3GPP TSG SA WG3 Security — S3#33 **S3-040272**
10 - 14 May 2004
Beijing, China

Source: Intel Corporation

Contact: Selim Aissi selim.aissi@intel.com

Contributing Companies: Intel, Gemplus

Title: Use of a Trusted Tunnel to Secure Local Terminal Interfaces

Document for: Discussion

Agenda Item: TBA

Abstract

Bluetooth security has been criticised on several occasions and in several research papers during the last two years [Bluetooth-SEC]. Also, in light of the several SA3 submissions that discussed various WLAN and SIM-Reuse scenarios (e.g., [S3-040163], [SIM-WLAN-THREAT]), it is clear that Bluetooth security is inadequate for any serious, security-sensitive applications in the 3G-WLAN interworking environment.

This submission proposes the use of a TLS-based Trusted Tunnel, which can provide the adequate level of protection required for several use scenarios that require a secure Local Terminal Interface, including Bluetooth as well as other types of local transport protocols.

1 Introduction

1.1 Background

The insecurity of applications in open PC platforms due to viruses, worms and other similar attacks are well known today. The security issues with using secure hard credentials such as SIM cards/UICCs, Smart cards and similar security tokens on such platforms without running the risk of being compromised is serious both from a business perspective as well as from a user privacy and liability perspective. This especially true because some of these credential access protocols were designed for less hostile environments than the ones in use today and hence require enhancements for dealing with new security threats [SEC-THREAT].

Presently there is significant industry interest in using a GSM SIM or USIM for authenticating a WLAN subscriber using a Laptop PC platform or other mobile computing platform. As there are security issues in enabling such functionality [SIM-WLAN-THREAT], this document is intended to address that problem using an adaptation of the Transport Layer Security Protocol (TLS) [RFC2246]. Though the immediate problem is to secure a WLAN application of UICC, the protocol definition is not restricted to those specific uses alone. Therefore, throughout this document there is an attempt to be as generic as possible. However, we use the WLAN-SIM application as the example of a Credential Application for our discussion throughout this document due to its immediate relevance, where WLAN-SIM denotes a SIM card/UICC that is enhanced for WLAN security for use with the EAP-SIM protocol. Other variations of the Credential Application may also be used in practice.

1.2 Overview

This document defines a generic protocol for securing the interface between a UICC [UICC] and a PC which we refer to as the ‘Terminal’. The UICC is of specific interest as it is widely in use for the millions of GSM [SIM], USIM [USIM] and CDMA [R-UIM] cards in use today and will be in the future. Hereafter we use the term UICC to encompass all variants of such cards that are compliant to ISO 7816 Part 4 [APDU] and ETSI 102 221 specifications

[UICC]. We use the term ‘Terminal’ to refer to PCs and any other computing platform that can implement this document.

This protocol for securing the UICC-Terminal interface is based on the Transport Layer Security (TLS) Protocol and is designed to provide the same security properties as TLS. Hence for the protocol security constructs the reader is requested to refer to the TLS protocol described in RFC 2246 and AES ciphersuites for TLS described in RFC 3268 for computation of cryptographic values used in the protocol. In this document, this adaptation is called Local Interface Trusted Tunnel, or LITT, to reflect the specific characteristics of the application that is targeted. This document defines protection of APDUs for any application on the UICC as long as LITT functionality is implemented on the UICC. The design is such that the TLS record layer also carries APDUs inside it, so it allows UICC credential applications to be protected transparently.

The following figure gives the basic concept behind LITT. A possible configuration is for the UICC to behave as the protocol client and the Terminal to behave as the server. The Credential Application (for example a WLAN-SIM Application) residing on the UICC is required to be accessed by the Host Application (for example an EAP-SIM supplicant) running on the Terminal. This is accomplished by first establishing an LITT session between the Terminal and the UICC. This requires authentication to be performed between the UICC and the Terminal. Mutual authentication provides a stronger protection against Man-in-the-Middle (MiM) attacks. Thereafter, normal credential data is accessed from the UICC credential application by the Host Application over the LITT protected channel. From now on, we use the WLAN or WLAN-SIM application to refer to the generic credential application for simplicity.

