INTERNATIONAL STANDARD

ISO/IEC 11889-1

First edition
2009-05-15

Information technology — Trusted Platform Module —

Part 1: Overview

Technologies de l'information — Module de plate-forme de confiance — Partie 1: Aperçu général

Reference number
ISO/IEC 11889-1:2009(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Table of Contents

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 11889-1 was prepared by the Trusted Computing Group (TCG) and was adopted, under the PAS procedure, by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, in parallel with its approval by national bodies of ISO and IEC.

ISO/IEC 11889 consists of the following parts, under the general title *Information technology — Trusted Platform Module*:

— *Part 1: Overview*

— *Part 2: Design principles*

— *Part 3: Structures*

— *Part 4: Commands*

Introduction

Designers of secure distributed systems, when considering the exchange of information between systems, must identify the endpoints of communication. The composition and makeup of the endpoint is as important to the overall ability of the system to serve as an authentication and attestation device of the system as is the communications protocol.

Endpoints are minimally comprised of asymmetric keys, key storage and processing that protects protocol data items. Classic message exchange based on asymmetric cryptography suggests that messages intended for one and only one individual can be encrypted using a public key. Furthermore, the message can be protected from tampering by signing with the private key.

Keys are communication endpoints and improperly managed keys can result in loss of attestation and authentication. Additionally, improperly configured endpoints may also result in loss of attestation and authentication ability.

This is an informative background document and contains no specifications or normative information. To find normative information and specifications about the TPM, refer to ISO/IEC 11889-2 to ISO/IEC 11889-4.

A Trusted Platform Module (TPM) is an implementation of a defined set of capabilities that is intended to provide authentication and attestation functionality for a computing device, and protect information by controlling access to plain-text data.

A TPM is self-sufficient as a source of authentication and as a means of enhancing the protection of information from certain physical attacks. A TPM requires the cooperation of a TCG “Trusted Building Block” (outside the TPM, that is also part of the computing device) in order to provide attestation and protect information from software attacks on the computing device.

Typical TPM implementations are affixed to the motherboard of a computing device.

A computing device that contains both a TPM and a Trusted Building Block is called a Trusted Platform. Trusted Platforms offer improved, hardware-based security in numerous applications, such as file and folder encryption, local password management, S-MIME e-mail, VPN and PKI authentication and wireless authentication for 802.1x and LEAP.

