Trusted Mobile Platform

Hardware Architecture Description

10/27/2004

Trusted Mobile Platform
NTT DoCoMo, IBM, Intel Corporation

File Name: TMP_HWAD_rev1_00_20040405.doc

Trusted Mobile Platform

Hardware Architecture Description

Rev. 1.00

June 23, 2004

Copy Right Notice

Status

This is a stable revision of the Trusted Mobile Platform Hardware Architecture Description that was agreed upon by Trusted Mobile Platform Promoters.

Contents

1. INTRODUCTION

The Trusted Mobile Platform initiative defines a comprehensive end-to-end security architecture for mobile wireless devices. It consists of the hardware architecture, the software architecture, and the protocol specifications. The Hardware Architecture Description (HWAD) and the Software Architecture Description (SWAD) define a generic architecture for Trusted Mobile Devices (TMD). This document describes the hardware components that are necessary to create TMDs. Some of these software components also bridge the security features with the protocol defined in the protocol specifications.

The architecture defined herein is based upon the prior work performed in the Trusted Computing Platform Alliance (now the Trusted Computing Group) but extends and modifies the basic concepts for low power, handheld devices with limitations on available MIPS and memory. In addition, the concept of trust levels is introduced. Manufacturers and carriers can make their own decisions based upon guidance found in the Trusted Mobile Platform documents as to what trust level is appropriate for the services they choose to deploy. This document does not define implementation specifics. A variety of technology options could be used to build a TMD that supports the architecture defined here.

This document is organized as follows. Section 2 lists related documents and standards. Section 3 provides a list of acronyms and their definitions. Section 4 describes the overall architecture for the trusted platform and defines the components that impact system trustworthiness. Sections 5 through 10 describe in more detail the individual components that make up the trust system. Section 5 describes the platform root of trust. Section 6 describes the Trusted Platform Module (TPM), the functionality required and implementation options. Section 7 defines the security features required for a secure processor that can enforce hardware domain separation between trusted and untrusted applications. Section 8 describes the requirements to protect the data paths between the CPU and TMD input and output devices. Section 9 describes physical and environmental protection features that should be applied to a

TMD. Section 10 describes peripheral devices that have a potential impact on the security of the TMD.

2. Related Documents

[1] FIPS PUB 140-2 Security Requirements for Cryptographic Modules
[2] FIPS PUB 180-1 Secure Hash Standard
[3] FIPS PUB 197 – Advanced Encryption Standard
[4] FIPS PUB 46-3 Data Encryption Standard
[5] RFC 1321 The MD5 Message Digest Algorithm
[6] PKCS #11 Cryptographic Token Interface Standard
[7] RFC 2119
[8] Trusted Computing Group (TCG) Design Philosophies and Concepts Version 1.0
[9] Trusted Computing Group (TCG) Main Specification Version 1.1b, http://www.trustedcomputinggroup.org/, February 2002 (also known as Trusted Computing Platform Alliance (TCPA) Main Specification Version 1.1b)
[10] TCG Software Stack (TSS) Specification Version 1.0
[11] Trusted Mobile Platform Security Requirements
[12] Trusted Mobile Platform Protocol Specification Document
[13] 3GPP TS 31.101: " UICC-Terminal Interface; Physical and Logical Characteristics".
[14] 3GPP TS 31.102: "Characteristics of the USIM Application".
[15] 3GPP TS 33.102: " 3G Security; Security architecture".
[16] 3GPP TS 11.11: "Specification of the Subscriber Identity Module - Mobile Equipment Interface
[17] 3GPP TS 51.011: "Specification of the Subscriber Identity Module - Mobile Equipment Interface".
[18] TS 03.20: “Digital cellular telecommunications system (Phase 2+); Security related network functions”.
[19] TS 23.048: “Security mechanisms for the (U)SIM application toolkit”.
[20] “BioAPI Specification Version 1.1”, March 16, 2001, The BioAPI Consortium, http://www.bioapi.org.
[21] S. Prabhkar, S. Pankanti, A. Jain, “Biometric Recognition: Security and Privacy Concerns”, IEEE Security and Privacy, March/April 2003, pp33-42.

[22] *"Content Protection for Recordable Media Specification: SD Memory Book Common Part", Rev 0.96, Nov. 26, 2001, by Intel, IBM, Matsushita, and Toshiba*

[23] Trusted Mobile Platform Software Architecture Description

3. Definitions and Abbreviations

For the purpose of this document, the following definitions apply:

2G	2nd Generation
3G	3rd Generation
AES	Advanced Encryption Standard
AKA	Authentication and Key Agreement
AuC	Authentication Center
AUTN	Authentication Token
BD	Biometric Devices
BIS	Boot Integrity Services
BIU	Bus Interface Unit
CK	Cipher Key (in 3G)
COTS OS	Commercial off-the-shelf Operating System
CRTM	Core Root of Trust for Measurement
DES	Data Encryption Standard
DIR	Data Integrity Registers
DMA	Direct Memory Access
GSM	Global System for Mobile Communications
HLR	Home Location Register
ICC	Integrated Circuit Card
IK	Integrity Key
IPC	Inter Process Communications
K	USIM Individual Key (in 3G)
Kc	Ciphering Key (in 2G)
Ki	Individual Subscriber Authentication Key (in 2G)
LCD	Liquid Crystal Display
LPAR	Logical Partition
MCM	Multi-chip Module
ME	Mobile Equipment

MMU	Memory Management Unit
MS	Mobile Station
MTC	Monotonic Counter
NV	Non-volatile
OLED	Organic Light Emitting Diode
PCR	Platform Configuration Register
PID	Process Identity
PIN	Personal Identification Number
Platform ROT	Platform Root of Trust
RAND	RANDom number (used for authentication)
RES	User RESponse bit signed RESponse that is the output of the function f2 in a 3G AKA
ROT	Root of Trust
RSA	Rivest, Shamir, Adleman
RTM	Root of Trust for Measurement
RTR	Root of Trust for Reporting
RTS	Root of Trust for storage
SHA	Secure Hash Algorithm
SGSN	Serving GPRS Support Node
SIM	Subscriber Identity Module
SN	Serving Network
SRES	Signed RESponse. Authentication value returned by the SIM or by the USIM in 2G AKA
TBC	Trusted Boot Code
TBH	Trusted Boot Hardware
TCG	Trusted Computing Group
TCPA	Trusted Computing Platform Alliance
TMP	Trusted Mobile Platform
TP	Trusted Platform
TPA	Trusted Platform Agent
TPM	Trusted Platform Module
TSS	TPM Support Software
UICC	UMTS IC Card

USIM	Universal Subscriber Identity Module
USB	Universal Serial Bus
VLR	Visitor Location Register
XRES	Expected RESponse. Authentication value delivered by the 3G HLR/AuC

4. Architecture Overview

The hardware architecture of a TMD generally consists of 1) a CPU that provides platform control, runs the Operating System (OS), runs a variety of software applications, manages the user interface, interfaces to memory, interfaces to external devices, and supports the communication channel with the appropriate protocols and security. One architectural option is based on a single CPU that performs all of these functions. More complex architectures use two CPUs, with the second CPU generally dedicated to support the communications channel and over-the-air interface. 2) System NV storage; this is generally flash technology, 3) system RAM, 4) memory controllers for the flash and RAM memories, 5) DMA controller, 6) interrupt controller, 7) clock circuits, including a real time clock, 8) keypad and display controllers 9) communications interfaces including over-the-air interfaces like Bluetooth or ISO 14443 and 10) miscellaneous logic.

In addition to these components, GSM phones also have an interface to a SIM. The SIM provides security services for user identity and for channel security. The SIM also can be used for secure storage.

To improve the security of the TMD, a Trusted Platform Module (TPM) or the functional equivalent of the TPM as defined by [9] must be added along with hardware needed to execute a secure boot operation. These components ensure that the Trusted Mobile Device executes a trusted boot process at power up so that the TMD always boots into a trusted environment.

4.1. One Processor Architecture

Figure 4-1 shows a generic one processor architecture for a handheld device. Components residing inside the red dotted line indicate components that might reasonably be integrated into a single device. The Core Root of Trust for Measurement technology could be an integrated ROM, part of the flash memory, or a discrete ROM. Similarly, the TPM could be embodied as an integrated component or a discrete TPM.

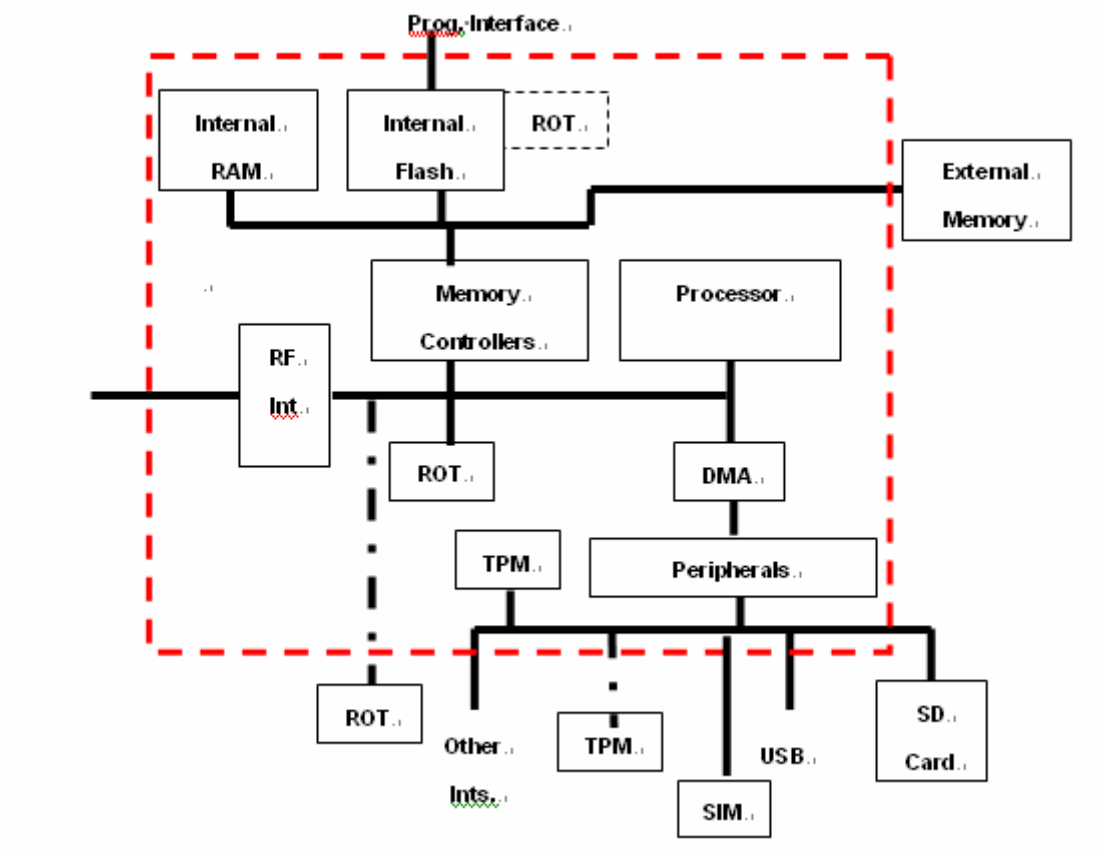

Figure **4-1 One Processor Architecture**

4.2. Two Processor Architecture

Figure 4-2 shows a generic two processor architecture for a TMD. The implementation options regarding the CRTM and TPM are identical to the options for the one processor architecture. The advantage derived from adding a second processor is the potential to cleanly separate applications from communications processing. This model

significantly reduces the risk of compromise to the communications processor.

