3GPP TSG SA WG3 Security — S3#34 **S3-040480**

July 6 - 9, 2004

Acapulco, Mexico

Source: Intel Corporation
Contacts: Selim Aissi, Sundeep Bajikar {selim.aissi, sundeep.bajikar}@intel.com
Contributing Companies: Intel, Toshiba
Title: Trust Requirements for Open Platforms in WLAN-WWAN Interworking

Document for: Discussion
Agenda Item: TBA

Abstract

While several decisions (e.g., selection of Alternative-2 for EAP-SIM UE Split Scenarios [1, 2, 3, 6]) have been made in 3GPP based on the open-platform nature of the PC, several 3GPP documents also call for the need to protect some specific functions on the terminal (.e.g., Local Link Trusted Terminal [4], [15]) Furthermore, in order to employ high-level protocols to perform secure transactions, it is necessary to ensure that attackers cannot hack Notebooks in WLAN-WWAN Interworking.

The trusted Open Platform (referred to as OP in this document) has built-in security mechanisms that place minimal reliance on the user or administrator to keep the OP and its peripheral devices secure. Trusted OPs are being developed that maximize the security of individual OPs through hardware and operating system-based mechanisms rather than through add-in programs and policies. To that end, security mechanisms are being built into chips, chipsets, and motherboards, among other system devices, because industry consensus is that hardware-based mechanisms are inherently more trustworthy than those created with software.

This document addresses the appropriate 3GPP trust requirements for an OP. It is based on a platform architecture comprised of hardware and software with security features that can be used as a basis for establishing trust in the entire OP. With the enhanced trust, a 3GPP infrastructure that includes OPs can support various mobile business applications and enable emerging data service businesses (e.g., DRM, Web Services).

1. Introduction

1.1. Background

In the 3GPP Wireless Local Area Network (WLAN) Interworking Security specification [2], it is stated that:

> *The security functionality required on the terminal side for WLAN-3G Interworking may be split over several physical devices that communicate over local interfaces. If this is the case, then the following requirements shall be satisfied:*

- *Any local interface shall be protected against eavesdropping, attacks on security-relevant information. This protection may be provided by physical or cryptographic means.*
- ***The endpoints of a local interface should be authenticated and authorised.*** *The authorisation may be implicit in the security set-up.*
- *The involved devices* ***shall be protected against eavesdropping, undetected modification attacks on security-relevant information. This protection may be provided by physical or cryptographic means.***

Therefore, it is clear that, for successful deployment of WLAN-3GPP Interworking, it is necessary that, in addition to the ME, the OP must have protection mechanisms against eavesdropping, and malicious modification of user data and operator applications residing on the OP. Furthermore, the OP must have secure authentication and authorization mechanisms.

Also, in the same specification [2], several types of attacks on a victim OP are described:

> ***Open platform terminals may be infected by viruses, Trojan horses or other malicious software. The software operates without the knowledge of the user on his terminal, and can be used for different types of attacks****:*
>
> - *If the user has credentials stored on a smart card connected to his terminal,* ***a Trojan residing in the terminal can make fake requests to the smart card and send challenge-response results to another MS****. For example, the owner of the latter MS could then get access with the stolen credentials.*
>
> *NOTE: This attack is performed inside the terminal, and it is independent of the external link between the terminal and the smart card reader, which can be secured or assumed to be physically secure.*
>
> - ***Trojans may perform all the usual activities: monitor the user's keyboard or sensitive data, and forward the information to another machine****.*
>
> - *Malicious software can be used to perform Distributed DoS (DDoS) attacks. That is, several instantiations of the software (residing on different hosts) synchronise and start a DoS attack simultaneously against a target.*
>
> - *Malicious software could be trying to connect to different WLANs, just to annoy the user.*

Also, the same specification [2] calls for several types of protection that must be implemented on all involved devices, including an OP:

> *The involved devices shall be protected against eavesdropping, undetected Modification attacks on security relevant information. This protection may be provided by* ***physical or cryptographic*** *means.*

Hence, it is clear that, securing the storage as well as input/output of sensitive data on an OP is of critical importance. Also, the above statements indicate that is it is necessary to isolate all applications that are managing (U)SIMs and (U)SIM readers, EAP-SIM and EAP-AKA protocols, and SAP applications from Trojans that can attack such applications and spoof sensitive credentials.