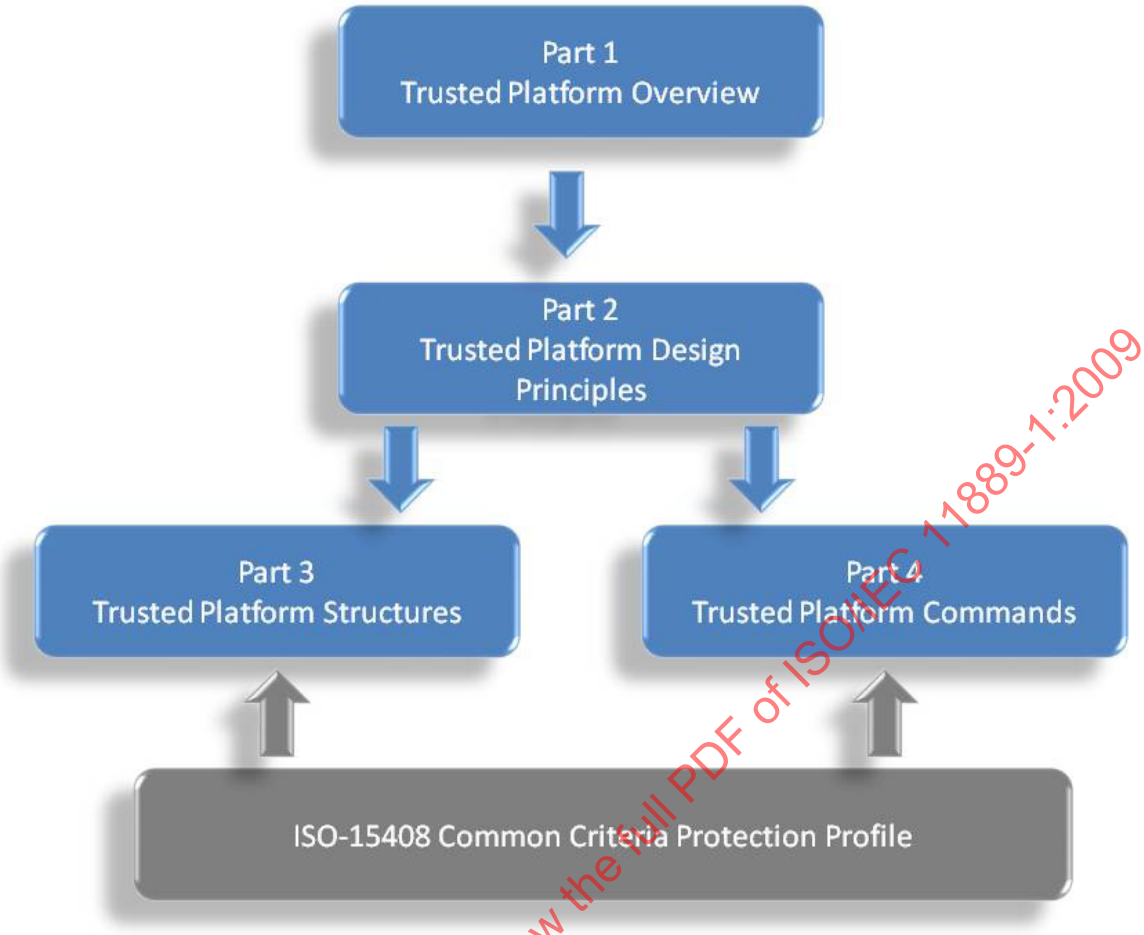

Figure 1. TPM Documentation Roadmap

Start of informative comment

ISO/IEC 11889 is from the Trusted Computing Group (TCG) Trusted Platform Module (TPM) specification 1.2 version 103. The part numbers for ISO/IEC 11889 and the TCG specification do not match. The reason is the inclusion of the Overview document that is not a member of the TCG part numbering. The mapping between the two is as follows:

ISO Reference	TCG Reference
Part 1 Overview	Not published
Part 2 Design Principles	Part 1 Design Principles
Part 3 Structures	Part 2 Structures
Part 4 Commands	Part 3 Commands

End of informative comment

Information technology — Trusted Platform Module —

Part 1: Overview

1. Scope

ISO/IEC 11889 defines the Trusted Platform Module (TPM), a device that enables trust in computing platforms in general. ISO/IEC 11889 is broken into parts to make the role of each document clear. Any version of the standard requires all parts to be a complete standard.

A TPM designer MUST be aware that for a complete definition of all requirements necessary to build a TPM, the designer MUST use the appropriate platform specific specification for all TPM requirements.

Part 1 provides an overview of the concepts behind the TPM and trusted platforms.

2. Abbreviated Terms

Abbreviation	Description
AACP	Asymmetric Authorization Change Protocol
ADCP	Authorization Data Change Protocol
ADIP	Authorization Data Insertion Protocol
AIK	Attestation Identity Key
AMC	Audit Monotonic Counter
APIP	Time-Phased Implementation Plan
AuthData	Authentication Data or Authorization Data, depending on the context
BCD	Binary Coded Decimal
BIOS	Basic Input/Output System
CA	Certification of Authority
CDI	Controlled Data Item
CMK	Certifiable/Certified Migratable Keys
CRT	Chinese Remainder Theorem
CRTM	Core Root of Trust Measurement
CTR	Counter-mode encryption
DAA	Direct Autonomous Attestation
DIR	Data Integrity Register
DOS	Disk Operating System

Abbreviation	Description
DSA	Digital Signature Algorithm
DSAP	Delegate-Specific Authorization Protocol
ECB	Electronic Codebook Mode
EK	Endorsement Key
ET	ExecuteTransport or Entity Type
FIPS	Federal Information Processing Standard
GPIO	General Purpose I/O
HMAC	Hash Message Authentication Code
HW	Hardware Interface
IB	Internal Base
I/O	Input/Output
IV	Initialization Vector
KH	Key Handle
LEAP	Lightweight Extensible Authentication Protocol for wireless computer networks
LK	Loaded Key
LOM	Limited Operation Mode
LPC	Low Pin Count
LSB	Least Significant Byte
MA	Migration Authority/Authorization
MIDL	Microsoft Interface Definition Language
MSA	Migration Selection Authority
MSB	Most Significant Byte
NV	Non-volatile
NVRAM	Non-Volatile Random Access Memory
OAEP	Optimal Asymmetric Encryption Padding
OEM	Original Equipment Manufacturer
OIAP	Object-Independent Authorization Protocol
OID	Object Identifier
OSAP	Object-Specific Authorization Protocol
PCR	Platform Configuration Register
PI	Personal Information
PII	Personally Identifiable Information
POST	Power On Self Test
PRIVEK	Private Endorsement Key
PRNG	Pseudo Random Number Generator