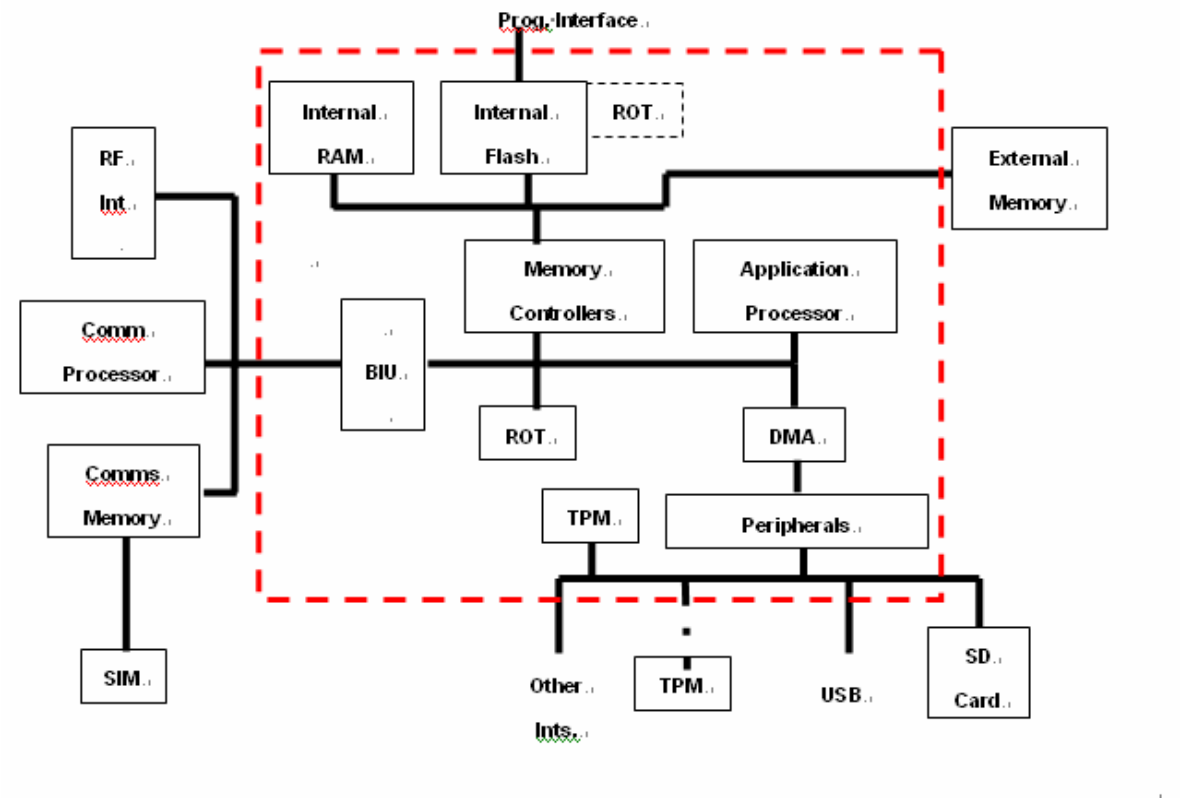

Figure 4-2 Two Processor Architecture

4.3. Definition of the Security Boundary

The security boundary consists of those components that impact the trust/security operations performed by the TMD. The components that comprise the security architecture for the two processor system are highlighted in red in Figure 4-3 below.

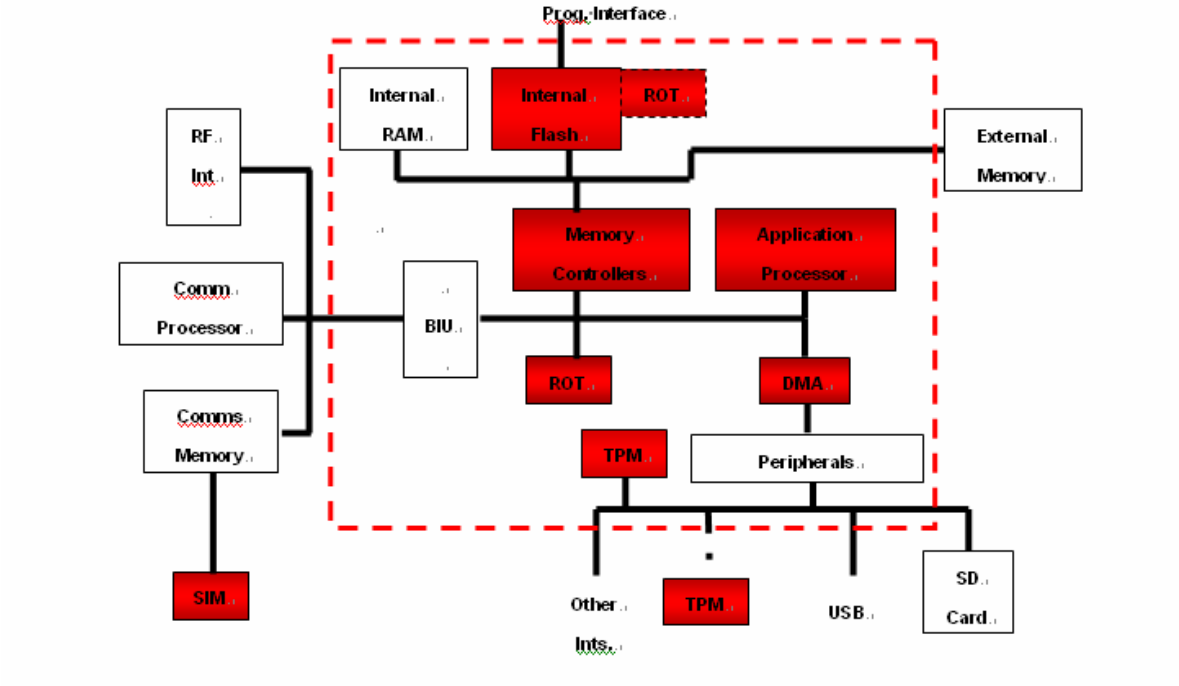

Figure 4-3 Security Boundary

The components that make up the trust boundary include 1) the application processor 2) the TPM 3) the SIM 4) The CRTM (could be a discrete device or part of another device) 5) NV memory (internal flash), 6) the memory controllers 7) DMA controller. The rationale for including each of these inside the hardware trust boundary is contained in the following sections.

4.3.1. Application Processor

In this architecture, the Application Processor plays a role in many security and trust operations. At cold start, trusted boot is executed under control of the application processor. After boot, the application processor executes trusted and untrusted applications and in a trusted system, isolates and protects the data associated with trusted applications. The application processor also interfaces with security peripherals like the TPM, SIM, or a biometric input device.

The main security mechanism that the application processor must provide is protection

of trusted applications and their data from untrusted applications. The current generation of CPUs used in handheld applications does not provide this feature, and many existing handheld OS's are not designed to take advantage of domain isolation hardware if it did exist. In addition, there may be other implementation options that may be used to provide domain isolation without modifications to the CPU.

4.3.2. TPM

The Trusted Computing Group (TCG) in [9] has defined the functionality and interfaces for a trust device call the Trusted Platform Module (TPM). The TPM provides a protected execution environment where secrets can be processed without being exposed, secure and hidden storage, and cryptographic engines needed to measure the integrity and validate the source of software residing on the platform. The TPM works closely with the CRTM to perform trusted boot. The TPM cryptographic engines are used to measure and validate the hardware and/or software configurations of the platform. The TPM can compare these measurements to stored "known good" measurements that are securely loaded onto the device and can use these to validate the platform configuration as trusted or untrusted. These values can also be reported to a remote verifier.

Platform measurements are performed each time a platform is activated from a cold start. At cold start, the Platform's Core Root of Trust for Measurement initiates a trusted boot sequence and uses the TPM to perform the cryptographic measurements based on the Secure Hash Algorithm (SHA-1) and RSA algorithm that prove the platform's integrity and authenticate the source of the platform code. Platform measurements can also be performed at times other than power up and can also measure only a part of the platform HW and SW configuration. For example, the integrity and source of a specific application can be checked each time the application is launched.

The cryptographic capabilities of the TPM can be used for more than just trusted boot. TPM internal storage can be used for protected storage of cryptographic keys that are used by the platform. The TPM cryptographic assets can be used to encrypt sensitive data before it is stored in system memory. The internal TPM described in section 6 also provides a bulk encryption capability that can be used for processes like encryption

of private data before it is stored in system memory or real time encryption of streaming data.

The TMD requires much of the functionality of the [9] defined TPM. However, mobile platforms have limitations with respect to available power, physical area available, and bill of material costs that are more stringent than for desktop platforms. The design of current discrete TPM devices has not fully taken these constraints into account. The Trusted Mobile Platform project will investigate alternatives to discrete TPMs that provide the same essential functionality, but with simplifications that may result in physical and electrical characteristics more compatible with the constraints of handheld devices.

4.3.3. Platform Core Root of Trust for Measurement

The Platform Core Root of Trust for Measurement is functionally ROM storage for the trusted boot code (CRTM). The CRTM performs the initial trust measurements for the platform. On power up, the CRTM is invoked to perform the trust measurements on the remainder of the platform. To accomplish this, the CRTM must be stored with a memory technology that does not allow for the code to be modified. In addition, the platform architecture must ensure that the platform CRTM will be invoked each time there is a power up sequence. If the CRTM can be bypassed, then there is not assurance that the platform has been measured and the platform cannot be trusted.

The Platform CRTM does not need to actually be implemented as ROM technology. Other memory technology (i.e. flash memory) could be implemented so that it is functionally equivalent to a ROM. In the case of flash memory, this means that at least that part of the flash used to store the CRTM must be designed so that once programmed, it cannot be overwritten with new code.

4.3.4. System Memory and Memory Controllers

Non-volatile system memory is used to store the OS, applications, drivers, and other essential SW components. NV memory is also used to store data. Data that is not private can be stored in memory without it being encrypted. Private data must be encrypted with the TPM or a software application prior to being stored.

Encrypting private data before its stored protects the information from being read, but does not protect that data from being modified or moved. The TMD must provide hardware mechanisms that provide domain isolation and protect stored sensitive data from being modified in memory. This requirement may be satisfied in a number of ways, including possible modification to the memory architecture so that data associated with one process is protected from being modified by other processes.

4.3.5. DMA Controller

In Security Class 3 level systems, the DMA must be controlled by the trusted kernel. The DMA cannot allow trusted physical memory to be accessed by untrusted applications or by an untrusted OS.

For Security Class 1 and Class 2 level systems (refer to Section 4.4) there are no additional security requirements against the DMA Controller.

4.3.6. SIM

In a GSM phone, the SIM's primary purpose is to authenticate the user to the platform. This authentication approach does not uniquely identify an individual, but does prove that the person physically owning the SIM also knows the password – two of the elements generally required to authenticate a user. The TMD can use the authentication data to establish access control based on the user's identity.

4.4. Levels of Security

Different usage models merit different trust solutions based on the value of the service, the threat level, how liability is distributed for theft of the service or item, and the cost that someone will bear to provide a trusted solution. Varying levels of trust can be achieved by choosing different implementation options. The trust levels can be associated with different usage models that require different levels of trust. The trust levels defined for Trusted Mobile Devices are Security Class 1, Security Class 2 and Security Class 3. The major technology components associated with each class can be summarized as follows:

- Security Class 1 devices
  - No hardware security features
  - Minimal Software integrity checks
  - Closed system architecture – limited ability to add new software
  - Usage model – Very limited secure transactions and payment protocols, cannot validate the state of the platform at boot

- Security Class 2 devices
  - HW Trusted Platform Module implementing a [9] subset. TPM functionality focused on trusted boot and measurement of the platform SW configuration.
  - Trusted boot from ROM or equivalent
  - Common Criteria level 2 or equivalent
  - Usage model – Secure transactions and protocols, OTA SW download, can validate the state of the platform at boot

- Security Class 3 devices
  - Hardware and software domain separation
  - TPM must be able to make platform measurements available to protocols supporting remote attestations and the ability to exchange platform measurements with remote parties.
  - Common criteria level 3 or equivalent
  - Trusted Computing Base
  - Open system
  - Usage model – Legally binding digital signatures, systems that require run time protection plus trusted boot, very high value transaction, enterprise remote access.

