

# IEEE Standard for Secure Computing Based on Trusted Execution Environment

IEEE Consumer Technology Society

Developed by the  
Emerging Technology Standards Committee

IEEE Std 2952™-2023

# IEEE Standard for Secure Computing Based on Trusted Execution Environment

Developed by the

**Emerging Technology Standards Committee**  
of the  
**IEEE Consumer Technology Society**

Approved 30 March 2023

**IEEE SA Standards Board**

**Abstract:** A framework of TEE-based secure computing system, and technical requirements of a general secure computing platform for isolation, confidentiality, compatibility, performance, usability, and security aspects are specified in this standard. Use cases and scenarios of secure computing technology are also specified.

**Keywords:** IEEE 2952™, secure computing, trusted execution environment

---

The Institute of Electrical and Electronics Engineers, Inc.  
3 Park Avenue, New York, NY 10016-5997, USA

Copyright © 2023 by The Institute of Electrical and Electronics Engineers, Inc.  
All rights reserved. Published 14 July 2023. Printed in the United States of America.

IEEE is a registered trademark in the U.S. Patent & Trademark Office, owned by The Institute of Electrical and Electronics Engineers, Incorporated.

PDF: ISBN 978-1-5044-9756-5 STD26192  
Print: ISBN 978-1-5044-9757-2 STDPD26192

*IEEE prohibits discrimination, harassment, and bullying.*

*For more information, visit <https://www.ieee.org/about/corporate/governance/p9-26.html>.*

*No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.*

## Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE Standards documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page (<https://standards.ieee.org/ipr/disclaimers.html>), appear in all standards and may be found under the heading “Important Notices and Disclaimers Concerning IEEE Standards Documents.”

### Notice and Disclaimer of Liability Concerning the Use of IEEE Standards Documents

IEEE Standards documents are developed within IEEE Societies and subcommittees of IEEE Standards Association (IEEE SA) Board of Governors. IEEE develops its standards through an accredited consensus development process, which brings together volunteers representing varied viewpoints and interests to achieve the final product. IEEE standards are documents developed by volunteers with scientific, academic, and industry-based expertise in technical working groups. Volunteers are not necessarily members of IEEE or IEEE SA and participate without compensation from IEEE. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

IEEE makes no warranties or representations concerning its standards, and expressly disclaims all warranties, express or implied, concerning this standard, including but not limited to the warranties of merchantability, fitness for a particular purpose and non-infringement. IEEE Standards documents do not guarantee safety, security, health, or environmental protection, or guarantee against interference with or from other devices or networks. In addition, IEEE does not warrant or represent that the use of the material contained in its standards is free from patent infringement. IEEE Standards documents are supplied “AS IS” and “WITH ALL FAULTS.”

Use of an IEEE standard is wholly voluntary. The existence of an IEEE standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard.

In publishing and making its standards available, IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity, nor is IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing any IEEE Standards document, should rely upon their own independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

IN NO EVENT SHALL IEEE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: THE NEED TO PROCURE SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE PUBLICATION, USE OF, OR RELIANCE UPON ANY STANDARD, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

## Translations

The IEEE consensus balloting process involves the review of documents in English only. In the event that an IEEE standard is translated, only the English version published by IEEE is the approved IEEE standard.

## Official statements

A statement, written or oral, that is not processed in accordance with the IEEE SA Standards Board Operations Manual shall not be considered or inferred to be the official position of IEEE or any of its committees and shall not be considered to be, nor be relied upon as, a formal position of IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that the presenter's views should be considered the personal views of that individual rather than the formal position of IEEE, IEEE SA, the Standards Committee, or the Working Group. Statements made by volunteers may not represent the formal position of their employer(s) or affiliation(s).

## Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE or IEEE SA. However, **IEEE does not provide interpretations, consulting information, or advice pertaining to IEEE Standards documents.**

Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is important that any responses to comments and questions also receive the concurrence of a balance of interests. For this reason, IEEE and the members of its Societies and subcommittees of the IEEE SA Board of Governors are not able to provide an instant response to comments, or questions except in those cases where the matter has previously been addressed. For the same reason, IEEE does not respond to interpretation requests. Any person who would like to participate in evaluating comments or in revisions to an IEEE standard is welcome to join the relevant IEEE working group. You can indicate interest in a working group using the Interests tab in the Manage Profile and Interests area of the [IEEE SA myProject system](#).<sup>1</sup> An IEEE Account is needed to access the application.

Comments on standards should be submitted using the [Contact Us](#) form.<sup>2</sup>

## Laws and regulations

Users of IEEE Standards documents should consult all applicable laws and regulations. Compliance with the provisions of any IEEE Standards document does not constitute compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

## Data privacy

Users of IEEE Standards documents should evaluate the standards for considerations of data privacy and data ownership in the context of assessing and using the standards in compliance with applicable laws and regulations.

<sup>1</sup>Available at: <https://development.standards.ieee.org/myproject-web/public/view.html#landing>.

<sup>2</sup>Available at: <https://standards.ieee.org/content/ieee-standards/en/about/contact/index.html>.

## Copyrights

IEEE draft and approved standards are copyrighted by IEEE under US and international copyright laws. They are made available by IEEE and are adopted for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making these documents available for use and adoption by public authorities and private users, neither IEEE nor its licensors waive any rights in copyright to the documents.

## Photocopies

Subject to payment of the appropriate licensing fees, IEEE will grant users a limited, non-exclusive license to photocopy portions of any individual standard for company or organizational internal use or individual, non-commercial use only. To arrange for payment of licensing fees, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400; <https://www.copyright.com/>. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

## Updating of IEEE Standards documents

Users of IEEE Standards documents should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect.

Every IEEE standard is subjected to review at least every 10 years. When a document is more than 10 years old and has not undergone a revision process, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE standard.

In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit [IEEE Xplore](#) or [contact IEEE](#).<sup>3</sup> For more information about the IEEE SA or IEEE's standards development process, visit the IEEE SA Website.

## Errata

Errata, if any, for all IEEE standards can be accessed on the [IEEE SA Website](#).<sup>4</sup> Search for standard number and year of approval to access the web page of the published standard. Errata links are located under the Additional Resources Details section. Errata are also available in [IEEE Xplore](#). Users are encouraged to periodically check for errata.

## Patents

IEEE standards are developed in compliance with the [IEEE SA Patent Policy](#).<sup>5</sup>

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. If a patent holder or patent applicant has

<sup>3</sup>Available at: <https://ieeexplore.ieee.org/browse/standards/collection/ieee>.

<sup>4</sup>Available at: <https://standards.ieee.org/standard/index.html>.

<sup>5</sup>Available at: <https://standards.ieee.org/about/sasb/patcom/materials.html>.

filed a statement of assurance via an Accepted Letter of Assurance, then the statement is listed on the IEEE SA Website at <https://standards.ieee.org/about/sasb/patcom/patents.html>. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims, or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

## **IMPORTANT NOTICE**

Technologies, application of technologies, and recommended procedures in various industries evolve over time. The IEEE standards development process allows participants to review developments in industries, technologies, and practices, and to determine what, if any, updates should be made to the IEEE standard. During this evolution, the technologies and recommendations in IEEE standards may be implemented in ways not foreseen during the standard's development. IEEE standards development activities consider research and information presented to the standards development group in developing any safety recommendations. Other information about safety practices, changes in technology or technology implementation, or impact by peripheral systems also may be pertinent to safety considerations during implementation of the standard. Implementers and users of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, and interference protection practices and all applicable laws and regulations.