Abbreviation	Description
PSS	Probabilistic Signature Scheme
PUBEK	Public Endorsement Key
RNG	Random Number Generator
RSA	Algorithm for public-key cryptography. The letters R, S, and A represent the initials of the first public describers of the algorithm.
RTM	Release to Manufacturing/Ready to Market
RTR	Root of Trust for Reporting
RTS	Root of Trust for Storage
SHA	Secure Hash Algorithm
SRK	Storage Root Key
STF	Self Test Failed
TA	Time Authority
TBB	Threading Building Blocks
TCG	Trusted Computing Group
TCV	Tick Count Value
TIR	Tick Increment Rate
TIS	TPM Interface Specification
TNC	Trusted Network Connect
TOE	Target of Evaluation
TOS	Trusted Operating System
TPCA	Trusted Platform Computing Alliance
TPM	Trusted Platform Module
TPME	Trusted Platform Module Entity
TSC	Tick Stamp Counter
TSC_	TPM Software Connection, when used as a command prefix
TSN	Tick Session Name
TSR	Tick Stamp Reset
TSRB	TickStampReset for blob
TSS	TCG Software Stack
TTP	Trusted Third Party/Time-Triggered Protocol
TS	Tick Stamp
UTC	Universal Time Clock
VPN	Virtual Private Network

3. The Trusted Platform

Trust in the context of “Trusted Platforms” is the expectation that a device will behave in a particular manner for a specific purpose.

In Trusted Platforms, Roots of Trust are components that must be trusted because misbehavior may not be detected. A complete set of Roots of Trust has at least the minimum functionality necessary to describe the platform characteristics that affect the trustworthiness of the platform.

There are commonly three Roots of Trust in a trusted platform; a root of trust for measurement (RTM), root of trust for storage (RTS) and root of trust for reporting (RTR). The RTM is a computing engine capable of making inherently reliable integrity measurements. Typically the normal platform computing engine, controlled by the core root of trust for measurement (CRTM).

The CRTM is the instructions executed by the platform when it acts as the RTM. The RTM is also the root of the chain of transitive trust. The RTS is a computing engine capable of maintaining an accurate summary of values of integrity digests and the sequence of digests. The RTR is a computing engine capable of reliably reporting information held by the RTS.

Each root is trusted to function correctly without external oversight. Trusting “roots of trust” may be achieved through a variety of ways but is anticipated to include technical evaluation by competent experts.

3.1 Trusted Platform Building Block

The Trusted Building Block (TBB) is the parts of the Roots of Trust that do not have shielded locations. Normally these include just the instructions for the RTM and TPM initialization functions (reset, etc.). Typically they are platform-specific.

One example of a TBB is the combination of the CRTM, connection of the CRTM storage to a motherboard, the connection of the TPM to a motherboard, and mechanisms for determining Physical Presence

The TBB is trusted, meaning it is expected to behave in a way that doesn’t compromise the goals of trusted platforms.

3.2 The Trust Boundary

The combination of TBB and Roots of Trust form a trust boundary where measurement, storage and reporting can be accomplished for a minimal configuration. More complex systems may require measurements be taken by other (optional) ROM code besides the CRTM. For this to occur trust in other ROM code must be established. This is done by measuring the ROM code prior to transferring execution control. The TBB should be established such that devices containing other measurement code do not inadvertently extend the TBB boundary where trustworthiness of the linkages has not been previously established.

3.3 Transitive Trust

Transitive trust (also known as “Inductive Trust”), is a process where the Root of Trust gives a trustworthy description of a second group of functions.

Based on this description, an interested entity can determine the trust it is to place in this second group of functions. If the interested entity determines that the trust level of the second group of functions is acceptable, the trust boundary is extended from the Root of Trust to include the second group of functions.