In the case of MBMS applications, there is a need to secure the streaming data as well as the storage of data key. Although it was agreed [5] that those requirements need further study, it is clear that the OP must have an acceptable support for a secure key management infrastructure (e.g., generation, storage, input/output).

In the case of 3GPP UE function split for a 3GPP WLAN user equipment, SA3 has identified 3 [3, 6] 3 alternatives for splitting the EAP-SIM application. SA3 has agreed to select Alternative 2, which calls for *all* functions of the EAP peer executed on the UMTS UE (with the GSM/UMTS cryptographic algorithms on the (U)SIM).

That decision was based on the threats resulting from an open-platform Notebook. However, even in the case of Alternative 2, the SAP applications residing on the OP need to be protected from malicious software attacks.

It is also worth mentioning that SA3 made Alternative 3 as optional [3]:

> *The functional split is as follows: the SIM/USIM performs the GSM/UMTS cryptographic algorithms, the GSM/UMTS UE derives the EAP-SIM or EAP-AKA master key MK from the GSM or UMTS session keys obtained from the SIM or USIM. All the other functions of an EAP peer are performed on the TE.*

In this case, the *other* functions of an EAP peer that are performed on the OP must also be protected with some physical or cryptographic means.

1.2. Trust Requirements for Open Platforms

While MEs are perceived to be trusted to perform today's mobile transactions (e.g., voice communication, SMS, micro-payments), the OP is conceived as subject to many known software and hardware attacks and a growing number of malware (e.g., viruses, Trojan horses, spyware).

Today's OP environment is built on *flexible, extensible, and feature-rich platforms* that enable consumers to take advantage of a wide variety of devices, applications, and services. Unfortunately, the evolution of shared networks and the Internet has made OPs more susceptible to attacks at the hardware, software, and operating system levels.

However, increasing the existing security measures, such as adding more firewalls and creating password protection schemes, can slow data delivery and frustrate users. Also, using only software-based security measures to protect existing OPs is starting to reach the point of diminishing returns. In general, many of the security benefits of a trusted OP could be achieved in some form simply by rewriting software, but this may be impractical or backward-incompatible.

Therefore, there is need for is a protected operating environment (combination of hardware and operating system features) that can provide a solid foundation on which privacy- and security-sensitive applications can run. This environment can help protect applications from malicious programs running in the main operating system.

Based on the threats described in section 1.1 above, this document proposes the following protection mechanisms [7, 8, and 9] for the OP (described in more detail in section 1.3 below):

- **Trusted Execution Environment** - Running without being altered, copied, spied upon, or interfered with, by other programs or the user himself.

- **Trusted Storage** - Storing data in encrypted form such that no other program can decrypt it. The process of decryption also needs to happen in trusted execution environment.

- **Trusted Input and Output** - By encrypting input and output, the OP creates a secure path from the keyboard and mouse to trusted applications and from those applications to a region of the OP's screen. These secure paths ensure that valuable information remains private and unaltered.

- **Trusted Identity Management** - Trusted identity is related to protecting the OP from revealing any information about user or platform identities.

By meeting those trust requirements, it is hoped to achieve the following characteristics for the OP in 3GPP environment:

- **Trusted** – the OP acts in a recognized manner and in able to communicate what that manner is supposed to be.

- **Reliable** – the OP is readily available for transactions and communications, as well as prepared to act against viruses and other intrusions.

- **Safe** – the OP is able to stop unwanted intervention or observation.

- **Protected** – the OP is shares information with only those that need to know within commonly accepted parameters for computer privacy.

Figure 1 below describes a possible configuration of a Trusted OP.

