CORE Working Group
Internet Draft
Intended status: Experimental

P. Urien
Telecom Paris

February 4 2022

Expires: August 2022

TLS for Secure Element Input Output (TLS-SE-IO)
draft-urien-core-tls-se-io-01.txt

Abstract

The goal of TLS-SE-IO is to provide virtual IO pins for secure elements running TLS servers, in order to interact with sensors and actuators. TLS-SE device processes TLS packets in secure element. It may work like a black box (server mode) that exchanges fully encrypted packets. It may also export encrypted packet in clear form, in order to provide virtual output pin. Output messages may include cookies and/or cryptographic materials. Virtual input pin forwards input messages, triggered by previous output messages, and sent to TLS-SE device for further processing.

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

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

Abstract.. 1
Requirements Language... 1
Status of this Memo... 1
Copyright Notice.. 2
1 Overview.. 4
2 TLS-SE-IO... 4
3 TLS-SE-IO Protocol.. 5
3.1 Output messages... 5
3.2 Input message... 6
4 Example... 6
5 Security Considerations....................................... 8
6 IANA Considerations... 8
7 References.. 8
7.1 Normative References.. 8
7.2 Informative References...................................... 9
8 Authors' Addresses.. 9

1 Overview

Input output (IO) interfaces are used in the internet of things context (IoT) to manage sensors and actuators.

Output pin has two binary states, one or zero, and can generate a bit stream, i.e. messages comprising a set of bytes.

Input pin detects two binary states, one or zero, which can realize a bit stream, i.e. messages comprising a set of bytes.

Usually input messages are triggered by previous output messages.

Secure elements, mainly specified by [ISO7816] standards have multiple form factors, such as SIM card, or surface mounted device (SMD). They provide tamper resistant computing resources. According to Common Criteria (CC) standards, their Evaluation Assurance Level ranges between EAL4 to EAL6+, EAL7 being the highest level.

Nevertheless secure elements have no IO pins, and are not able to physically communicate with sensors and actuators. However they may be connected to processors, with physical IO capacities (i.e. equipped with IO pins)

This document describes the processing of output and input messages by secure elements that execute TLS server. Output messages are exported from the secure element in clear form; they provide an output byte stream. Input messages are triggered by output messages; a byte stream is forwarded to secure element for further processing.

2 TLS-SE-IO

The draft [TLS-SE] defines [TLS 1.3] support for secure elements. Two procedures RECV and SEND realize a logical bridge between TLS packets and [ISO7816] messages.

```
        +-----------+ recv +------------+ RECV +-----------+
        |     IP    + ---> |   TCP/IP   | ---> |  TLS 1.3  |
        |  Network  +------+ Constraint +------+  Secure   |
        |           | <--- |    Node    | <--- |  Element  |
        +-----------+ send +------------+ SEND +-----------+
                       |                    |
                Network Interface    ISO7816 interface
```

A processor physically connected to secure element (the secure element processor, SEP) can read TLS packets transparently transported by ISO7816 requests and responses. It also knows if the secure channel is opened, thanks to a dedicated ISO7816 status word (sw-open = 0x9001)

```
              +------------+ TLS-Packet +-----------+
              |            | ---------> |  TLS 1.3  |
              | Processor  +------------+  Secure   |
              |   (SEP)    | <--------_ |  Element  |
              +------------+ TLS-Packet +-----------+
               SE Processor             Secure Element
```

TLS-SE provides two classes of service:

- TLS-SE as server, this is the technological basis for [IOSE] framework. The secure element is a black box providing secure storage and tamper resistant computing resources.

- TLS-SE as stack. The secure element fully processes TLS session opening, i.e. TLS flights. It provides TLS packets encryption (TLS-Encrypt) and decryption (TLS-Decrypt) procedures.

```
        +------------+                     +------------+
        |            +-<----- Output ----<-+-+          |
        |            |                     | |Decryption|
        |            +-- Rx-TLS-Packets ->-+-+          |
        |    SEP     |                     |   TLS-SE   |
        |            +-<- Tx-TLS-Packets <-+-+          |
        |            |                     | |Encryption|
        |            +->------Input ----->-+-+          |
        +------------+                     +------------+
```

The main idea of TLS-SE-IO is to provide virtual Input/Output (IO) resources (i.e. virtual IO pins) to TLS-SE secure element.

- Output requests MUST be received in encrypted TLS record messages. Clear messages are returned by secure element.

- Input messages are triggered by output requests. They MUST be encrypted by the secure element thanks to the SEND procedure described in [TLS-SE]

3 TLS-SE-IO Protocol

The SEP entity can read incoming and outgoing TLS packets.

A TLS record packet has a five bytes header in clear form, which comprises 3 fields, type (one byte), version (2 bytes), and length (2 bytes)

3.1 Output messages

An output message is received in encrypted TLS record packet. It is decrypted within TLS-SE secure element. It MUST contain an attribute ('output-mark') that triggers the output message exportation.