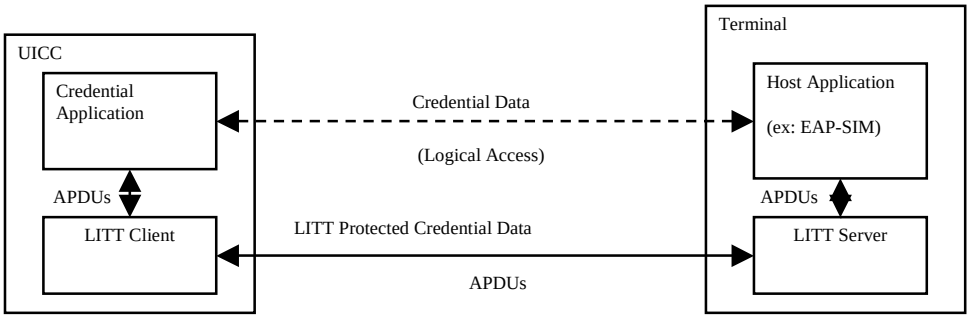

Figure 1: LITT Protection for UICC Credential Application Access

2 Scope

This document describes the details of the LITT protocol architecture on the UICC and also the corresponding functionality that is needed on the Terminal.

For the UICC side, it specifically describes the following:

- High level architecture of the client or server component on the UICC;
- APDUs used for communication with the Terminal;
- File structure requirements for the LITT component.

For the Terminal side, it specifically describes the following:

- Functionality of the server or client component;
- APDUs used for communication with the UICC.

The following is also discussed:

- Secure encapsulation of an example UICC application (i.e. WLAN-SIM Application) data.

3 Protocol Design Concepts

This section captures the precepts and the related assumptions we are making in the LITT protocol design.

3.1 UICC Interface Aspects

On the UICC interface we expect the following requirements to be supported:

- ISO 7816 Part 4 APDUs;
- Compatibility with ETSI TS 102 221 (v 4.3.0 or equivalent) required;
- T=0 protocol support required;
- Mapping from C-APDUs to C-TPDUs.

3.2 Terminal Interface Aspects

The Terminal should support ISO 7816 Part 4 APDUs and also UICC-Terminal Interface APDUs specified in ETSI TS 102 221 (v4.3.0 or equivalent). The design of LITT does not restrict the use of a physical APDU interface directly. If UICCs are embedded inside GPRS card modules or accessible remotely over a Bluetooth Local interface, LITT can still function over such UICC access methods as long as the underlying transport provides reliable message delivery.

3.3 Basic Operation

The UICC is assumed to have an LITT component, either as an applet or more likely a library as part of an applet that is capable of performing the LITT protocol with the terminal. The terminal is assumed have an LITT component..

The server sets up one or more channel sessions (logical channel based) for transferring APDUs between the card and the terminal within a card session [UICC]. The security protocol described in this document is intended to protect an entire channel session. It is possible for different channel sessions to operate in different security contexts. The idea is to use LITT to provide a secure channel session within a single security context.

The secure channel session can be maintained as long as the channel session (logical channel) is maintained. The secure channel session is intended to be used with one or more first level applications (ex: SIM, USIM specified in EF_{DIR}). Any second level application (selected with an Application Identifier, or AID) can also avail the secure channel session. Presently, for simplicity the document assumes a single credential application (e.g., WLAN-SIM) using the LITT protocol library on the UICC to securely interact with the Terminal. When multi-application cards using multiple logical channels use LITT, the protocol allows protecting a single application in a single logical channel using a specific instance of the protocol. So, if all the applications on the card require protection there needs to be as many instances of the protocol as the number of logical channels and applications used.

The secure channel session is setup with the authenticated mode of TLS. Mutual authentication should be used for adequate security as MiM and other types of attacks are possible when no authentication or one-way authentication is performed.

UICC Requirements: If UICC authentication is necessary, the UICC should possess a unique card certificate issued by a CA that is trusted by the terminal. If terminal authentication is necessary, the UICC should store a CA public key, where this CA is the one that issued the terminal public key certificate.

Terminal Requirements: If terminal authentication is necessary, the terminal should also possess a unique certificate that is issued by a CA trusted by the UICC. If UICC authentication is necessary, the terminal should store a CA public key, where this CA is the one that issued the UICC public key certificate.

It is possible for a CA to certify both UICC and terminal public keys, in which case the same CA public key may be stored both on the UICC and the terminal for certificate verification.

Figure 2 captures the general architecture of an LITT-enabled UICC. The APDUs to/from the terminal are handled first by the LITT Module (client role), before they are unwrapped and delivered to the Credential Application.