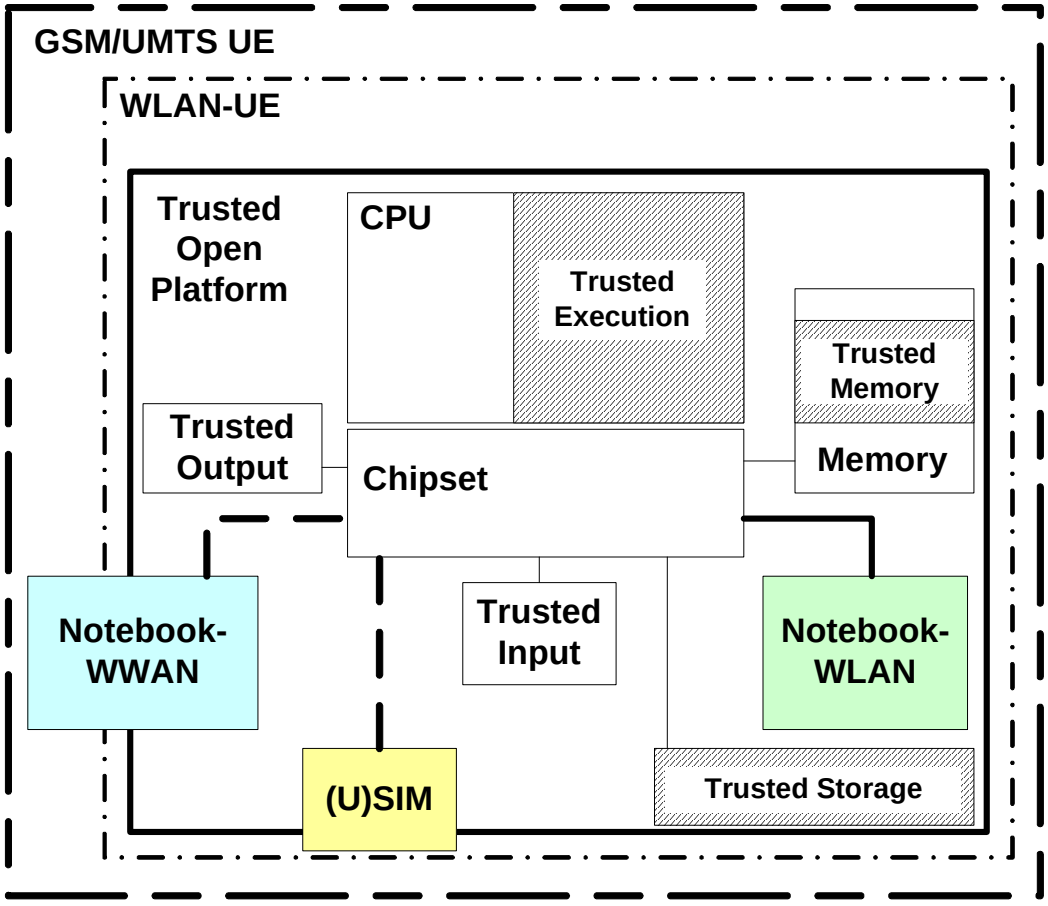

Figure 1: Trusted Open Platform Configuration Overview

Note: In Figure 1, the (U)SIM box is not intended to suggest changes to the agreed-upon 3GPP WLAN-WWAN Interworking architecture (e.g., ME as UICC holding device), but it is rather intended to indicate that, in a Trusted Open Platform, the local link from an application running on the OP to a (U)SIM can be trusted from end-to-end when using a Trusted Tunnel [4].

By achieving those characteristics, a trusted OP *can protect the platform* (software and hardware) from logical, or software-based, attacks.

While the OP can still be subverted by physical means, this mode of attack exposes only the secrets of the subsystem on the local platform, and not on other connected platforms. In other words, if a trusted OP were to be attacked by a virus, it could first of all notify the user that its software had been affected (not to be confused with anti-virus software that identifies and eliminates the virus, which is used as additional protection). Then the OP could notify all other computers on the network about the problem, so that no other computer would access the infected OP and spread the virus.

Furthermore, a trusted OP should not be only limited to protection from attacks, but it also:

- Provides protected storage of cryptographic and sensitive data
- Authenticates the OP by verifying its identity to other computing devices
- Supplies owner-defined metrics for reliable, secure network environment access of only other trusted computing devices

2. Protection Mechanisms

A trusted OP requires that a separate subsystem in that OP be trusted [7]. This subsystem is designed to provide reliable mechanisms for measuring and reporting integrity metrics, which ensure that the OP is trusted. This consists of two building blocks: Integrity Metrics and Cryptographic Hashing.

To ensure system integrity for the trusted OP, “integrity metrics” are used. These are defined as measurements of key platform characteristics that can be used to establish platform identity (such as boot-loader, hardware configuration, OS loader, and the OS security policy).

Cryptographic hashing is employed to extend trust from the BIOS to other areas of the platform, in the following simplified sequence:

The core elements of trust, built into the OP through trusted subsystem, first extend their trust to secure launch (e.g., boot-loader). Secure launch then extends its trust to the OS loader. The OS loader in turn extends its trust to the OS, which can then extend its trust to applications. This process ensures that the initial point of trust spreads the trust throughout the whole OP, thus resulting in a trusted OP.