## Participants

At the time this standard was completed, the Secure Computing based on Trusted Execution Environment Working Group had the following membership:

**Wenting Chang, Chair**  
**Yang Gao, Vice Chair**

<i>Organization Represented</i>	<i>Name of Representative</i>
Alipay (China) Technology Co., Ltd.....	Wenting Chang
Beijing Baidu Netcom Science and Technology Co.,Ltd. ....	Jiwen Zhou
China Zheshang Bank Co., Ltd. ....	Cheng Zang
Hangzhou Qulian Technology Co., Ltd.....	Xiaofeng Chen
Hangzhou Nuowei Information Technology Co., Ltd. ....	Danye Tang
Huawei Technologies Co., Ltd.....	Dongyang Xu
Impulse Online Co., Ltd. ....	Haodong Chen
InsightOne Tech Co., Ltd. ....	Yukun Wang
Intel Corporation .....	Ligang Wang
Lenovo Group Limited.....	Yunhao Wang
Shanghai Pudong Development Bank .....	Yang Gao
Senses Global Corp. ....	Qi Wang
State Grid Corporation of China (SGCC).....	Yong Yan
Shanghai Jiao Tong University.....	Yubin Xia
Shandong Computer Science Center .....	Zhen Zhang
Sichuan Changhong Electric Co., Ltd. ....	Bo Tang
Terminus Group Co. Ltd .....	Yu Yang
Wuxi SensingNet Industrialization Research Institute .....	Mingjuan Wu
Zhejiang University.....	Bingsheng Zhang

The Working Group gratefully acknowledges the contributions of the following participants. Without their assistance and dedication, this standard would not have been completed.

Fen Bao	Wei Li	Guozheng Yang
Liang Cai	Chengsui Lu	Ming Yao
Jiajun Chen	James R. Quaranta, Jr.	Ying Yao
Qingxiao Guo	Bincheng Shuai	Xu Yin
Hao He	Qi Sun	Huan Yu
Zhichao Hua	Shuang Wang	Jingzhi Zhang
Dejun Huang	Junxian Xiao	Xiaomeng Zhang
Hui Jin	Jing Xu	Yingwei Zhang
Fan Li	Shoumeng Yan	Yuan Zhao



The following members of the entity Standards Association balloting group voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

OxSenses Corporation  
Beijing Impulse Online  
Technology Co., Ltd.  
Beijing Tsingzhi Shuyuan  
Tech Inc.

Hangzhou Qulian  
Technology Co., Ltd.  
InsightOne Tech Co., Ltd.  
Institute of Biomedical  
Engineering

Shandong Computer Science  
Center (National  
Supercomputing Center  
in Jinan )  
Zhejiang University

When the IEEE SA Standards Board approved this standard on 30 March 2023, it had the following membership:

**David J. Law**, *Chair*  
**Ted Burse**, *Vice Chair*  
**Gary Hoffman**, *Past Chair*  
**Konstantinos Karachalios**, *Secretary*

Sara R. Biyabani  
Ted Burse  
Doug Edward  
Ramy Ahmed Fathy  
Guido R. Hiertz  
Yousef Kimiagar  
Joseph L. Koepfinger\*  
Thomas Koshy

John D. Kulick  
Joseph S. Levy  
Howard Li  
Johnny Daozhuang Lin  
Gui Lin  
Xiaohui Liu  
Kevin W. Lu  
Daleep C. Mohla  
Andrew Myles

Paul Nikolich  
Annette D. Reilly  
Robby Robson  
Lei Wang  
F. Keith Waters  
Karl Weber  
Philip B. Winston  
Don Wright

\*Member Emeritus

## Introduction

This introduction is not part of IEEE Std 2952-2023, IEEE Standard for Secure Computing Based on Trusted Execution Environment.
---

As the global data is geometrically increased day by day, the dilemma between data sharing and data confidentiality becomes more and more serious. During data mining and value-added data derivation, it is critical to protect the data security. Secure computing based on a trusted execution environment can help prevent the disclosure and abuse of data when conducting calculation tasks.

The demand for secure computing comes from many aspects. First, it comes from the users who need to protect their data security in an untrusted environment. Second, there is a defense requirement from the enterprise itself, which needs to protect the data security against increased internal and external attacks. Third, in the data sharing scenario among multiple organizations, different partners with insufficient mutual trust still desire to cooperate with each other to train more intelligent models. Currently, there are many scenarios in need of secure computing technology. However, there is a lack of relevant standards to specify the definition, technical framework, and security characteristics.