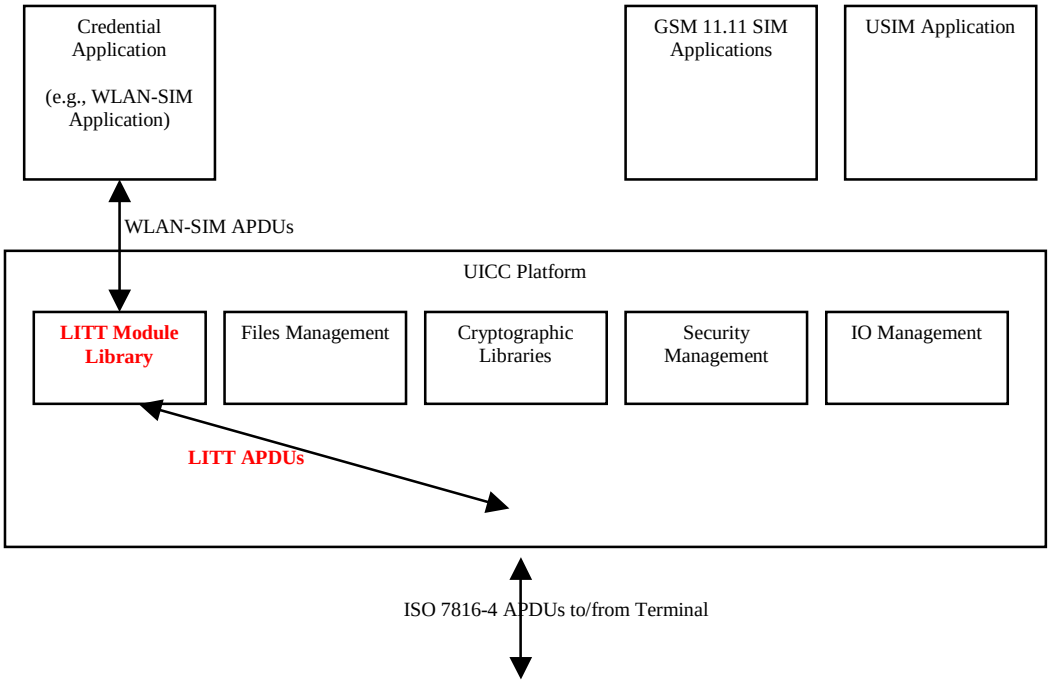

Figure 2: Basic Architecture of LITT on the UICC

The basic protocol encapsulation model is as shown in the figure 3. The application data which is also in APDU form is encapsulated within TLS.

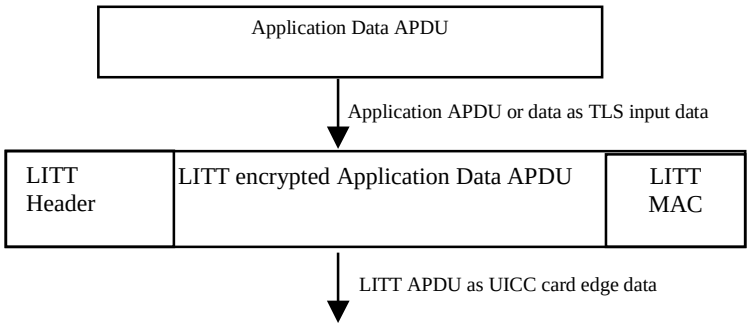

Figure 3: Encapsulation of Application APDU in LITT

3.4 Supported TLS Cipher Suites

For now we only define the use of the following cipher suites defined in [RFC 2246] and [RFC 3268]:

CipherSuite TLS_NULL_WITH_NULL_NULL = {0x00, 0x00};

CipherSuite TLS_RSA_WITH_AES_128_CBC_SHA = {0x00, 0x2F};

CipherSuite TLS_RSA_WITH_AES_256_CBC_SHA = {0x00, 0x35};

CipherSuite TLS_RSA_WITH_3DES_EDE_CBC_SHA = {0x00, 0x0A};

Support for RSA as the key exchange algorithm is recommended, as it is widely implemented and available in the industry.

The reasons for choosing AES as the block cipher are:

- AES is the current industry standard for block ciphers in the market place with several implementations available;
- AES has no IPR encumbrances;
- The immediate application of interest, EAP-SIM [EAP-SIM], uses AES and hence, the AES CBC code block can be re-used;
- Java Card 2.2 supports AES cipher suites.