Security Feature	Security Level		
	Security Class 1	Security Class 2	Security Class 3
TPM	-SW TPM or equivalent	- HW TPM ([9] subset)	-Integrated TPM or MCM
CPU architecture	-No requirement	- MMU	-HW domain separation, trusted

			DMA
Hardware tamper resistance	-Interlocking enclosure	- Tamper evident enclosure	-Sensored enclosure -Trusted I/O
Integrity and attestation	-Minimal integrity checks	- Integrity checks and source authentication (Trusted boot)	-Trusted boot -Runtime integrity checks
Domain separation enforcement	-COTS OS separation (user account + process) -Java (JAAS or OSGi)	- Hardened OS - Encrypted memory system	-Secure processor architecture -SW domain separation
Access control	- Discretionary	- Mandatory + discretionary	-Mandatory + discretionary
Software certification	- No certification	- CAPP EAL 2 or equivalent (No formal certification.)	-CAPP EAL 3 or equivalent
User authentication	-PIN	-Passphrase	-Hardware crypto -Biometrics
Root of Trust	-None	-ROM or equivalent	-ROM or equivalent
SW Architecture	-No TCB	-TCB	-TCB
Secure storage	-No	- Through encryption	-Encryption & domain separation
Random number generation	-SW PRNG (Pseudo Random Number Generator)	-HW based PRNG	- True Random Number Generator
Environmental protection	None	- Interfaces protected	-Interfaces protected -Voltage and temp. sensors

Table 4-1 Trust Levels

5. Platform Root of Trust

This section describes the hardware components that comprise the Platform Root of Trust. This section defines the roles and responsibilities of the PlatformROT in establishing an initial trusted state for the TMD. It identifies secure ROM storage options for the Core Root of Trust for Measurement (CRTM) which works in conjunction with the hardware Trusted Platform Module (TPM) to perform platform integrity measurements.

The techniques described in this section to measure and report the integrity of a TMD are based on concepts first developed by the [9].

5.1. Requirements

Security Class 3 requirements for the Platform Root of Trust are as follows:

5.1. The CRTM must be the first code executed after power is applied to the TMD.

5.2. It must not be possible to modify the contents and policies contained in the Core Root of Trust for Measurement.

5.1 Trusted Boot

The sole responsibility of the CRTM is to perform initial trust measurements for the host platform. The CRTM must be invoked upon boot-time and will be constituted as a "trusted" process that is executed by the Platform ROT. The CRTM performs boot time operations that gather trust measurements within the designated parameters set by the Platform ROT policies.

5.2 Core Root of Trust for Measurement

Platform trust measurements require the existence of trusted components on the platform that perform this function. The [9] defines the Platform Root of Trust as a hardware/firmware component with predefined policies specified in the memory device for Core Root of Trust for Measurement (CRTM). The CRTM is registered within the memory blocks, and must not allow for the policies to be modified. In order for the mobile device's processor to perform these functions for integrity checking on the mobile device, it must meet certain criteria specified within the CRTM. Meeting a criteria level for a trusted environment can be a lengthy and difficult process impacting the architecture design phase for the TMD. In order to achieve a trusted architecture, the architecture must reduce the number of trusted components to the minimum number required to establish a trust statement.

These polices in the CRTM are comprised of several metrics and require the presence of several components necessary to establish a trust statement of authenticity. These are the Root of Trust for Measurement (RTM), Root of Trust for Reporting (RTR) and or Root of Trust for Storage (RTS). In addition, the platform architecture must ensure that the platform ROT will be invoked each time there is a power cycle sequence. If the ROT can be bypassed, then there is not an assurance that the platform has been measured and the platform cannot be trusted.

5.3 CRTM Lifecycle

What follows is an example of a CRTM lifecycle for a mobile device when it is powered on for the first time. In order for the mobile device to enter its trusted state the CRTM is initiated starting the trusted boot sequence. The CRTM WILL conduct several levels of integrity measurements. It performs a measurement process to gather system level metrics and also conducts an integrity test for the intended mobile device owner, as mentioned in the Trusted Mobile Platform Software Architecture document in Section 11 (*User Authentication*).

The Platform Root of Trust confirms the functionality and the trustworthiness of the operating environment and enables the TPM to exchange “trusted” data to be processed within the mobile environment’s solid state flash memory and or other means of storage. The mobile device specific information and environment is derived from conducting Root of Trust for Measurement (RTM), Root of Trust for Reporting (RTR) and or Root of Trust for Storage (RTS). Given the normal usage model for mobile devices and their associated “persistent ON state”, the mobile device manufacturer has the option to register the trusted mobile device prior to user provisioning.

5.4 Interfacing with the Platform Root of Trust

The Root of Trust on the platform is embodied in the TPM. The CRTM has hardware and software interfaces to the ROT. At boot time, the TPM interfaces with the CRTM

which WOULD be registered in the designated memory boot blocks contained within the TMD's solid state storage. Please reference section 5.5 (Memory Technology for the Root of Trust) for more information in how the TPM interfaces with flash, third party memory structures and/or ROM technologies.

5.5 Memory Technology for the Root of Trust

Memory technology used to store the CRTM must be designed so that once programmed, it cannot be overwritten with new code. The concept of designating memory blocks within the internal memory for the CRTM can be referenced in (*Software Architecture Specification – Encrypted File Systems*). Alternative means of memory technology can utilize USIM and or other peripheral devices as the storage medium.

5.6 CRTM Operating Environment

The CRTM must be placed in a non-rewritable, protected high security domain within the memory architecture. Permission to rewrite and or alter the CRTM is impossible even with super-user / root privileges. The CRTM allows the authenticity and integrity checking of the boot time parameters, by checking the hardware and software integrity within the trusted mobile device. The CRTM would then interface with the Trusted Boot Hardware.

5.7 Trusted Boot Hardware

The trusted boot hardware (TBH) is solely responsible for executing the mobile platform's RTM. The TBH is associated with measurement agents and the TPM. Trusted boot is normally conducted at the boot/power cycle of the mobile device and checks each functional process (PID), determining whether processes can be 'trusted' as they come online. The measurement process takes advantage of the transitive trust model described in greater detail in section 5.1 of the Software Architecture Description (SWAD). All boot time activity is recorded by the TPM in the PCRs and the measurement logs. The [9] refers to this process as an authenticated boot. If the

process has been altered and has been not properly registered at the time of the trusted boot, the TMD can be in an arbitrary state and that state will be recorded and reported. As stated in the [9], "Process that use secrets are indirectly protected by TP hardware. Secrets can be sent to a platform after the platform's software state has been measured and reported. Stored secrets can be released after the platform's software state has been measured and checked. Reporting, storage and retrieval is carried out by the TPM. A platform's software state includes all software processes. So, if a process relies on the use of secrets, it cannot operate unless it and its software environment are correct." [9]. According to [9], in order for a mobile device to be in a 'trusted state' it needs the following:

- At least one root of trust for measuring integrity metrics.
- Exactly one root of trust for storing and reporting integrity metrics
- At least one Trusted Platform Measurement Store
- At least one item of Validation Data
- Exactly one Trusted Platform Agent

At the stage of each boot time, the RTM or measurement agent computes an overall digest of the table of expected digests and compares the overall digest with the value in the data integrity registers (DIR). If a difference exists at the time of boot than that of what is expected based on the data supplied by a Validation Entity (VE), then the TMD platform cannot be trusted.

6. Trust Module - TPM

The TPM as defined by [9] uses a set of cryptographic primitive operations to measure the hardware and software of a platform. In the TMD, the TPM is a trusted security peripheral that is employed by the CRTM during the trusted boot process. The minimal set of capabilities required for a TPM are the RSA and SHA-1 algorithms, a random number source, a monotonic counter, secure internal storage, and a secure processing environment.

A functional block diagram of a minimal capability TPM is provided in Figure 6-1 below.

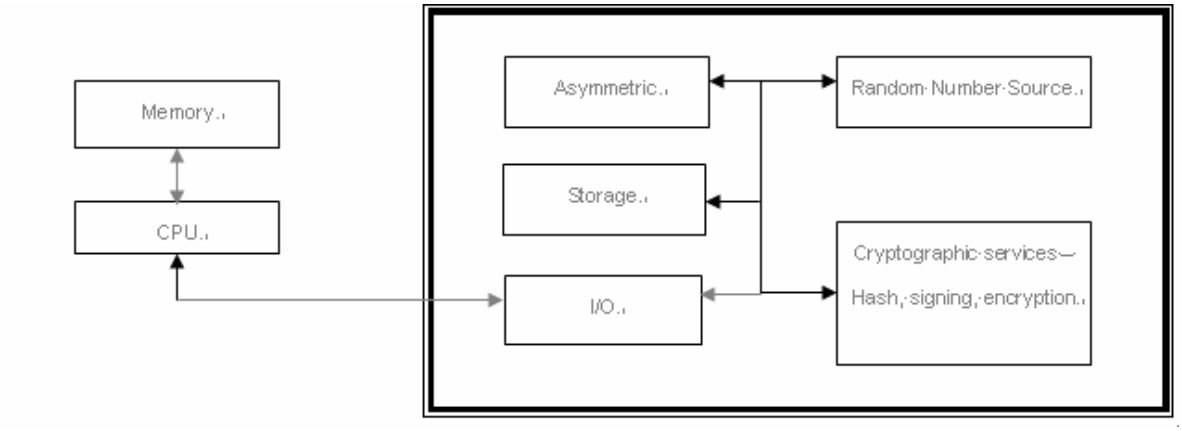

Figure 6-1 TPM Functional Block Diagram

[9] does not specify the implementation of the TPM. Possible embodiments include but are not limited to a software based TPM, a discrete device, or a TPM embedded as part of another device. This specification does link the TPM embodiment to the specific level of TPM security, but a TPM is required in both Security Class 2 and Class 3 security devices. A software-based TPM may be used in Security Class 1 trusted platforms. A hardware based TPM, either discrete or embedded, is required for either Security Class 2 or Class 3 TMDs. Of the two hardware options, the embedded TPM has advantages in cost, performance, electrical interfaces, and physical security.

6.1. Requirements

The following requirements apply to the TPM used in a TMD.

6.1. The TPM provides secure and hidden storage for TPM secrets
6.2. The TPM provides a secure and hidden environment to process secrets
6.3. The TPM includes a random number source and the ability to generate keys
6.4. The TPM shall include the RSA algorithm for digital signature creation and verification
6.5. The TPM shall include the SHA-1 algorithm for hashing
6.6. The TPM shall include SHA-1 based HMAC for command authentication
6.7. The TPM shall be capable of performing atomic security operations. This

property means that security operations run to completion and that the intermediate results cannot be modified or exposed

6.8. The TPM shall include a monotonic counter

6.9. The TPM shall provide capabilities for secure key storage and secure data storage. This can be achieved either by storage of unencrypted data inside the TPM boundary or by storing encrypted data or keys in system memory.

For a Trusted Mobile Device, it is also desirable that the TPM have additional characteristics including but not limited to:

- Extensions that allow the minimal set of TPM cryptographic engines (SHA-1, RSA and RNG) to be used to support security protocols
- Additional crypto primitives that allow the TPM to further support standard security protocols. These may include symmetric cryptographic engines), other hashing algorithms like MD5, and support for additional asymmetric algorithms.
- Reduced size and power dissipation. To achieve this, architectures that allow the TPM to be integrated with other devices is permitted.

6.2. Cryptographic Engines

The [9] defines a minimal set of cryptographic engines/algorithms that must be supported by the TPM. These include the RSA algorithm and the Secure Hash Algorithm (SHA-1). The TPM used in a Trusted Mobile Device must support these algorithms.

In addition to the minimal set of cryptographic engines, it is desirable that the TPM on the Trusted Mobile Device support additional cryptographic algorithms used by common protocols including but not limited to SSL/TLS, IPSec, Internet Key Exchange (IKE), and content protection among others. Among the most likely algorithms for inclusion are the symmetrical algorithms Advanced Encryption Standard (AES) and the Data Encryption Standard (DES), the hashing algorithm MD5 (Message Digest 5), SHA variants SHA-256, SHA-384 and SHA-512, the Diffie-Hellman (DH) key exchange algorithm and Elliptic Curve Cryptography (ECC).

6.3. Platform Configuration Registers

The TPM internal architecture includes 160 bit Platform Configuration Registers (PCRs) that are used to store information about the current state of the platform. [9] specified that TPMs have sixteen PCRs. However, the embedded TPMs for the TMD are only required to have eight 160 bit PCRs.

The PCRs must be volatile registers. This is due to the fact that the platform measurements must be re-started each time the platform is powered up. The TPM must be able to store individual integrity metrics as well as sequences of integrity measurements. Sequences of integrity measurements are stored using the TPM Extend operation. In this operation, the current measurement is concatenated to the present value stored in the PCR being used to store the measurement of the operation. The resulting concatenated value is hashed and the result is stored back to the PCR initially used for storage. This operation assures that at any point in time, the value stored in a PCR is dependent upon both the specific operation being measured and the sequence of measurements preceding that measurement. The TPM QUOTE function can be used to request that information stored in a PCR be output from the TPM.

