access to a genuine secure element.

```
  SE-App Provider    Infrastructure   Secure
     Provider           Provider      Element
       |                    |             |
       | RACS: Download --->|------------>|Private and Public Key
       | TLS-PSK App in SE  |             |Generation
       | With PSK-Provider  |             |TLS-PSK Ready
       | <--------------Done|<-------Done|
       |                    |             |
       | RACS: Link  ------>|             |
       | SEN to SEID        |             |
       | <--------------Done|             |
       |                    |             |
       | TLS-PSK:  ---------|------------>|
       | Read Public Key    |             |
       | <------------------|--------Done|
       |                    |             |
       | Generate           |             |
       | Certificate        |             |
       |                    |             |
       | TLS-PSK:    -------|------------>|
       | Write Certificate  |             |
       | <------------------|------- Done|
       |
       | SEND --------------------------------------------------->|
       | PSK-Provider & SEN                                       |
                                      Secure                  User
                                      Element                    |
                                         |<---------Open TLS-PSK|
                                         |Compute        Compute|
                                         |HS                  HS|
                                         |<---TLS-PSK Opened--->|
                                         |                      |
                                         |<------Read Public Key|
                                         |Done----------------->|
                                         |                      |
                                         |<-----Read Certificate|
                                         |Done----------------->|
                                         |                      |
                                         |     Check Certificate|
                                         |                      |
                                         |<-------------Send rnd|
                                         |Compute               |
                                         |Sign(HS | rnd)        |
                                         |Send Sign------------>|
                                         |                      |
                                         |            Check Sign|
                                         |                      |
                                         |<-------Write PSK-User|
                                      |Done----------------->|
```

8.1 Service Request

The User requests to an on-line Service Provider a secure element for a specific application.

8.2 SE-App Downloading

The Service Provider requests the SE-App Provider to download the user's application in a secure element hosted by an Infrastructure Provider.

The SE-App Provider downloads the selected application thanks to protocols such as [RACS].

There is a mutual authentication between SE-App Provider and Infrastructure Provider. The SE-App Provider owns a set of secure elements identified by their SEID (Secure Element Identifier). He may erase their content and write binary image, but he can't read binary images.

The SE-App includes a TLS 1.3 server with a pre shared key (PSK-App-Provider) and a server name (SEN). Upon instantiation, the downloaded SE-App generates a pair of private key (SE-App-Priv-Key) and public key (SE-App-Pub-key).

Thanks to a dedicated [RACS] command, the SE-APP provider notifies to the Infrastructure Provider the SEN associated to the secure element SEID.

At this step the secure element is on-line, and may process TLS sessions, with the right server name (SEN), and authenticated by PSK-App-Provider.

The PSK-App provider has the exclusive knowledge of the pre-shared-key, and consequently is the only entity able to communicate with the secure element.

8.3 SE-App Certificate

The SE-App Provider opens a TLS (with PSK= PSK-App-Provider) session with the SEN secure element, reads its SE-App-Pub-key, and computes a certificate (SE-Cert) for this public key.

The SE-Cert is remotely written in the secure element

The SE-App provider forwards the secure element URI and PSK-App-Provider to the Service Provider or to the User, according to pre-defined agreements.

8.4 User Notification

The User receives the secure element URI and pre-shared-key (i.e. PSK-App-Provider).

8.5 User Enrollment

A secure element only manages a unique TLS session at a given time.

The User opens a TLS session with the secure element (with PSK-App-Provider and SEN). According to [RFC8446] a TLS handshake secret (HS) is computed from the Diffie-Hellman exchange.

He reads the secure element public key (SE-App-Pub-key) and its certificate (SE-Cert)

He checks the certificate SE-Cert with the Infrastructure Provider public key.

He performs the attestation procedure that sends a random value (RND) to the secure element. The secure element returns a signature of the concatenation of HS and RND values, computed with the private key SE-App-Priv-key.

He checks the signature with the public key SE-App-Pub-key.

At this point the User has the proof that it shares an exclusive TLS session with the secure element.

The user sets the pre-share-key value to new one: PSK-User-Provider.

At this step the User has the exclusive access to a genuine secure element.

9 IANA Considerations

This draft does not require any action from IANA.

10 Security Considerations

This entire document is about security.

11 References

11.1 Normative References

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, https://www.rfc-editor.org/info/rfc8446.

[RFC6066] Eastlake 3rd, D., "Transport Layer Security (TLS) Extensions: Extension Definitions", RFC 6066, DOI 10.17487/RFC6066, January 2011.

[RFC7301] Friedl, S., Popov, A., Langley, A., and E. Stephan, "Transport Layer Security (TLS) Application-Layer Protocol Negotiation Extension", RFC 7301, July 2014

[RFC3986] Berners-Lee, Tim; Fielding, Roy T.; Masinter, Larry. Uniform Resource Identifiers (URI): Generic Syntax. Internet Engineering Task Force. doi:10.17487/RFC3986, January 2005

[ISO7816] ISO 7816, "Cards Identification - Integrated Circuit Cards with Contacts", The International Organization for Standardization (ISO).

[CC] ISO/IEC 15408, "Common Criteria for Information Technology Security Evaluation", The International Organization for Standardization (ISO)

[GP-SPI-I2C] GlobalPlatform Technology, APDU Transport over SPI/I2C Version 0.0.0.39", July 2019

11.2 Informative References

[ESNI] "TLS Encrypted Client Hello", draft-ietf-tls-esni-13, 2021

[RACS] "Remote APDU Call Secure (RACS)", draft-urien-core-racs-15.txt, 2021

[TLS-SE] IETF Draft, "Secure Element for TLS Version 1.3", draft-urien-tls-se-03.txt, 2021

[TLS-SE-CODE] "tls-se.java", https://github.com/purien/TLS-SE

[KEYSTORE-CODE] https://github.com/purien/keystore

[IOSE-CODE] https://github.com/purien/iose

12 Authors' Addresses

Pascal Urien
Telecom Paris
19 place Marguerite Perey
91120 Palaiseau   Phone: NA
France   Email: Pascal.Urien@telecom-paris.fr