The required protection mechanisms are described in more detail in the following subsections.

2.1. Trusted Execution Environment

Memory isolation [7, 8, 9, 16, and 17] refers to a strong, hardware-enforced memory-protection feature to prevent programs from being able to read or write one another's memory. Today, an intruder or malicious code can often read or alter sensitive data in a OP‘s memory. In a trusted OP, even the operating system should not have access to isolated memory, so an intruder who gains control of the very operating system would not be able to interfere with programs' secure memory.

The protected operating environment on the OP must isolate a secure area of memory that is used to process data with higher security requirements.

The protected operating environment provides a restricted and protected address space for applications and services that have higher security requirements. The primary feature of the protected operating environment is isolated or curtained memory, a secure area of memory within an otherwise open operating system.

Random access memory (RAM) in current computers is divided into two sections: the operating system, which is ring 0, and the user space, which is ring 3. Two addressing-mode bits control access to these sections. Ring 0 contains important system functions, including memory management, scheduling, and peripheral device drivers. User programs that run on the computer execute in ring 3. These user programs can also call into ring 0 whenever they require a system function, such as additional memory.

Using the current memory scheme, only virtual memory protection is achievable, and it is relatively easy for an attacker to add malicious programs to both the operating system and user space memory. Connecting to the Internet exacerbates transmission of these malicious programs.

By isolating off and hiding pages of main memory, trusted OPs can ensure that trusted applications running in the trusted execution space are not modified or observed by any other program or even the operating system itself. The RAM isolation effectively blocks any program that is running in the user space memory from accessing isolated memory and from even discovering that isolated memory exists. Isolated memory, which provides physical memory protection, is isolated from the open hardware and software environment on the OP, and therefore is protected from software attacks.

Each of the following trust mechanisms has a different security rationale, although the mechanisms can be used in conjunction with one another.

2.2. Trusted Storage

Trusted Storage [7, 8, 9, 16, 17] guarantees that the relevant system state—including the operating system and trusted helper programs—is identical at data storage and data access times. If the current OP state's hash does not match a copy of the expected system state sealed inside encrypted data, the tamper-resistant module managing the machine verification process prevents further decryption.

Trusted storage addresses a major OP security failing: the inability of an OP to securely store cryptographic keys, Session Keys, and passwords. Customarily, keys and passwords that protect private documents, protocol sessions (e.g., EAP-SIM) or accounts are stored locally on the OP's hard drive, alongside the documents themselves. This has been compared to leaving the combination to a safe in the same room with the safe itself. In practice, intruders who break into an OP can frequently copy decryption and signing keys from that OP's hard drive. Since the keys must be accessible to the OP users in order to be usable for their intended purpose, security engineers have faced a quandary: how can keys be stored so that they are accessible only to legitimate users and not to, say, a virus, which might acquire the same privileges as a legitimate user?

Trusted storage generates keys based in part on the identity of the software requesting to use them and in part on the identity of the OP on which that software is running. The result is that the keys themselves need not be stored on the hard drive but can be generated whenever they are needed -- provided that authorized software tries to generate them on an authorized machine. If an application other than the program that originally encrypted, private data should attempt to decrypt, that data, the attempt is guaranteed to fail.

Similarly, if the data is copied in encrypted form to a different machine, attempts to decrypt it will continue to be unsuccessful. Thus, your e-mail could be readable to your e-mail client, but incomprehensible to a virus.

An example of the threats eliminated by this trust mechanism is the SirCam e-mail worm. Whenever it infected an OP, it sent files it found there as e-mail attachments to randomly chosen Internet users. While existing access control and encryption systems address this attack, they might be bypassed or subverted. If someone compromises an OP, or it becomes infected with a worm or virus (such as SirCam), local software could be altered or private documents could be e-mailed or copied to other computers.

Files can be encrypted using a password, but if the password is short, someone who can copy the encrypted file will still be able to decrypt it (by trying each possibility in a brute force attack).

What's more, if the encryption software used, or the editor in which it is composed, is surreptitiously replaced with a modified version, it might leak the decrypted file's text (or password) to a third party.

Trusted storage can work together with memory isolation and secure I/O (section 1.3.1 above) to ensure that files can only be read on your OP, and only by the particular software with which you created it. Even if a virus or worm like SirCam leaks your encrypted file, the recipient will not be able to decrypt it. If an intruder or a virus surreptitiously alters your encryption software, it will no longer be able to decrypt the diary, so the contents of your file will remain protected.