PCRs must be protected from manipulation of their value by anything other than the TPM itself. For that reason, the PCRs should not be part of the system storage, particularly is TMDs that do not have domain separation hardware and software, but should be part of dedicated TPM storage.

6.4. Key Storage

Two types of key storage are required in the TPM. One type of required storage is non-volatile storage used for long term storage of keys or other data. This storage must allow keys to be retained even when power is removed from the device. The second type of key storage is volatile key storage. This is essentially working storage for keys that are currently being used by a TPM.

Non-volatile key storage can be either internal to the TPM, part of the system memory, or could be in both. There are important differences in how keys are stored in the two locations. If a TPM has dedicated internal NV storage, then keys can be stored in their without being encrypted prior to being stored. In this instance, physical protection of the key storage memory by the TPM prevents the keys from being revealed outside the TPM boundary. Alternatively, keys that are stored in system memory do not have the intrinsic physical protection afforded to keys stored in internal TPM memory. As a result, keys that are stored in NV system memory cannot be stored without first being encrypted. TPMs used in TMDs may use either symmetric or asymmetric algorithms to protect keys and other sensitive data. TPMs that do not have symmetric crypto engines must use RSA to encrypt the keys before they are stored in NV system memory. Those TPMs that have symmetrical crypto engines may optionally use the symmetric crypto engine for key encryption. The use of the symmetrical crypto engine results in better throughput and reduced power dissipation. The recommended minimal key size to support these functions is 128 bits for symmetric algorithms and 2048 bits if an asymmetric algorithm is used. In either case, the keys used to encrypt the keys/data being stored in NV system memory must themselves be protected.

Trusted Mobile Device TPMs must have dedicated internal storage for keys that are currently being used by the TPM. The types of keys can include RSA private keys, symmetric keys and HMAC keys. The internal key storage must be dedicated key storage available only to the TPM. Since these keys are unencrypted while being used, they must not be accessible by the rest of the system.

6.5. Trust Module Processor

The Trusted Mobile Device TPM must have its own dedicated Trust Module Processor subsystem including the TPM processor, dedicated program memory, dedicated RAM and scratchpad registers. The Trust Module Processor controls all of the internal operations of the TPM. It performs functions such as managing the TPM interface to the system, controlling operation of the TPM crypto engines, processing TPM commands received from the system, managing keys inside the TPM, and performing security checks on the TPM system. The Trust Module Processor ensures that TPM

operations are atomic.

Based on the required functions of the Trust Module Processor, the processor can be relatively primitive. It is essentially a microcontroller that moves data from one crypto engine to another or to/from a crypto engine and memory. Implementation options range from a simple custom microcontroller up to and including full featured CPUs. Cost and power issues favor the use of simple microcontrollers.

A key security requirement satisfied by the Trust Module Processor is that the TPM has a hidden execution environment. Cryptographic processes are executed fully in the TPM using TPM internal memory and the Trust Module Processor. The specific operations, the intermediate results, the cryptographic keys and other sensitive information cannot be observed from outside the device. This requirement applies to both the discrete TPM and for the embedded TPM.

6.6. Monotonic Counter

The Trusted Mobile Device TPM must include a monotonic counter (MTC) that can be used by protocols to reduce the threat from replay attacks. There are a variety of implementation options for the MTC including pure software, hardware MTCs based on flash memory technology, and hardware based on counter logic. In any case, the state of the MTC must be non-volatile, must not be able to be manipulated by entities outside the TPM, and must be guaranteed to never repeat values. Note that the MTC is not required to increment in units of “1”.

6.7. Atomic Operations

The Trusted Mobile Device TPM must be capable of performing atomic operations. This means that a given cryptographic operation is guaranteed to run to completion and that the intermediate results of a primitive operation cannot be modified.

Simple TPM operations may only involve one engine and performing a single operation and are inherently atomic. An example of this is to calculate the hash for a block of

data. In this example, the data is input to the hashing engine; the engine operates on the data and returns the resulting hash. This requirement becomes more complex when multiple engines are involved.

Creating a digital signature is an example of a more complex atomic operation. In this case, atomicity requires that the data being signed in first hashed and then the hash must be provided as an input parameter to the signing engine. In this example, the hash of the data must be contained inside the TPM until it is signed. If the intermediate result, in this case the hash, is exported from the TPM and then subsequently imported for the sign operation, the TSS, TPM driver or other code has an opportunity to substitute a different hash value. If that happens, there is no assurance that the hash being signed corresponds to the hash originally calculated for the message being signed.

6.8. Cryptographic Algorithms

The following cryptographic algorithms are required in the TMD TPM.

- RSA for encryption and signing (PKCS#1 v1.5)
- SHA-1 (FIPS 180-1)

The following cryptographic algorithms are optional in the TMD TPM.

- RSA (PKCS#1 v2.1)
- DH key agreement
- ECDH key agreement
- DSA signature
- ECDSA signature
- AES (FIPS 197)
- DES and Triple DES (FIPS 46-3)
- MD5 (RFC 1321)
- SHA-256/384/512

Other cryptographic algorithms may also be provided.

Refer to the Trusted Mobile Platform Software Architecture Description Section 9.2 for additional information on cryptographic algorithms.

6.9. TPM Deviations for Trusted Mobile Devices

Sections 6.1 through 6.7 define areas where the TPM used in the TMD is essentially the same as that defined by the [9]. However, due to the unique constraints in terms of boot time, system performance, power dissipation, operating environment, etc., deviations from the [9] are allowed. Deviations permitted for the TMD include:

1. Only 8 PCRs are required
2. External key storage and data storage can be based on the use of a symmetric crypto algorithm. If this option is chosen, the AES algorithm in CBC mode with a minimum 128 bit key is recommended.
3. The cryptographic algorithms implemented in hardware inside the TMD TPM may be used as accelerators to support security protocols or platform security operations. When used in this mode, the [9] defined authentication protocols (OIAP and OSAP) are not required.
4. In addition to the minimal set of algorithms required for [9] interoperability, the TMD TPM may include additional symmetrical, asymmetrical or hashing algorithms including but not limited to DES, TDES, ECC, and MD5.
5. Implementation of the full set of TPM commands is not required. The minimal required subset includes:
    - TPM_OIAP (Object Independent Authorization Protocol)
    - TPM_OSAP (Object Specific Authorization Protocol)
    - TSS_Bind
    - TPM_Unbind
    - TPM_CreateWrapKey
    - TSS_WrapKey
    - TSS_WrapKeyToPCR
    - TPM_LoadKey
    - TPM_EvictKey
    - TPM_GetPubKey
    - TPM_PCRRead

- TPM_Extend
- TPM_Quote

7. Processor Security Features

It is assumed that all processors of interest will have conventional general-purpose CISC or RISC architectures. Special purpose machines (e.g. Java machines or CLR machines) are not considerations. It is possible to implement a JVM on conventional processors, and it is also possible to extend conventional processors with the ability to interpret Java byte-codes, but these are performance considerations; not security considerations.

7.1. Introduction

The fundamental role of a processor in a secure system is to provide isolation. The purpose of isolation is to enable tamper-resistant software execution so that software can process secrets without interference or observation, and can be assured that secrets cannot leak outside its isolation boundary. How strong the isolation provided is, and how much impact on system software architecture can be tolerated depends on how important the security requirements are.

As indicated earlier in this document, three levels of security are defined: Security Classes 1, 2 and 3.. This section discusses the implications to the processor architecture for these three levels.

7.2. Security Class 1

Any processor architecture in the market today can support a Class 1 system; there are no specific requirements on the processor architecture.

7.3. Security Class 2

Any processor that supports supervisor privilege and virtual addressing can be used to build a Class 2 Security system. This would tend to exclude most Real-time embedded processors, but would include all existing general-purpose processors.

With supervisor mode and virtual addressing, a privileged kernel can isolate any application or service execution in a virtual address space. The actual degree of security (i.e., the strength of the isolation) depends more on a set of complex design trade-offs in the operating system (including its degree of openness). Isolation using virtual address spaces comes at a price in both performance and programming model changes depending on how strong the isolation is. Typically, operating system designs will trade-off complete isolation (thereby decreasing security) in favor of performance. Sometimes this trade-off can be as extreme as executing all code in the same virtual address space. Strong isolation would require that third-party device drivers do not execute in supervisor privilege, and that the only form of interaction between isolated components be of the form of secure Inter-Process Communication (IPC) through the privileged kernel.

Experience has shown that this type of operating system architecture, while meeting the requirements for Class 3 security, suffers from unacceptable performance characteristics, and presents a programming model that is incompatible with the programming models that the majority of existing applications are designed to.

Only if security is an overriding consideration would such an architecture be acceptable. No operating system in the market today has been willing to suffer these performance implications (and model changes) for the sake of security.

7.4. Security Class 3

Security Class 3, therefore, not only requires the re-structuring of operating systems, but also requires processor extensions to:

- Securely boot into trusted kernel

- Harden isolation boundaries
- Support legacy programs
- Mitigate performance degradation (optional)

The fundamental goal is to protect the system from any form of software attack.

7.4.1. Requirements

A Class 3 Security Processor must provide the facilities that enable the following requirements to be satisfied:

7.1. A Trusted Code Base (Trusted Kernel) must be able to control access to all hardware features (particularly physical memory and devices). In particular,

a. The Trusted Kernel must be able to restrict what physical memory is available to the legacy operating system.

b. The Trusted Kernel must be able to control what physical memory is accessible through DMA.

7.2. The Trusted Mobile Device must securely boot into the Trusted Kernel.

7.3. The Trusted Kernel must execute in its own isolated execution environment with access to all processor resources. This execution environment must allow the Trusted Kernel to execute in virtual addressing mode.

7.4. The Trusted Kernel must be able to support the execution of a legacy applications and services.

7.5. The Trusted Kernel must be able to provide multiple isolated execution environments in which trusted applications and services execute.

7.6. The trusted environments must be isolated from each other, and from the legacy execution environments.

7.7. The Trusted Kernel must be able to provide a high-performance secure inter-process communication mechanism between trusted and untrusted execution environments and between trusted environments. Thus, an untrusted application must be able to call a trusted service, and a trusted application must be able to call an untrusted service.

7.8. On reset, volatile memory used by trusted applications and services must be scrubbed of secrets.

7.5. Possible Approaches to Class 3 Security Processors

The following are two general approaches to Class 3-security processors. Each approach described may be implemented in multiple different ways depending on the base processor architecture that is being extended.

7.5.1. Physical partitioning

In this approach, sufficient hardware processor resources are added to allow the physical memory of the platform to be partitioned into multiple separate Domains. The partitioning would be managed by the trusted kernel that executes in its own Domain. Other Domains would be used to execute operating system services, while the remaining domains would be used to execute applications and network data services (with varying degrees of trust). All the requirements (as listed above) should be satisfied.

7.5.2. Virtual Partitioning

In this approach, the processor is extended with a traditional general-purpose virtualization architecture. This architecture gives a Virtual Machine Monitor fine-grained control over what physical resources a particular Virtual Machine can access; any violation would result in a trap to the Monitor. In addition to the virtualization extensions, the processor must ensure that the platform securely boots into the Virtual Machine Monitor. The Virtual Machine Monitor creates virtual machines. The Virtual Machine Monitor creates virtual machines. The Virtual Machine Monitor would act as the Trusted Kernel; it would create a separate virtual machine for the execution of each application or network data service.

7.6. CPU Peripherals

In addition to the fundamental modifications required to the CPU, any of the architectures described above in Section 7.5 may also require corresponding modifications to the DMA Controller, memory controllers, and Interrupt Controller.

The implementation of these peripherals must ensure that they cannot be exploited by untrusted operating system services or untrusted applications to gain access to trusted memory space.

8. Trusted I/O

The devices that provide the user interface in a Trusted Mobile Device must be trusted. The interactions between the user and the device must be protected or hidden from interception, observation, and hacking. This chapter describes the hardware architecture of Trusted I/O provided by the Trusted Mobile Device.

The following sets of trusted input/output requirements apply to the Trusted Mobile Device.