In this case, the process can be iterated. The second group of functions can give a trustworthy description of the third group of functions, etc. Transitive trust is used to provide a trustworthy description of platform characteristics, and also to provide evidence that non-migratable keys are non-migratable

3.3.1 Basic Trusted Platform features

A trusted platform should provide at least three basic features:

1. Protected storage
2. Integrity measurement
3. Integrity reporting

All three of these functions are related to attestation, which is the process of vouching for the accuracy of information. All forms of attestation require reliable evidence of the attesting entity. This can be provided by shipping TPMs with an embedded key called the Endorsement Key (EK). The EK is used in a process for the issuance of credentials for another type of key, called an Attestation Identity Key (AIK). A platform can attest to its description of platform characteristics that affect the integrity (trustworthiness) of a platform.

External entities can attest to shielded locations, protected capabilities, and Roots of Trust.

Attestation can be understood in four dimensions: Attestation by the TPM, attestation to the platform, attestation of the platform and authentication of the platform.

- **Attestation by the TPM** is an operation that provides proof of data known to the TPM. This is done by digitally signing specific internal TPM data using an Attestation Identity Key (AIK). The acceptance and validity of both the integrity measurements and the AIK itself are determined by a verifier.
- **Attestation to the platform** is an operation that provides proof that a platform can be trusted to report integrity measurements. It is performed using the set or subset of the credentials associated with the platform and used to issue an AIK credential.
- **Attestation of the platform** is an operation that provides proof of a set of the platform's integrity measurements. This is done by digitally signing a set of measurements using an AIK.
- **Authentication of the platform** provides evidence of a claimed platform identity. The claimed identity may or may not be related to a user or any actions performed by the user. Platform Authentication is performed using any signing key that cannot be removed from a TPM. Certified keys (i.e. keys signed by an AIK) have the added semantic of being attestable. Since there are an unlimited number of such keys associated with the TPM, there are an unlimited number of ways that a platform can be authenticated.

3.3.1.1 Protected Storage

The Root of Trust for Storage (RTS) protects keys and data entrusted to the TPM. The RTS manages a small amount of volatile memory where keys are held while performing signing and decryption operations.

Inactive keys may be encrypted and moved off-chip to make room for other more active keys. Management of the key slot cache is performed external to the TPM by a Key Cache Manager (KCM). The KCM interfaces with a storage device where inactive keys may be stored indefinitely. The RTS doubles as a general purpose protected storage service allowing opaque data also to be stored.

The RTS is optimized to store small objects roughly the size of an asymmetric key minus overhead (e.g. ~210 byte payload). A variety of object types can be stored, such asymmetric and asymmetric keys, pass-phrases, cookies, authentication results and opaque data. There are three key types that are not opaque to the TPM. AIK keys, Signing keys and Storage keys.

The Storage Root Key (SRK) is embedded in the TPM and cannot be removed from the TPM, but can be erased. However, a new SRK may be created as part of creating a new platform owner. This has the side-effect of leaving encrypted all data objects controlled by the previous SRK. The Storage Root Key is the root of a hierarchy of encrypted keys, where each parent key is used to encrypt (wrap) its child keys.

AIKs are direct children of the SRK. They are used to sign integrity measurements that have been gathered by the platform, and to sign certificates describing other keys that cannot leave the TPM. TPMs can have as many or as few AIK keys as are required. This protect privacy when the platform owner is concerned about the consequences of collusion by entities that receive signed information from a Trusted Platform.

3.3.1.2 Integrity Measurement, Logging and Reporting

Integrity measurement is the process of obtaining metrics of platform characteristics that affect the integrity (trustworthiness) of a platform and putting digests of those metrics in PCRs.

The starting point of measurement is called the root of trust for measurement. A static root of trust for measurement begins measuring from a well-known starting state such as a power on self-test. A dynamic root of trust for measurement transitions from an un-trusted state to one that is trusted.

An optional intermediate step between integrity measurement and integrity reporting is integrity logging, which stores integrity metrics in a log for later use. Logging is recommended. Otherwise integrity measurements might need to be repeated in order to interpret PCR values.

Integrity reporting is the process of attesting to integrity measurements recorded in PCRs. The philosophy of integrity measurement, logging and reporting is that a platform may be permitted to enter any state possible including undesirable or insecure states, but that it may not be permitted to lie about states that it was or was not in. An independent process may evaluate the integrity state(s) and determine an appropriate response.

3.4 Integrity Measurement

A measurement kernel generates measurement events.