The TLS-SE secure element produces a TLS packet with the first three bytes (type and version) set to zero, and a length. The payload comprises the data and the TLS type in clear form.

```
      Type=0x00  Version=0x0000  Length  output-message  TLS-Type
```

An output message MAY be encrypted. It MAY also contains cookie to be used in input messages.

Output messages are processed by SEP device. They contain any kind of information, such as object or data serialization, script, or AT commands for cellular serial modems.

3.2 Input message

Input message is triggered by a previous output message. It MAY be encrypted. It MAY also contains cookie found in a previous output message.

Input message is sent to the secure element thanks to the procedure RECV(F-Encrypt,input-message), as specified in[TLS-SE]. It MAY trigger a TLS record packet, according to the RECV procedure.

4 Example

```
PSK=
0102030405060708090A0B0C0D0E0F101112131415161718191A1B1C1D1E1F20
DHE=
45A4CB06906CD3426E9F8E02FD0EAA39E016729A8F00D08E34B907418723007E

TLS Reset
Tx: 00D8000000
Rx: 9000

RECV(F-First,ClientHello)
Tx: 00D80001F0 1603030103010000FF0303
    749F4A35B3E76B4554517F221F8BD54C
    C7DD96E9B70A0DBF4652821AD8AF095C
    000002130401000D4002D0003020001
    002B0003020304000D001E001C060305
    03040302030806080B0805080A080408
    09060105010401020100330047004500
    170041047ED91D5E7DC92EBF5D26444B
    A267299D8D5A89481C121E7691A7D782
    77668F366DEC65B6B35D707777F523C0
    05B48CC7165EC151E8D9A28F22B8603B
    14C11975000A00060004001800170000
    000D000B0000086B6579312E636F6D00
    29003A0015000F436C69656E745F6964
    656E7469747900000000002120F3793B
    B581E86684
    9000

RECV(F-Last,ClientHello))
Tx: 00D8000218 6F72919312AEF766B56275
    B26573A433A8BD1806A492620F
Rx: 6186

SEND(ServerHello)
Tx: 00C0000086
Rx: 1603030081 0200007D03030BBDF53C07
    596AACAF7724DB911E11C92F418ED96C
    008451A49E7AE08230B3D10013040000
    55002900020000003300450017004104
    07A637DCCBF63DE1A6EFB59ADDB796FE
    BC9106A96379081BC3547FB42C7982B7
    A8FA04A0F7E3F1784A1A0086CBCC03BE
    F8FDE7526EED3DB4F85DAF5BD26443E2
    002B00020304
    9F1C

SEND(ServerEncryptedExtension)
Tx: 00C000001C
Rx: 1703030017CD418DE7D2E6E8F393A5AC
    B0E4E2C06BFA0B0631C59A26
    9F3A

SEND(ServerFinished)
Tx: 00C000003A
Rx: 17030300350F106E7DB08E7CCB69644D
    7E0F9CB39FB2B5A0AC0D36FA462A9E40
    0517A548E7F9E07191ECC1F869671E3B
    1F1B39D9A38E09EE6DE8
    9000

RECV(F-First||F-Last, ClientFinished)
Tx: 00D800033A 1703030035C407E727ACBC
    7CE96EB6A81391FF8F8546976430B8F8
    65A1C8A41F279B7B4A72934A0225021B
    A4001793EBFFC2167FAE250A5B69A8
Rx: 9001

TLS Secure Channel is open

Encrypted opaque message

RECV(F-First||F-Last, TLS-Record-Message)
Tx: 00D800031F170303001AE4EBF10433EB
```

```
    78B4454D7ACE9EBBCB74455F232EC1A6
    A6D046B5
```

```
   Rx: 611C


   Encrypted opaque response
   SEND(TLS-Record-Message)
   Tx: 00C000001C
   Rx: 1703030017215C9D0932B76BBCD439C4
       5D5FEDC4C4A4253A3EC736E7
       9000

   Encrypted output message
   RECV(F-First||F-Last, TLS-Record-Message)
   Tx: 00D800031D 1703030018AEBBB88F858C
       C8325E85C75FF1C95FEFA2F5D3BBD1D3
       C86B
   Rx: 610D

   Output message in clear form (value=#tempCrLf, Type=0x17)
   Character '#' is the output mark
   Tx: 00C000000D
   Rx: 0000000008 2374656D700D0A 17
       9000

   An input message is triggered by the output message
   Input message in clear form
   RECV(F-Encrypt||F-First||F-Last, InputMessage=18.81CrLf, Type=0x17)
   Tx: 00D8020308 31382E38310D0A 17
   Encrypted TLS record packet
   Rx: 611D
   Tx: 00C000001D
   Rx: 1703030018C14E29429EA6F071D13FB8
       8653C7ABE7315423E9D1A2B58B
       9000
```

5 Security Considerations

This entire document is about security.

6 IANA Considerations

This draft does not require any action from IANA.

7 References

7.1 Normative References

[TLS 1.3] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, August 2018.

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

7.2 Informative References

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

[IOSE] IETF Draft, "Internet of Secure Elements", draft-urien-coinrg-iose-04.txt, 2021

8 Authors' Addresses

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