8.1. The Trusted Mobile Device shall be able to display trusted messages on the display

8.2. The Trusted Mobile Device shall provide an indicator to the operator that is activated only when the displayed message is trusted.

8.3. The Trusted Mobile Device shall provide security mechanisms that ensure that trusted input (i.e. from a keyboard) is received by the TMD without any modification, deletion, or addition to the input data stream. Security mechanisms to be employed may include hashing or encryption of the message or data being input

8.4. Independence of I/O device. The wiring signal of the I/O devices used by the Trusted Mobile Device should be separated from other parts of the device.

The I/O devices used in Trusted Mobile Devices are listed in Table 8-1. In the handheld device, the standard input method is finger input device and LCD devices are widely used as the general display unit. Here, the Trusted Mode Indicator (TMI) is a special device used only by the Trusted Mobile Device.

Table 8-1 Trusted I/O Devices

	Category	Used device	Security consideration

Input	Finger input	Key pad, touch panel	Used for PIN, password input
	Pointing device	Cursor, pen	
	Sound	Microphone	Voice recognition
	Image	Camera	Face recognition
	Biometrics	Fingerprint scanner	Authentication
Output	Bitmap display (General Display Unit)	LCD, OLED	Common display device. Must provide a Trusted Display Mode (works with TMI).
	Indicator	LED	Can be used as Trusted Mode Indicator.
	Sound	Speaker	Possible auxiliary TMI

8.1. Integrated Device / External Device

Figure 8-1 shows the block diagram of a typical Trusted Mobile Device. A TMD has both integrated input and output devices. For example the keypad is just buttons and is wired to its controller, which in many cases is embedded in the CPU chip. A malicious attacker may open the device and insert some kind of eavesdropping device. This is very difficult and requires advanced techniques. If the device has a tamper resistant or sealed packaging implementation, these kinds of attacks are made more difficult. The embedded processor used by a TMD may provide a built-in I/O interface, e.g., an LCD controller, touch panel controller, and keypad interface. Alternately, a discrete I/O interface chip on the peripheral bus may be used. The internal signal between the interface and the device should be routed directly between the source and destination and should not be accessible from the surface of the board to prevent malicious access. Independence of the I/O devices is also required for a Security Class 3 device.

An external I/O device can be attached to the Trusted Mobile Device. If we allow it to process sensitive data or information, a reliable cryptographic mechanism is required to

protect the data passing through the interface. Furthermore, an external I/O device must itself be a tamper resistant/evident device and have the same security level as the TMD or otherwise, the external I/O device must not handle sensitive information. Mutual authentication between such a trusted external I/O device and the Trusted Mobile Device must be performed. If encryption is used to protect the interface to external devices, the encryption device inside the TMD must also be secure, and it should be part of either the TPM or the Trusted Computing Base.

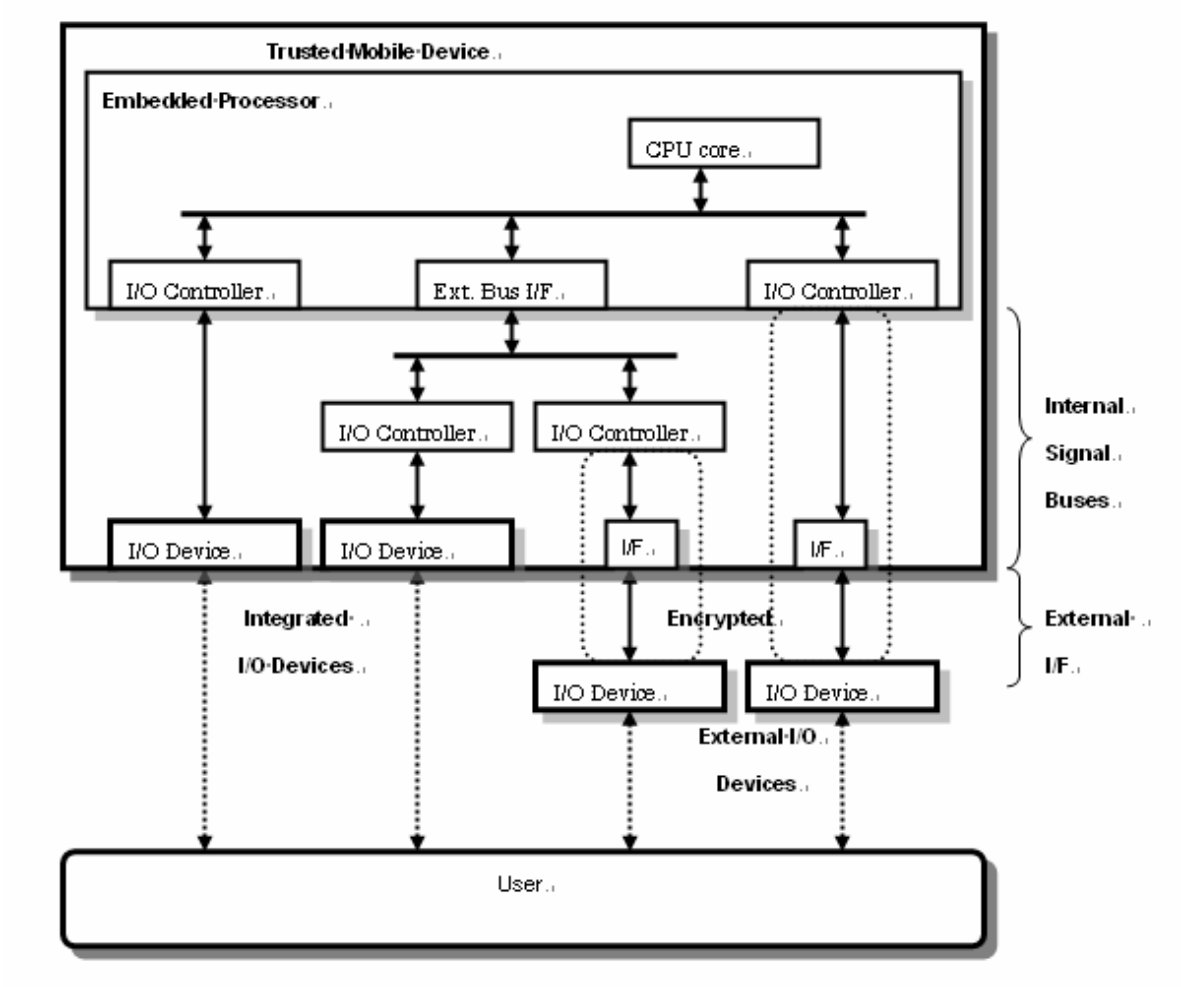

Figure 8-1 Implementation example of the Trusted User Interface Device

8.2. Input Devices

Keypads or touch panels are widely used as finger input devices. These devices are also used for password entry. When a device is used to handle such kinds of sensitive data, any input data entered by a user must be protected from malicious attacks. The left side of Figure 8-1 roughly shows the block diagram of an integrated input device. Physical tampering attacks against the internal signal buses between devices and controllers must be prevented. To prevent this kind of attack, tamperproof seals or tamper-resistant packaging must be used for the device. Security Class 3 Trusted Mobile Devices must provide this protection.

8.2.1. Biometrics

This is an optional user authentication mechanism based on biometric information about legitimate users. These features add authentication flexibility and reduce the exposure of a PIN or pass phase in public areas. They also allow for individuals to be identified to a platform as opposed to a device being identified to a platform. This allows the Access Control policy to be tied to a specific user, even if a device is shared. The security requirements for the biometric devices are same as for other I/O devices. Microphones or cameras can also be used as biometric devices.

8.3. Output Devices

The Trusted Mobile Device can have several output devices listed in Table 8-2.

8.3.1. General Display Unit

A flat-panel device such as an LCD (Liquid Crystal Display) or an OLED (Organic Light-Emitting Diode) is widely used as a graphical user interface device. The device has a frame buffer memory to store the screen image and is controlled by the TCB. The DMA controller periodically copies the data from the frame buffer to the LCD controller. Thus, the steps of this operation should be trusted. Memory separation and Trusted DMA are used to achieve this requirement.

8.3.2. Trust Mode Indicator

A Trusted Mobile Device must support a Trusted Mode Indicator (TMI). There are several implementation choices to achieve this requirement. See also the Trusted User Interface Chapter in the Trusted Mobile Platform Software Architecture Document. This indicator must be controlled by the TCB.

8.4. Summary of Trusted I/O and Security Levels

Table 8-2 Trusted I/O for each Security Level

	Security Level		
	Security Class 1	**Security Class 2**	**Security Class 3**
Trusted I/O	No	Yes	Yes
Tamper-resistant or sealed package for Integrated I/O devices	None	Sealed packaging	Tamper resistant packaging
Trusted Mode Indicator	No	Yes	Yes
Secured I/F for External I/O device	No	Yes, if device is used for authentication, etc.	Yes, if device is used for authentication, etc.

8.5. SD Card

This subsection discusses the SD card as an example for secure external storage.

8.5.1. SD card overview

Secure Digital Memory Card, otherwise known as the SD card is a non-volatile memory with digital rights management capabilities, (www.sdcard.org). The SD card enables secure downloading of all types of digital data, (e.g., music, movies, photos, news, e-books, etc). Aside from offering a solution as an external memory device, the SD card

complies with the Content Protection for Recordable Media (CPRM) specification as determined by 4C Entity (http://www.4centity.com) to ensure digital rights management.

The CPRM architecture within the SD Card consists of Media Key Blocks (MKB). The MKB is composed of a matrix of encrypted versions of a media key, which is encrypted by a number of Device Keys. Several Device Keys are securely stored in a device.

8.5.2. Secure Key Storage

If a Device Key is compromised, the Media Key is compromised. Hence, a TMD shall securely store the Device Keys.

During the processing, several keys are generated. A TMD shall protect the confidentiality and integrity of such keys.

8.5.3. Integrity protection of processing unit

A TMD shall protect the integrity of the SD card processing modules resident in the device.

8.5.4. Content Encryption – Recording device

Three scenarios are possible how content can be provided to a TMD. Content being encrypted compliant to the SD card, channel encrypted (e.g., SSL/TLS), and plain text content.

(a) If the content is encrypted compliant to the SD card, the content shall be sent to the SD card as it is.

(b) If the content is sent through an encrypted channel, the content must be decrypted in the device first, into plain text content before it is encrypted into the SD card. Therefore, the TMD shall protect the confidentiality and integrity of the original

content after it arrived at the device.

(c) If the content is provided in a plain text to the device, a TMD should not transfer it into the SD card encryption system because there is no assurance if the content is legal.

8.5.5. Content Encryption – Playback device

The encrypted content is sent to the playback device. The device decrypts it and renders (or play) it.

(a) If the rendering device is a hardware component such as an LSI chip which is capable of taking encrypted content directly as an input, and, decrypts it and renders it all inside of the hardware device, there is no chance that the original content is stolen.

(b) If software modules are used to decrypt the encrypted content and/or to render it, there is a chance for an attacker to steal the content unless the integrity of the processing modules and the confidentiality of the data being processed by the modules are protected.

Therefore, a TMD shall protect the integrity of the playback module and the confidentiality of the plain text content data being transferred.

9. Physical and Environmental Protection

Physical protection for the TMD includes three types of protections: 1) protection of the interfaces so that attackers cannot gain access to sensitive information on the device including keys and private data, 2) protection against environmental attacks against the platform. These are primarily based on voltage and temperature attacks and 3) protecting the device from being tampered with. This is focused on detecting instances where the handheld device is physically attacked at the package level.

The hardware described in this section must satisfy the following requirements to be considered a Security Class 3 TMD:

9.1. Trusted Mobile Devices shall provide physical evidence of tamper attacks (applicable to all three security levels)

9.2. Trusted Mobile Devices shall not perform trusted operations if the package (case) has been opened (applicable to Security Class 2 and Class 3 levels only)

9.3. Trusted Mobile Devices shall be able to detect temperature conditions outside the normal operating range of the device and shall not perform trusted operations if the temperature is outside the normal operating range (Security Class 3 TMDs only).

9.4. Trusted Mobile Devices shall be able to detect voltage conditions outside the normal operating range of the device and shall not perform trusted operations if the voltage is outside the normal operating range (Security Class 3 only).