A measurement event consists of two classes of data

1. Measured values - a representation of embedded data or program code.
2. Measurement digests - a hash of those values.

Data are scanned by the measurement kernel, which generates a message digest. Digests are snapshots of the machine's operational state. The two data elements (measured values and measurement digest) are stored separately. The measurement digest is stored in the TPM using RTR and RTS functionality. The measured values may be stored virtually anywhere at the discretion of the measurement kernel. In fact, the measurement may not be stored at all, but re-computed whenever the serialized representation is needed.

Measurement data describe properties and characteristics of the measured components. It is the responsibility of the measurement kernel implementer to understand the syntax and semantics of measured fields in sufficient detail to produce an encoding suitable for measurement event consumers.

Implementers play a role in determining how event data may be partitioned. TCG's platform specific specifications contain additional insight in specifying the platform configuration, its representation and anticipated measurement consumers.

The *Stored Measurement Log (SML)* contains sequences of related measured values. Each sequence shares a common measurement digest. Measured values are appended to the common measurement digest and re-hashed. This is more commonly referred to as extending the digest. Extending ensures related measured values will not be ignored and order of operations is preserved.

Data encoding rules for SML contents are not defined, but following appropriate standards such as Extensible Markup Language (XML) to ensure broad accessibility is recommended. Nevertheless, different platforms may require different representation, hence the Platform Specific Specifications (e.g., the PC-Specific Platform Specification) may define other encoding rules.

The SML can become very large. Therefore it does not reside in the TPM.

3.5 Integrity Reporting

The TPM contains a set of registers, called Platform Configuration Registers (PCRs), that are "extended" (as described above) with the same measured values that are stored in the SML. Algebraically, updates to a PCR follows as:

PCR[n] := SHA-1 (PCR[n] || measured data),

where PCR[n] denotes the n-th PCR, := denotes assignment and || denotes concatenation.

A PCR therefore contains a summary of all the measured values and their ordering. PCR values are temporal and are reset at system reboot. PCRs may be implemented in volatile or non-volatile storage. PCRs must be protected from software attack. Steps to prevent physical tampering with PCRs should be taken into consideration.

.

PCR values are digitally signed using Attestation Identity Keys (AIK) to authenticate them. A nonce is included with the signed PCRs to prevent replay.

Verification of measurement events requires recreation of the measurement digest and a simple compare of digest values (using the PCR value as one of the comparators).

The SML does not need the protection afforded by the TPM as attacks against the SML would be detected. However, SML is still subject to denial of service attacks. Implementers should take steps to replicate or regenerate the log.

4. The TPM

4.1 Cryptographic Algorithms Required with TPM

The TPM requires use of cryptographic algorithms to provide confidentiality, integrity and authenticity. Currently, the specification uses RSA, SHA-1, HMAC, and symmetric algorithms.

Symmetric algorithms are used for some purposes (instead of asymmetric algorithms), because they improves TPM performance. The symmetric algorithms cannot be used outside the TPM for general purpose encryption. The Trusted Computing Group will continue to evaluate developments in cryptography.

4.1.1 Algorithm Assumptions

This specification makes explicit security assumptions on cryptographic algorithms.

The RSA assumptions are:

- RSA, when used for signatures with the signature padding schemes PKCS#1 v1.5 or PSS, is assumed to be secure against adaptive forgery attacks.
- RSA, when used for encryption with the encryption padding schemes OAEP, is assumed to be secure against adaptive chosen ciphertext attacks.

The SHA-1 assumptions are:

- SHA-1 is assumed to be secure against finding collisions when used with RSA signatures and for computing measurements.
- SHA-1 is assumed to provide a secure one-way function when used for MGF1.
- SHA-1 is assumed to provide a secure MAC when used with the HMAC construction.

4.2 Operating Systems Supported by TPM

Specifications are Operating System-agnostic. Software stacks for multiple operating systems are available. In addition to work on the PC platform, TCG has specifications for Trusted Servers and mobile devices and is working to finalize specifications for other computing devices, including storage and infrastructure.

4.3 Protected Capabilities

Protected capabilities are a set of commands with exclusive permission to access shielded locations. Shielded locations are places (memory, register, etc.) where it is safe to operate on sensitive data (data locations that can be accessed only by protected capabilities).

The TPM requires protected capabilities and shielded locations to ensure that its functions are separated from other processes in the computing device, and hence protected from other processes in the computing device.