The reason for supporting 3DES is for compatibility with legacy applications.

3.5 Certificate Handling

The LITT protocol can use credential certificates or authorization certificates. But the key requirement for these certificates is that they provide for authentication of the UICC-Terminal communication link.

The Terminal and the UICC may use different certificate formats for performance reasons. The terminal certificate may be based on the Card Verifiable Certificate format described in section 14.7 of the [CV-CERT] specification. These use RSA signature algorithms and the data elements are encoded using Tag-Length-Values. For details on the certificate data elements, the reader is referred to the [CV-CERT] reference document. The terminal certificate may also be based on the X.509v3 certificate format. The UICC certificate may be based on a profile of the X.509v3 certificate format and the base 64 encoded PEM files. Please refer to the Profile of the X.509v3 specified in [RFC 2459] and encoding rules specified in [RFC 1421].

The exact certificate format details and signature verification details are beyond the scope of this document as long as the LITT messages for sending and receiving a certificate is utilized and appropriate signature verification is performed and status indicated when errors are encountered.

Assuming a simplified PKI model, support for certificate chains up to 3 levels may be required for certain applications. The details of the PKI model are based on deployment considerations and are not currently addressed by this document. However, no certificate revocation capability is assumed. All the certificates used in LITT have no revocation capability. Hence their scope by definition is fairly restricted and is only intended for securing the communication channel between the Terminal and the UICC.

3.6 Shared-Key Exchange Alternative

Besides the use of certificates, TLS includes descriptions that other authentication and key exchange mechanisms can be employed. While the full TLS Handshake Protocol requires the use of Public Key Certificates (at the client and the server) to accomplish mutual authentication and Master Secret generation, LITT does not require Public Key Certificates.

Proposals generated within the IETF for using the TLS “Session Resumption” mechanism allow a session to be established without the expensive part of the certificate based handshake based on public key cryptography. For example, [TLSKEYS] describes the use of the TLS session resumption capability to set up a protected tunnel without the extra overhead required in certificate-based session initialisation.

Other methods based on the use of a shared secret to generate the Master Secret may also be used with LITT.

4 Protocol Overview

4.1 UICC-Terminal Roles

If UICC authentication is necessary, the UICC should possess a unique card certificate issued by a CA that is trusted by the terminal. If terminal authentication is necessary, the UICC should store a CA public key, where this CA is the one that issued the terminal public key certificate.

If terminal authentication is necessary, the terminal should also possess a unique certificate that is issued by a CA trusted by the UICC. If UICC authentication is necessary, the terminal should store a CA public key, where this CA is the one that issued the UICC public key certificate.

The protocol does not describe certificate validation procedures, but it expects validation to be performed both on the UICC and the Terminal and appropriate status indicated if errors are encountered. However, as the protocol only supports RSA based key exchange, RSA public key based certificates are assumed.

4.2 TLS adaptation

The adaptation of TLS for the LITT protocol targeted for the APDU transport preserves the following:

- The usage model of individual cipher suites;
- The complete key derivation and cryptographic procedures for TLS [RFC 2246];
- The TLS protocol message set.

The following are the changes when compared to TLS:

- Only a subset of cipher suites is supported;
- TLS mandates X509v3 certificates but this adaptation allows for variations of both CVC and X509v3 formats;
- TLS client and server namely the UICC and the Terminal can use different certificate formats for performance reasons.

5 UICC Requirements

5.1 General Requirements

The UICC should have a good source of randomness for generating random numbers.

5.2 Certificate Related Requirements

The UICC should support certificate chains up to 3 levels. There is no support required for handling certificate revocation. The minimum public key size that should be supported is 1024bits. The certificate signature algorithm that should be supported is RSA SHA1, or optionally MD5 RSA.

5.3 CPU Requirements

As the key cryptographic blocks are AES, 3DES, MD5, SHA and RSA public/private key operations, hardware-support would be appropriate for performance reasons, but this is not mandatory. For RSA, only support for 1024 bit public key size is expected. For AES, supporting up to 256 bits is desirable, but a minimum of 128 bits at least should be supported.

5.4 Memory Requirements

The memory requirements for the LITT protocol are still being estimated. Currently, the assumption is that the credential application along with the LITT library will fit on a 64kB-EEPROM-based UICC.