9.5. External interfaces of security critical components shall be protected so that security critical information cannot be accessed by external probing of the TMD or the packages that make up the device (both Security Class 2 and Class 3 TMDs).

9.1. Packaging

The Trusted Mobile Device requires physical protection at the package level. The objective is to protect the TCB and internal security resources from physical tampering or replacement. The three platform trust levels require different protection mechanisms. Security Class 1 level platforms may use only an interlocking enclosure. Security Class 2 security platforms must add tamper evident tamper protection that can be used in conjunction with the interlock switch. At the Security Class 3 level, an interlocking sensor should be employed that can provide an indication to the TCB that the security perimeter has been breached.

For Security Class 1 devices, the interlocking enclosure is similar to the enclosures that are currently used in handheld devices. These enclosures do not have any special security features and the enclosure does not provide any evidence to the user that the device has been attacked. In addition, these enclosures do not have a tamper sensor so that the platform SW cannot be notified that the tamper boundary has been penetrated.

Security Class 2 devices add passive tamper evident protection at the device level. Several technologies can be used to provide tamper evidence including various stickers and protective coatings. The coatings are not intended to actively prevent penetration of the device. Similarly, tamper evident devices are not designed to provide notification to active circuitry that can respond to the tamper event. These devices merely provide evidence to a user that the TMD has been opened and potentially modified in a way that could defeat the trust mechanisms on the platform.

Security Class 3 level devices add an interlocking device with a sensor. The sensor is activated if the interlock is broken. The output from the sensor is returned as a non-maskable interrupt to the TCB. The interrupt routine must respond by erasing unprotected keys and by erasing and/or encrypting sensitive data before it can be compromised. There are a variety of non-destructive technologies that could be used such as mechanical switches, pressure sensitive switches, or even light sensitive switches.

The interlock sensor must be tested periodically to ensure that it is functioning properly. Although the switch itself cannot easily be tested, the circuit path between the switch and the interrupt controller can be tested. The CPU can issue a Test Switch Logic signal to the Switch Test Logic that forces the non-maskable interrupt from the interlock switch to be asserted. The test software validates the source of the interrupt and clears the tamper interrupt. A block diagram showing the interlock device and the test path is shown below in Figure 9-1.

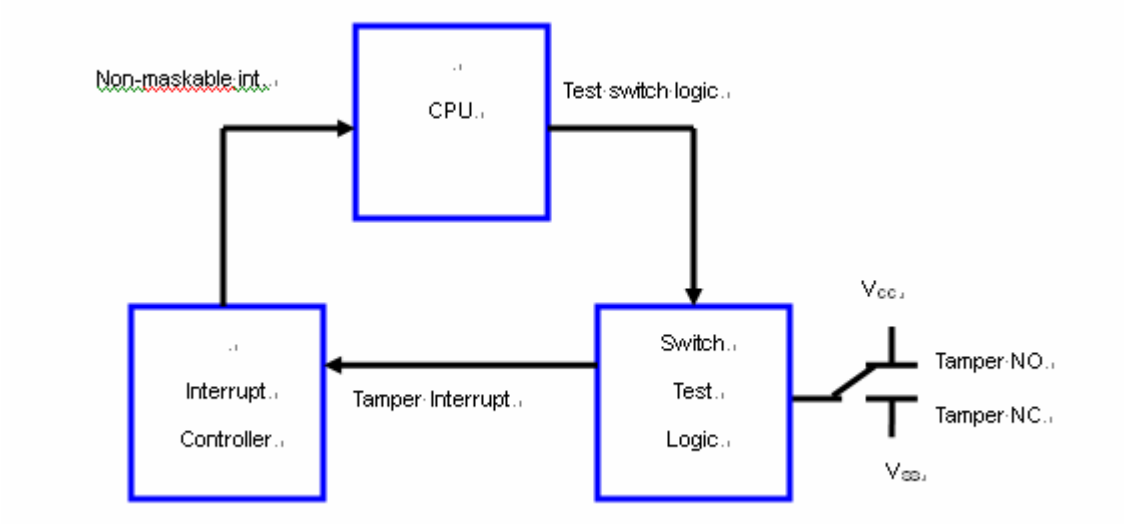

Figure 9-1 Tamper Protected Interlock Switch

9.2. External Interfaces

Figure 9-2 depicts the block diagram of a two processor architecture (presented earlier in section 4.2) with an interface to a discrete external TPM, an interface to a SIM card, and a number of additional interfaces. Each of these interfaces is an access point that could potentially lead to a compromise. These interfaces must be protected in two ways. First, the hardware design must prevent access to or monitoring of the internal operations of the device. Second, the TCB must be designed so that it does not export unprotected sensitive data out to the interfaces.

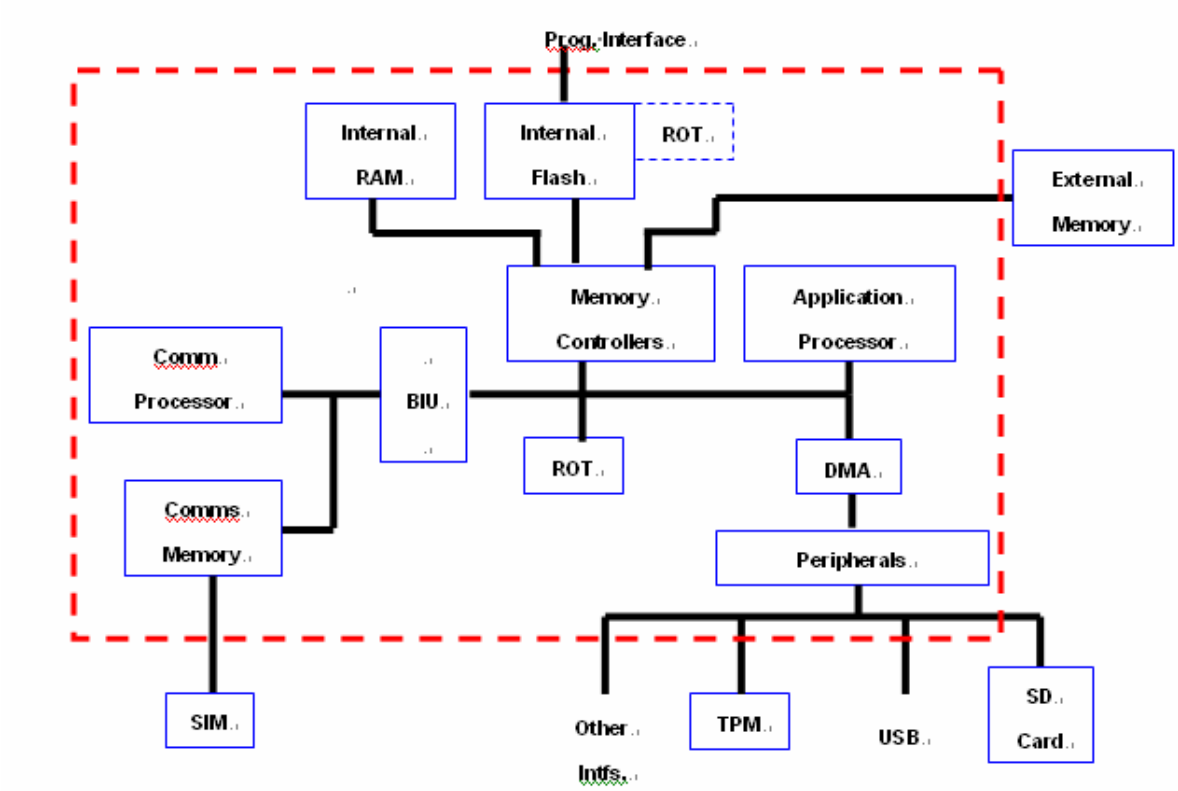

Figure 9-2 Two Processor Architecture with Interfaces

9.3. Voltage & Temperature Protection

The voltage and temperature protection circuits monitor the platform environment and protect the security and trust assets from being exploited through environmental attacks. These protection circuits monitor the current status of the temperature and voltage and generate an error message to the TCB if the current temperature or voltage is outside the normal operating range for the device. The TCB in turn generates a message for the user, informing the user of the problem.

Since the security resources cannot be trusted to operate correctly if experiencing an adverse operating environment, the TCB must ensure that the platform exits any trusted mode operations and initiates steps to protect the platform if an adverse environmental condition is detected. The TCB must terminate all trusted processes, make sure that any unprotected private data is encrypted and stored, erase unprotected keys, and block any further requests for trusted processes. The TCB cannot re-enable

trusted processes until the error condition is removed.

The environmental protection circuits must have a test/monitor function that allows the circuits to be tested periodically for correct operation. The essential steps are:

1. The TCB forces a failure condition exceeding the lower threshold to the sensor/detector
2. The TCB monitors that an environmental failure is generated
3. The TCB clears the environmental failure
4. The TCB forces a failure condition exceeding the higher threshold to the sensor/detector
5. The TCB monitors that an environmental failure is generated
6. The TCB clears the environmental failure

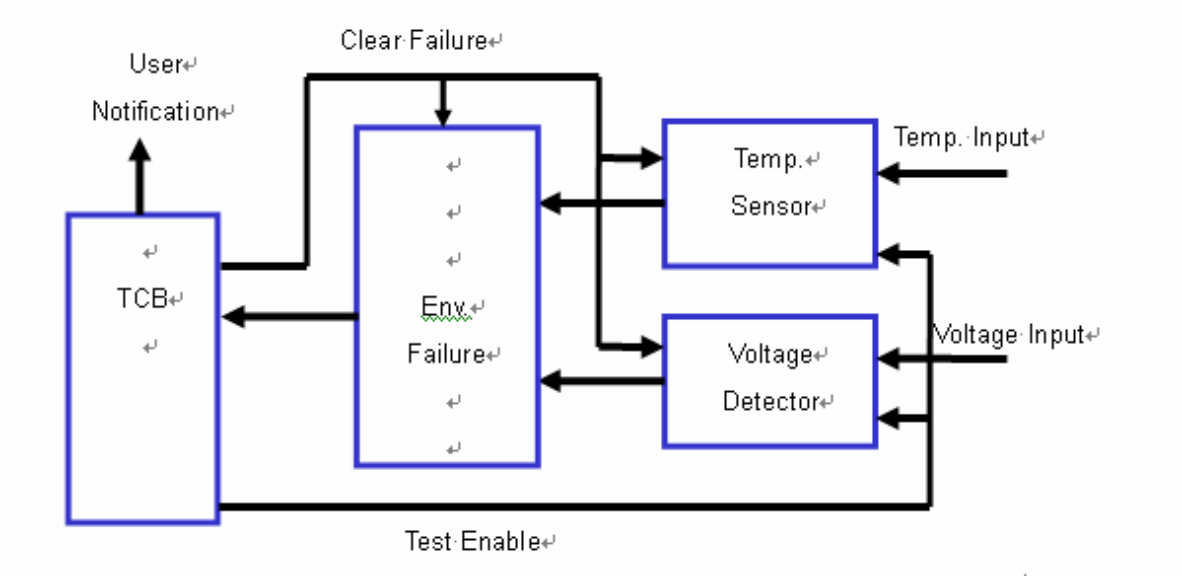

Figure 9-3 Environmental Test Circuits

10. Peripheral Devices

10.1. Overview

This chapter addresses requirements for peripheral devices in the context of the

Trusted Mobile Devices. There are several peripheral devices that are addressed in this document; specifically this document covers peripheral devices that are currently available – USIM/SIM, biometrics and external memory devices. Among all the potential peripheral devices, the requirements for the USIM or SIM are the most important ones for mobile devices. The SIM/USIM plays the key role in mutual authentication between the user and the network in addition to many other functions performed on the SIM/USIM.

Another means of user authentication is the utilization of biometrics which provides advanced capabilities in providing physical user authentication. Typical biometric system use physical features such as fingerprint, face characteristics, hand geometry, iris properties, voice, etc. for recognition.

External memory cards are used for storing data for digital cameras, digital music and movie distribution, etc. There are many types of memory cards available. This document addresses the security requirements for the SD card from Trusted Mobile Device perspective. The SD card is a non-volatile memory with digital rights management capabilities.

Lastly, this chapter addresses the requirements for general peripheral devices.

10.2. Requirements

The following requirements apply to TMD peripheral devices and their interfaces

10.1. Biometric matching data stored on the TMD must be protected from alteration, substitution, theft, or misuse by applications.