2.3. Trusted Input and Output

Trusted input and output [7, 8, 9], or trusted I/O, aims to address the threats posed by key-loggers and screen-grabbers as well as software used by snoops and intruders to spy on computer users' activities. OP users can be tricked into loading malware and rogue programs. To prevent this trusted I/O can prevent some isolated areas from loading this type of software.

A key-logger records what you type, and a screen-grabber records what's displayed on the screen of an OP.

Trusted I/O provides a secure path from the keyboard to an application -- and from the application back to the screen. No other software running on the same OP will be able to determine what the user typed, or how the application responded.

At the same time, trusted I/O will provide protection against some more esoteric attacks. It will allow applications to determine whether their input is provided by a physically present user, as distinct from another application impersonating a user. And it will defeat some cases of forgery where one program attempts to corrupt or mask another's output in order to deceive the user.

2.4. Trusted Identity Management

Trusted identity management [7, 8, and 9] is related to protecting the OP from revealing information about any user or platform identities.

A set of policies may be defined for subscriber identity information, and then it is expected that those identity policies are enforced. Therefore, identity management is a very important aspect of user privacy.

Identity Management builds atop the trust requirements outlined above to create a comprehensive solution for protecting user data. Pseudonym can be used during communication to hide the user's true network identity. Mobile user's identity management requirement is to generate (or request to generate), maintain, delete (or request to delete), and apply for identities in accordance with the mobile user's security policy.

3. Conclusions

To meet several security requirements for 3GPP-WLAN Interworking (section 1.1), there is a need to have more trust in Open Platforms. This document addresses the appropriate trust characteristics for an Open Platform to meet those requirements.

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

4. Terminology

Notebook – A personal computer that can easily be carried by hand.
ME – Mobile Equipment [14].
Notebook-WLAN – Set of devices that support WLAN capabilities on a Notebook.
Notebook-WWAN – Set of devices that support WWAN capabilities on a Notebook.

5. References

[1] 3GPP TR 22.934, 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Feasibility study on 3GPP system to Wireless Local Area Network (WLAN) Interworking.

[2] 3GPP TS 33.234, Technical Specification 3rd Generation Partnership Project;
Technical Specification Group Service and System Aspects; 3G Security; Wireless Local Area Network (WLAN) Interworking Security; (Release 6).

[3] 3GPP S3-040197, Further Liaison on Termination of EAP authentication over Bluetooth for 3GPP UE function split.

[4] 3GPP S3-040272, Use of a Trusted Tunnel to Secure Local Terminal Interfaces.

[5] S3-040037, BMSC handing of the previous keys.

[6] S3-030780, SIM Access Profile in split WLAN-UE.

[7] Andrew Huang. The Trusted PC: Skin-Deep Security, Computer, October 2002.

[8] B. Balacheff, D. Chan, L. Chen, S. Pearson, and G. Proudler. Securing smartcard intelligent adjuncts using trusted computing platform technology. In *the Proceedings of IEIF Fourth Smart Card Research and Advanced Application Conference (CARDIS 2000)*, pp 177-195, Bristol, UK, 20-22 September 2000.

[9] Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone, Handbook of Applied Cryptography, CRC Press, ISBN: 0-8493-8523-7, Fifth Printing, August 2001

[10] Recommendations for the Protection against Distributed Denial-of-Service Attacks in the Internet, Bundesamt für die Sicherheit in der Informationstechnik, 2000.
http://www.iwar.org.uk/comsec/resources/dos/ddos_en.htm

[11] Security Techniques Advisory Group (STAG): Glossary of security terminology, ETSI Technical Report 232, November 1995

[12] Shirey R. Request for Comments 2828, Internet Security Glossary. May 2000

[13] Schiller J. Request for Comments 3365, Strong Security Requirements for Internet Engineering Task Force Standard Protocols. August 2002.

[14] TR-3GA-21.905, Vocabulary for 3GPP Specifications.

[15] Eric Gauthier. A man-in-the-middle attack using Bluetooth in a WLAN Interworking Environment. December, 2003.

[16] G. Proudler. What's in a Trusted Computing Platform? http://www.informit.com

[17] S. Pearson. Trusted Computing Platforms, the Next Security Solution. http://www.informit.com