Protected capabilities and shielded locations are used to protect and report digests of integrity measurements (called Platform Configuration Registers: PCRs), for example. The TPM also uses protected capabilities and shielded locations to store and manipulate the cryptographic keys used to authenticate reported integrity measurements.

TPM protected capabilities include additional security functionality such as cryptographic key management, random number generation, sealing data to system state and others as determined necessary by TCG members.

4.4 Trusted Platform Module components

This section describes the logical layout of the TPM and its discrete components. Implementations of TPMs may be done in hardware or software.

A model that favors hardware interpretations of the TPM specification is presented, but it is sufficiently abstract so as not to exclude software implementations.

As a building block of a trusted platform, TPM components are trusted to work properly without additional oversight. Trust in these components is derived from good engineering practices, manufacturing process and industry review. The manufacturer of the TPM may evaluate the implementation using an evaluated Common Criteria protection profile.

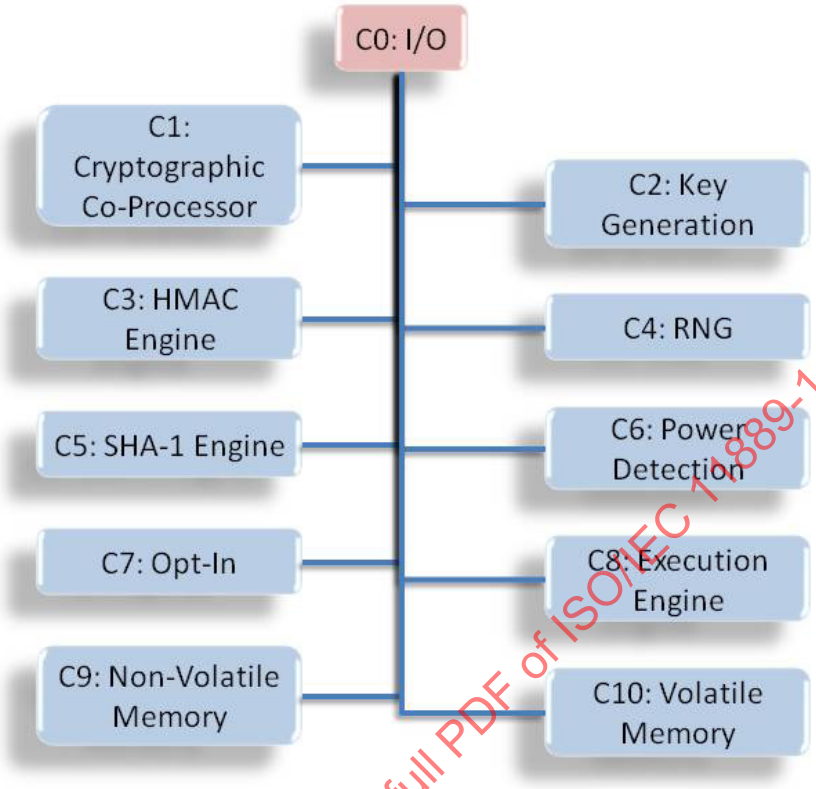

Figure 2: TPM Component Architecture

4.5 Naming Conventions

The TPM specification follows a stylized convention when referring to functionality of various components.

Terms such as ***command***, function, operation and interface are frequently encountered in the TPM Specification.

- Command: discrete functionality of the TPM exposed externally and recognizable by a TPM's command processor. TPM commands are enumerated and have an ordinal value associated.
- Function: discrete functionality of non-TPM modules having programmatic interfaces.
- Operation: refers to a sequence of steps or protocol flow that may be implemented using one or more commands or functions.
- Interface: The set of command or function entry points, including parameters and return codes, to a particular module. When used in singular context, Interface may refer to a single entry point. Operations are classified according to security relevance. They are:

a. Protected Operations: Operations affecting the security properties of TPM platforms. These include all TPM commands. TPM command interface names begin with the "TPM_" prefix.

b. Unprotected Operations: Operations that support TPM protected functionality, but are not TPM commands. These are normally implemented outside the TPM. Function interface names begin with a prefix that is not "TPM_".

c. Connection Operations: Operations involving platform to TPM binding. Commands implementing connection operations are typically defined in the Platform-Specific specifications.
These command interface names begin with the "TSC_" prefix. For example, TSC_PhysicalPresence() includes functionality defined only in the platform-specific specification. TSC stands for "TCG Software Connection".

Functions that extend, layer or encapsulate TPM functionality are prefixed with the module name.