10.2. TMD applications that process content protected data must pass an integrity check.

10.3. Protected content processed on the TMD must not be stolen from the TMD.

10.4. Content to be protected by the Trusted Mobile Device combined with the SD card should be properly protected before it is sent to the storage device.

10.5. Confidentiality and integrity must be provided for critical/sensitive data that is transferred via the external interface with a peripheral device. A

mechanism must be provided to prevent replay attacks if needed.

10.6. External interfaces that allow a peripheral device to become a bus master must be prohibited

10.3. SIM/USIM

This section addresses USIM security features in the context of Trusted Mobile Device. The conventional SIM [16] [17], otherwise known as an ICC (Integrated Circuit Card), is a micro-controller based access module or smart card. The utilization of the SIM technology was first introduced and defined for 2G wireless systems. Originally it has been specified as one physical and logical entity. The SIM functionality never distinguished itself as a platform and or application. With its introduction in 3G, the SIM has advanced as a means to be utilized in a form of an application on the 3G UICC (UMTS IC Card), allowing capabilities to represent its logical characteristics. If the SIM application is active, the UICC is functionally identical to a 2G SIM. The SIM or SIM application on a UICC only accepts 2G commands.

The USIM [14] is not a physical entity. Unlike its SIM counterpart, it is purely a logical application that resides on a UICC. The UICC [13] is the physical and logical platform for the USIM. It contains at least one USIM application and may additionally contain a SIM application. Additionally, the UICC may contain additional USIMs and other applications. In order to protect user identity, confidentiality, user data and user information confidentiality, the AKA (Authentication and Key Agreement) procedure for the SIM or the USIM allows the SIM or the USIM to be authenticated by the serving network domain. The USIM AKA also enables the serving network to be authenticated by the USIM (mutual authentication). In addition to the authentication, 2G AKA procedure generates the key Kc as specified in [18]. The ciphering key, Kc, is used to cipher the layer 1 data flow, by a bit per bit or stream cipher [18].

The 3G AKA procedure generates the CK, the ciphering key, and IK, the integrity key, as specified in [15]. The CK and IK are used to protect confidentiality and integrity of data that is transferred over the radio channel. The reference [15] defines the security architecture for the 3G mobile telecommunication system. Each security feature that is

defined in [15] is considered in the context of Trusted Mobile Device, and security implications and requirements for each feature are addressed in this section.

The reference [15] defines the following security architecture in 3G implementations:

- Network access security
- User domain security
- Application security
- Security visibility and configurability

10.3.1. SIM Interface (Access / Readers)

There are various types of SIM readers available as an external attachment to personal computers as described below [intel]:

USB/PCMCIA SIM Access: This is currently the most common way to access a SIM card. In this usage model, a USB (universal serial bus) SIM reader and SIM reader driver software can be purchased or provided by an operator for use on a notebook computer. Since the USB SIM reader is a PC/SC standard compliant reader, it is accessible using the smart card subsystem. The USB SIM card reader is treated as a generic smart card reader. The smart card access is provided using a generic set of API. The application can use the GSM 11.11 specification and APIs for interacting with SIM cards. The USB readers are very easy to use and there are no additional regulatory requirements for using them to access SIM. However, the user has to physically remove the SIM from the cell phone and insert it into the USB reader. In general, physical access to a SIM in a cell phone is not easy. Another issue with USB readers is that the data path to and from the SIM is over an open bus and is thus susceptible to SIM attacks.

SIM Access from GPRS PC Cards: PCMCIA (Personal Computer Memory Card International Association) cards provide GPRS (general packet radio service) network access for notebook computers. PCMCIA GPRS cards and access software are provided by GPRS service providers. The GPRS card contains a GPRS radio, a slot for the SIM, and the firmware software. GPRS cards are installed as modems on notebook computers. Accessing SIM from a GPRS card for use in WLAN authentication is a very flexible and easy usage model. The biggest advantage is that the user already has a subscription for GPRS service. Also, since the SIM is always present in the GPRS card

there is no need for the user to physically take out the SIM card from another device, such as a cell phone, and place it in a GPRS card. As is the case with USB readers, there is no need to satisfy any additional regulatory requirements. However, the data path used by the application/authentication software to access the SIM over the AT modem interface is still not completely secure.

SIM Access from Mobile Handset over Bluetooth*: In this usage scenario the mobile notebook computer connects to the handset over a Bluetooth connection. The notebook has a SIM access client, which connects to the SIM access server on the handset. The SIM access server is preinstalled on the phone/handset. The request for SIM data is passed from the notebook to the SIM access server on the phone. The data between the links is encrypted using Bluetooth base band encryption. There are many steps involved in this, and not all phones currently support the required SIM access subsystems. This approach reuses the SIM on the cell phone and the user does not have a SIM reader attached to the notebook. The phone acts as a remote SIM reader connected over a Bluetooth connection. This usage scenario doesn't cover the potential hacking of the authentication software, and it has open system issues similar to those associated with the USB or GPRS card reader.

SIM Access from Reader Hardwired to Notebook: This method modifies the current notebook architecture and provides the SIM reader with a secure access to the device; for example, not over current open USB or PCI buses. This is not a very flexible approach, since the legacy systems will not be able to support these modifications. This method may also require full type approval (FTA). At present, FTA is required for WWAN (wireless wide area network) modules with integrated SIM. This architecture will require modifications to the existing WWAN module, thus requiring new FTA. This approach provides complete data path security.

10.3.2. Analysis of the USIM and Security Level

The requirement for the USIM in the context of the Trusted Mobile Device is to ensure that the USIM does not compromise the trustworthiness of the mobile device. It must be able to detect possible malicious software on the USIM and to prevent software that may; steal internal secrets (e.g., keys, user-USIM authentication data) and jeopardise the Trusted Mobile Device. The Trusted Mobile Device in turn must securely store the

secrets (Integrity and confidentiality) needed to establish radio channel protection.

Access to the USIM must be restricted to an authorised user or to a number of authorised users. To achieve this, the user and USIM must share a secret (e.g., PIN) that is stored securely within the USIM. When a user is trying to utilize USIM related functions, the user types in the PIN through a key pad and the PIN is sent to the USIM for verification. Trusted Mobile Devices should protect key pad as a part of the trusted I/O.

10.4. Biometric Authentication

10.4.1. Biometrics Introduction

What was once dreamed possible only on the big screens of Hollywood has become reality. In futuristic spy flicks, James Bond™ used some form of biometrical device to register/validate himself to that special gun which “Q“ provided for his mission. Since, then biometric authentication has received much public interest with varying commercial applications deployed for user authentication. There are countless automated methods whereby an individual's identity is confirmed by examining a unique physiological trait or behavioral characteristic, such as a fingerprint, facial pattern, or voice recognition. Physiological traits, such as finger prints and facial patterns are stable physical characteristics. Behavioral characteristics such as one's voice, or keystroke dynamics is influenced by both controllable actions and less controllable psychological factors. Although behavior-based biometrics can be less expensive and less intrusive to users, physiological based biometrics offer greater accuracy and security. In any case, both techniques provide more accurate user identification than does the use of pin codes or USIMs.

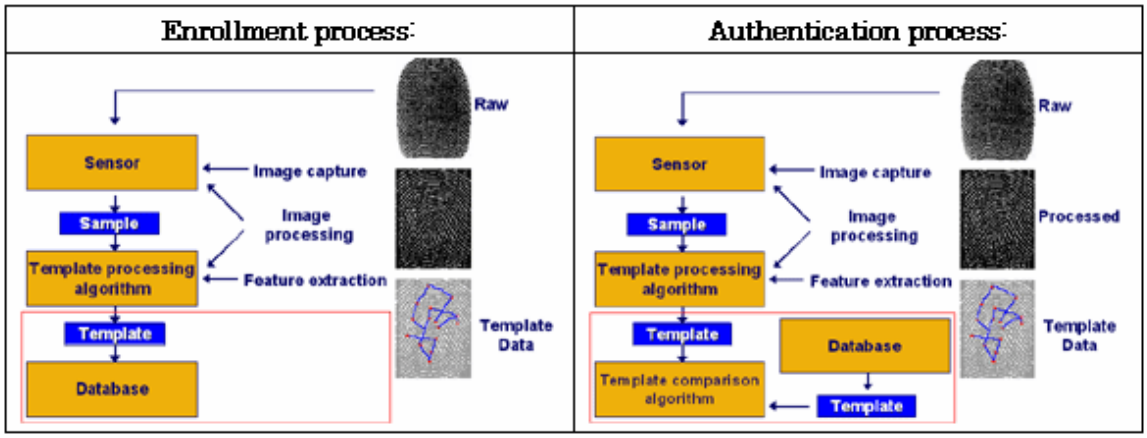

Figure 10-1 Biometric enrollment and authentication process

Cost effective implementation of user authentication using biometrics can be done through fingerprint identification, voice recognition or facial identification. As with most security solutions, there is simply no perfect technology. When deciding on a biometric authentication medium, the mobile device manufacturer should consider the following criteria among other basic factors:

- Accuracy - FRR/FAR/FER/EER (speed and maturity of biometric technology)
- User acceptance - personal, cultural, political
- Availability of technology
- Competition, or lack thereof
- Critical Vulnerabilities
- Standards support - AFIS, BAPI, FIPS, etc.
- Cost
- Manufacturer reputation and 'history'
- How critical is the data or physical location being secured

The performance of biometric based access systems using fingerprints or other means is limited by the performance of sensors and algorithms. When deploying biometrics, the manufacturer has to keep in mind public acceptance, which is often the key for deploying one or the other type of biometric technology. Regardless of how accurate the biometric enabled device is, if the system is difficult to use, users will become frustrated with resulting impact to training, marketing, maintenance, support and

implementation.

The use of multi-factor authentication provides the proper level of security for the Trusted Mobile Device. Selecting the right authentication technology or a combination thereof is indeed a complex matter. The decision in terms of functionality occurs on different levels. At the core of the biometric system is the biometric engine, a proprietary element based on the implementation the mobile device manufacturer chooses. The biometric engine core function is to extract and processes the biometric data. The mobile device may bind a mathematical algorithm (encryption) to the biometric signature. This can be processed at the TPM level using the appropriate API provided by the biometric engine.

A biometric system enables Verification and/or Identification of users. Verification validates a person's claimed identity by matching the captured biometric characteristic with its biometric template that was captured in advance by registering and storing the templates onto the system. The person needs to claim its identity via a personal identification number (PIN), login name, smart card, etc. The matching of verification is 1 to 1.

Identification recognizes an individual by searching all the templates that are stored in the database. The captured biometric characteristics are then matched with each template one after another and the best match is selected. The matching of Identification is 1 to N. Typically *identification* is more difficult than *verification*.

Before *verification* or *identification* is performed, the biometric characteristic for a user must be captured and processed to extract the critical features needed for recognition. The feature must be stored within the system with the associated identity of the individual as a template. This process is called *enrolment*.