6 Security Considerations

The primary goal of the LITT is to secure the end to end communication between a UICC and a Terminal.

The Credential and LITT applications running on the UICC are protected since the UICC is a tamper resistant device.

However, the Host and LITT applications running on the Terminal need the necessary protection against software attacks during execution, since the PC environment is open. A possible way to mitigate the risks of software-based attacks is to run the protocol in a protected environment that is resistant to tampering by malicious software.

7 Conclusion

This TLS-based Trusted Tunnel could provide the adequate level of protection required for several use scenarios that require secure Local Terminal Interfaces, including Bluetooth as well as other types of local transport protocols.

We kindly ask SA3 to take into account this proposal for further security discussions.

8 References

[Bluetooth-SEC]

M. Jakobsson, S. Wetzel: “Security Weaknesses in Bluetooth”, Proceedings of the RSA Conference 2001, San Francisco, USA, April 8 – 12, 2001, Springer Lecture Notes in Computer Science Vol. 2020, ISBN 3-540-41898-9. http://www.rsasecurity.com/rsalabs/staff/bios/mjakobsson/bluetooth/bluetooth.pdf.

T. Karygiannis and L. Owens, “Wireless Network Security, 802.11, Bluetooth and handheld devices", NIST special publication 800-48, November 2002.

Vainio, Juha, “Bluetooth Security”, May 2000, at http://www.niksula.cs.hut.fi/~jiitv/bluesec.html.

D. Kügler, "Preventing Tracking and "Man in the Middle" Attacks on Bluetooth Devices", Proceedings of Financial Cryptography ’03.

[S3-040163] Orange, “A man-in-the-middle attack using Bluetooth in a WLAN interworking environment”.

[SEC-THREAT]

Real Time Cryptanalysis of A5/1 on a PC (Biryukov, Shamir & Wagner) http://www.cryptome.org/a51-bsw.htm.

Status of GSM Crypto Attacks www.chiark.greenend.org.uk/pipermail/ukcrypto/1998-October/002552.html.

GSM Cloning (Cracking COMP128) www.isaac.cs.berkeley.edu/isaac/gsm.html.

IBM research news www.research.ibm.com/resources/news/20020507_simcard.shtml.

Partitioning Attacks: Or How to Rapidly Clone Some GSM Cards www.research.ibm.com/intsec/gsm.html.

[SIM-WLAN-THREAT] 3GPP TR 33.817, Technical Report on (U)SIM Security Reuse by Peripheral Devices on Local Interfaces (Release 6), March 2004.

[RFC2246] IETF RFC 2246, “Transport Layer Security Protocol 1.0”, January 1999.

[APDU] ISO 7816-4 Smart Card Standard: Part 4: Inter-industry Commands for Interchange, 1995.

[APDU-A] ISO 7816-4 Smart Card Standard Part 4: Amendment 1, 1997.

[UICC] ETSI Technical Specification 102 221 v4.3.0, “Smart cards; UICC-Terminal interface; Physical and logical characteristics (Release 4)”, July 2001.

[EAP-SIM] Haverinen, H., Salowey, J. “EAP SIM Authentication”, draft-haverinen-pppext-eap-sim-12.txt.

[RFC 3268] IETF RFC 3268, “Advanced Encryption Standard (AES) Ciphersuites for Transport Layer Security (TLS)”, June 2002.

[SIM] 3GPP TS 51.011, “Specification of the Subscriber Identity Module - Mobile Equipment (SIM-ME) interface”.

[USIM] 3GPP TS 31.102, “Characteristics of the USIM Application (Release 6)”, March 2003.

[R-UIM] 3GPP2 C.S0023-A Removable User Identity Module for spread spectrum systems, v1.0, Sept 13, 2002.

[CV-CERT] Application Interface for smart cards used as Secure Signature Creation Devices – Part 1 Basic Requirements Version 1.07; 10 July 2003.

[RFC 2459] Internet X.509 Public Key Infrastructure Certificate and CRL Profile.

[RFC 1421] Privacy Enhancement for Internet Electronic Mail: Part I: Message Encryption and Authentication Procedures.

[TLSKEYS] "Use of Shared Keys in the TLS Protocol", Gutman, P., October 2003, IETF, http://www.ietf.org/internet-drafts/draft-ietf-tls-sharedkeys-02.txt.