10.4.2. Biometric Device Implementation Model

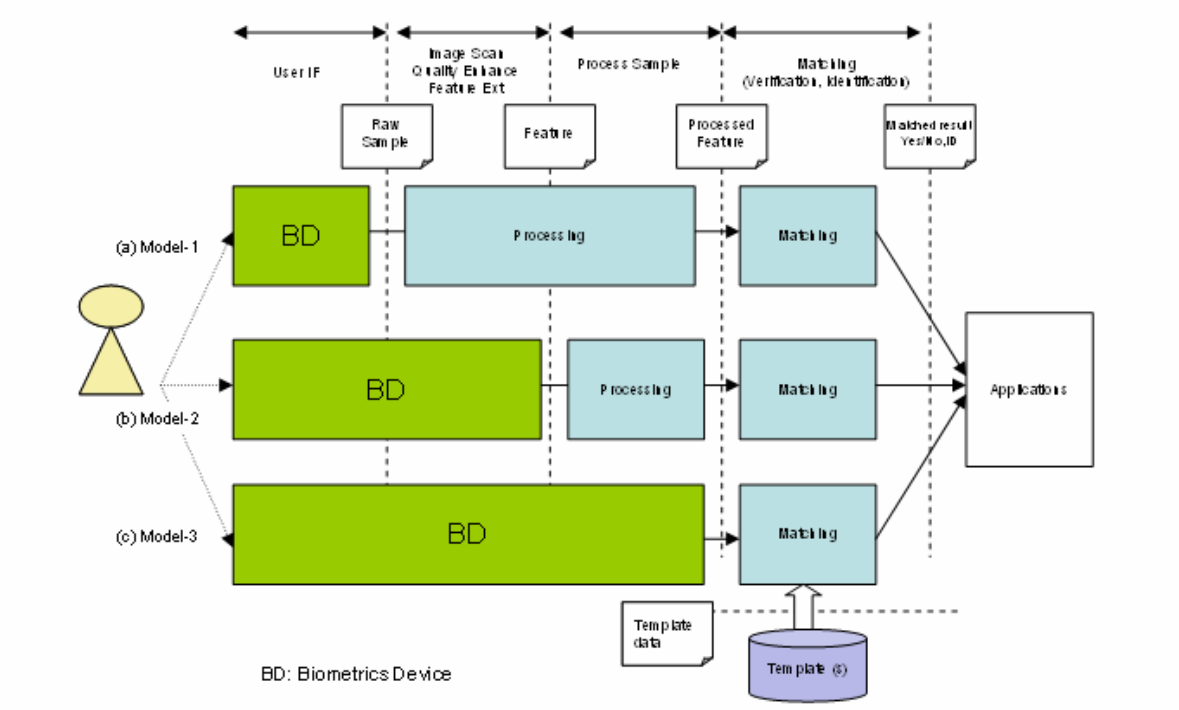

Figure 10-2 Generic biometric system models.

While there are various types of features that can be used, the basic system model is the same for all types of biometric systems. Figure 10-2 shows three possible implementation models. The variation reflects how much capability is integrated or embedded in the biometric device (BD). Mobile device manufactures are free to put whatever BD best fits their objectives depending on the level of security they would like to establish. The output of the BD is physically interfaced with the mobile device. The interface to the mobile device may be an internal bus, or an external interface such as a USB.

The BD in Model-1 consists only of capturing capabilities, (e.g., only a sensor, whose output is raw samples of the individual's biometric characteristic). The sensor captures live samples of the individual's biometric characteristics. The raw data is then processed to extract the critical features needed for recognition. The processing includes capabilities for input scanning, quality enhancement, and extraction of features. The output is then compared with the template(s). It is not feasible to reconstruct the original raw data from the feature data. The processing capabilities

that are outside of the BD are conducted by software which is typically resident within the mobile device.

The BD in Model-2 includes an earlier portion of the processing capabilities, which may include input scanning, quality enhancement and feature extraction. The BD in Model-3 includes all the capabilities for processing.

For *verification*, the output is matched with a template whose identity is claimed by the individual. The result is either YES or NO, and it is returned to an application that uses the result. For *identification* on the other hand, the output is matched with all the templates and the best matched template is used to identity the individual. The result is the identity which is returned to an application.

The templates are stored in a database. The integrity of the template itself, (e.g., the feature, the identity of individual and the binding of the two) must be protected by the TMD. Confidentiality of the database is often required depending on the target system configuration. The template database may be stored in the trusted mobile device or in an external memory device such as a smartcard.

10.4.3. Security Implication and Requirements

This section addresses security implication and requirements for biometric systems in the context of the Trusted Mobile Device.

10.4.3.1. Replacement of the Matching Result

It is possible for malicious software, which is installed on the mobile device, to replace the matching result with something else, so that the application which received the result may perform wrong functions. This can be avoided by using the integrity checking capabilities within the Trusted Mobile Device.

The Trusted Mobile Device must be able to detect such malicious software with the objective to prevent the following:

- Replacing the matching result with false data and forcing to the application perform wrong functions.

10.4.3.2. Falsification of Template Data

By replacing the template data with alternative data, it is possible for an attacker to be verified and or identified as a legitimate user. In order to prevent this type of attack, the following requirements are provided:

The TMD must provide integrity protection for:

- The template data,
- The identity
- The binding between both the template data and the identity.

In order to prevent stealing of the template data, confidentiality for both the template data and the identity is needed.

10.4.3.3. Stealing Data While Processing

It is possible to steal raw data that is captured by the sensor by probing or tampering with the interface between the BD and the Trusted Mobile Device. The stolen data may be used to produce false samples or to invade the user's privacy. In order to prevent this type of attack, the biometric system must not leave exposed or store the captured raw data in the TMD after processing to extract features is completed. The BD should employ some form of tamper resistant structure.

10.4.3.4. Enrolment Using a False Identification

It is possible for an individual to enroll claiming a false identity. This type of impersonation provides a legitimate template of an individual whose identity is false. If this is an issue, it must be resolved through operational policies associated with the enrolment process. For example, there must be a trusted authority that can authenticate the identity of the individual and their template. Therefore, there is no solution that the Trusted Mobile Device can provide for this problem. It is possible to mitigate the problem by using certain secrets such as a PIN for a SIM only after the secrets have been successfully shared between the legitimate user and the system.

10.4.3.5. Impersonation Using Stolen Samples

It is possible to create false samples that deceive the biometric system and result in the system making a false *verification* or *identification*. Such false samples may be made with or without the consent of the user. For example, it is possible to make a rubber stamp capturing the fingerprint made from a live-scan fingerprint image, or a wafer-thin plastic sheet housing a three-dimensional replication of a fingerprint [20]. Although there are several techniques to mitigate such threats (e.g. simultaneously measuring other physical traits such as temperature, pulse, blood stream, etc.) it is not known how these techniques can effectively protect from state of the art attacks. Therefore, there are no guaranteed means that can be required for the TMD to thwart such biometric attacks based on stolen fingerprint samples.

10.4.4. Summary of Biometric Systems and Security Level

The requirements for biometric systems in the context of Trusted Mobile Device are summarized as follows:

The Trusted Mobile Device must be able to detect malicious biometric software. The objective is to prevent software that may:

- Replace the matching result with false data and forces the application to perform wrong functions.

Integrity protection must be provided for:

- The template data
- The identity
- The binding between both the template data and the identity.

The biometric system must not leave or store the captured raw data in the device after it is processed to extract features. The Biometric Device should employ some form of tamper resistance.

10.5. External Memory Devices

10.5.1. External Memory Device Introduction

There are many types of external memory devices available. Applications of such memory devices include storage for digital cameras, digital music, movie distribution, etc. The following list shows typical external memory devices:

- Compact Flash
- Smart Media
- Memory Stick
- Multimedia Card
- SD Card

External memory devices are becoming essential storage peripherals for mobile devices as they incorporate advanced capabilities such as camera, music player, movie player, etc., which require large amounts of memory to store contents.

Some of the memory devices are simple memories and others (e.g. smart memory device) have intelligent capabilities in addition to simple basic data storage. A typical example of such smart memory devices is the SD card (Secure Digital Memory Card).

The SD card is a non-volatile memory with digital rights management capabilities built into it which is promoted by the SD Card Association (www.sdcard.org). The SD card enables secure downloading of all types of digital data, e.g., music, movies, photos, news, books, etc. It is designed to establish the digital rights management framework for such contents in addition to applications as a simple external memory device. The SD card complies with the CPRM specifications to ensure digital rights management. The CPRM is an abbreviation of Content Protection for Recordable Media and is promoted by the 4C Entity (http://www.4centity.com).

10.5.2. Security Implication and Requirements

This section addresses security implication and requirements of the smart memory device in the context of Trusted Mobile Devices.

10.5.2.1. Bus Interface between the Smart Memory and the device

There is no implication or requirement from a Trusted Mobile Device perspective concerning the bus interface between the SD card and the device. The transfer of data through the bus is secure, based on the shared secret provided by the MKBs and the device IDs.

10.5.2.2. Processing module in the device

In the processing module in the TMD, critical processing is performed including decrypting Media Keys, Session Keys, etc. Therefore, the integrity of this module shall be ensured. In addition, in order for the keys not to be stolen by eavesdropping, the confidentiality shall be protected.

- Integrity protection must be provided for the SD card processing module in the TMD.

- Confidentiality must be protected for all the data that is processed in the processing module in the TMD.

10.5.2.3. Input and output of content

10.5.2.3.1. Recording device

In the case of recording device, there are two types of inputs: encrypted content either compliant with the SD card or content protected by channel encryption and plain text content.

(a) Encrypted Content

If the input content is encrypted in such a way as is compliant with Smart Memory card

(or CPRM) specification, no encryption is necessary in the device and the input data is directly stored in the user area of the card. In this case, there is no security implication for the part of the content.

It should be noted how the key exchange is accomplished between the card and the originating device or the server, which is different from the device that is physically connected to the card. An AKE process must be performed between the card and the originating device. There must be a bidirectional communication channel which enables an AKE process. The AKE process itself is secure and may be conducted over general communication channels.

(b) Encrypted content – Channel encryption

The content may be transferred using a secure channel protection protocol such as TLS or SSL. In this case, the content must be first decrypted in the device and then encrypted into the smart memory card. There is a chance for an attacker to steal the content unless the integrity of the processing module and the confidentiality of the data being processed by the modules are ensured. The requirements are as follows:

- Integrity protection must be provided for the smart memory card processing modules in the device.

- Confidentiality must be protected for all the data that are processed in the processing modules in the device.

(c) Plain text content

If the incoming content is in plain text and the content is to be protected, the content should not be protected by the Smart Memory card, because the content may have already been stolen before it comes to the device. The requirement is defined as follows:

- The content to be protected by the Trusted Mobile Device combined with the Smart Memory card should be properly protected before it is given to the device.

10.5.2.3.2. Playback device

The encrypted content is sent to the playback device. The device decrypts it and renders (or plays) it.

(a) If the rendering device is a hardware component such as an LSI chip which is capable of taking encrypted content directly as an input, decrypt the content and render it all inside of the hardware device, there is no chance that the original content can be stolen.

(b) If software modules are used to decrypt the encrypted content and/or to render it, there is a chance for an attacker to steal the content unless the integrity of the processing modules and the confidentiality of the data being processed by the modules is protected. The requirements are as follows:

- Integrity protection must be provided for the Smart Memory card processing modules in the Trusted Mobile Device.

- Confidentiality must be provided for all the data that is processed in the processing modules executing in the TMD.

10.5.3. Summary of Smart Memory card System and Security Level

The requirements for the Smart Memory card system in the context of a Trusted Mobile Device are summarized as follows:

- Integrity protection must be provided for the Smart Memory card processing modules in the TMD.

- Confidentiality must be provided for all the data that is processed in the processing modules executing in the TMD.

- Content to be protected by the Trusted Mobile Device combined with the Smart Memory card should be properly protected before it is transferred to the storage

device.

Table 10-1 Smart Memory Card System and Security Level.

Security Level	Security Class 1	Security Class 2	Security Class 3
Integrity of Processing module	YES	YES	YES
Confidentiality of internal data	YES	YES	YES
Input data security	YES	YES	YES

10.6. Requirements for General Peripheral Devices

10.6.1. Requirements

10.6.1.1. External Interface Requirements

Peripheral devices are connected to the Trusted Mobile Device through physical external interfaces. It is possible for an attacker to monitor the signals that are transferred via the interface. It is also possible for an attacker to alter, replace, or reuse certain data. Hence, critical data that is transferred through the external interfaces must be protected from such attacks. The external interface requirement is defined below.

- Confidentiality and integrity must be provided for critical/sensitive data that is transferred via the external interface with a peripheral device. A mechanism must be provided to prevent replay attacks if needed.

10.6.1.2. Bus Master Interface

Certain external interfaces allow peripheral devices to become a bus master, e.g. Cardbus. If a peripheral device becomes a bus master, it can access data in the Trusted Mobile Device's physical memory space. Thus an attacker may use the peripheral device to steal the data or alter the data inside the mobile device. Such an attack must

be prevented.

- External interfaces that allow a peripheral device to be a bus master must be prohibited.

10.6.2. Summary

The requirements for general peripheral device in the context of a Trusted Mobile Device are summarized as follows:

Table 10-2 General Peripheral Device Requirements and Security Level.

Security Level	Security Class 1	Security Class 2	Security Class 3
Protection of interface data	YES	YES	YES
Prohibit bus master I/F	YES	YES	YES

Appendix A. Change History (Informative)

Type of Change	Date	Section	Description
Rev 1.00	06/22/04		Revision 1.0 for release