This standard specifies a framework of TEE-based secure computing systems, and defines the corresponding requirements in terms of isolation, confidentiality, compatibility, performance, usability, and security.

## Contents

1. Overview .....	11
1.1 Scope .....	11
1.2 Word usage .....	11
2. Normative references .....	11
3. Definitions, acronyms, and abbreviations .....	12
3.1 Definitions .....	12
3.2 Acronyms and abbreviations .....	12
4. Technical framework .....	12
4.1 Architecture framework .....	12
4.2 Layered functionalities .....	13
5. Functional components .....	14
5.1 Fundamental layer .....	14
5.2 Platform layer .....	15
5.3 Application layer .....	17
5.4 Service layer .....	18
5.5 Cross-layer functions .....	19
6. Reference process of secure computing .....	20
7. Technical and security requirements .....	21
7.1 Isolation requirements .....	21
7.2 Interoperability requirements .....	22
7.3 Performance requirements .....	22
7.4 Availability requirements .....	23
7.5 Data security requirements .....	23
7.6 Cryptography requirements .....	23
Annex A (informative) Introduction of popular TEEs .....	25
Annex B (informative) Introduction of popular TEE LibOS .....	26
Annex C (informative) Bibliography .....	27

# IEEE Standard for Secure Computing Based on Trusted Execution Environment

## 1. Overview

### 1.1 Scope

This standard specifies a framework of TEE-based secure computing systems, and defines the corresponding requirements in terms of isolation, confidentiality, compatibility, performance, usability, and security. It is applicable to guide the design, development, testing, and maintenance of TEE-based secure computing systems.

### 1.2 Word usage

The word *shall* indicates mandatory requirements strictly to be followed in order to conform to the standard and from which no deviation is permitted (*shall* equals *is required to*).<sup>6,7</sup>

The word *should* indicates that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required (*should* equals *is recommended that*).

The word *may* is used to indicate a course of action permissible within the limits of the standard (*may* equals *is permitted to*).

The word *can* is used for statements of possibility and capability, whether material, physical, or causal (*can* equals *is able to*).

## 2. Normative references

The following referenced documents are indispensable for the application of this document (i.e., they must be understood and used, so each referenced document is cited in text and its relationship to this document is explained). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

There are no normative references in this standard.

<sup>6</sup>The use of the word *must* is deprecated and cannot be used when stating mandatory requirements; *must* is used only to describe unavoidable situations.

<sup>7</sup>The use of *will* is deprecated and cannot be used when stating mandatory requirements; *will* is only used in statements of fact.

### 3. Definitions, acronyms, and abbreviations

#### 3.1 Definitions

For the purposes of this document, the following terms and definitions apply. The *IEEE Standards Dictionary Online* should be consulted for terms not defined in this clause.<sup>8</sup>

**enclave:** A private memory area that is isolated from other memory areas by using a set of hardware and software mechanisms. The contents in an enclave are protected and unable to be either read or tampered by anything outside the enclave itself.

**remote attestation:** The activity of making a claim about properties of a target by supplying evidence to a verifier over a network.

**root of trust:** Highly reliable hardware, firmware, and software components that perform specific, critical security functions. Because root of trust is inherently trusted, they must be secure by first design. As such, many roots of trust are implemented in hardware so that malware cannot tamper with the functions they provide. Root of trust provides a firm foundation from which to build security and trust.

**trusted application:** Either a complete stand-alone application or partitioned application component that runs in a trusted execution environment and should satisfy specified management methods.

**trusted execution environment:** A secure area that runs in parallel with, but isolated from, the normal environment and that helps ensure that the code and data running in it are protected with respect to privacy and integrity. Sometimes, the term TEE is used interchangeably with the term enclave, although TEE is more often used to denote the technology, while enclave is used to denote concrete TEE instances.

#### 3.2 Acronyms and abbreviations

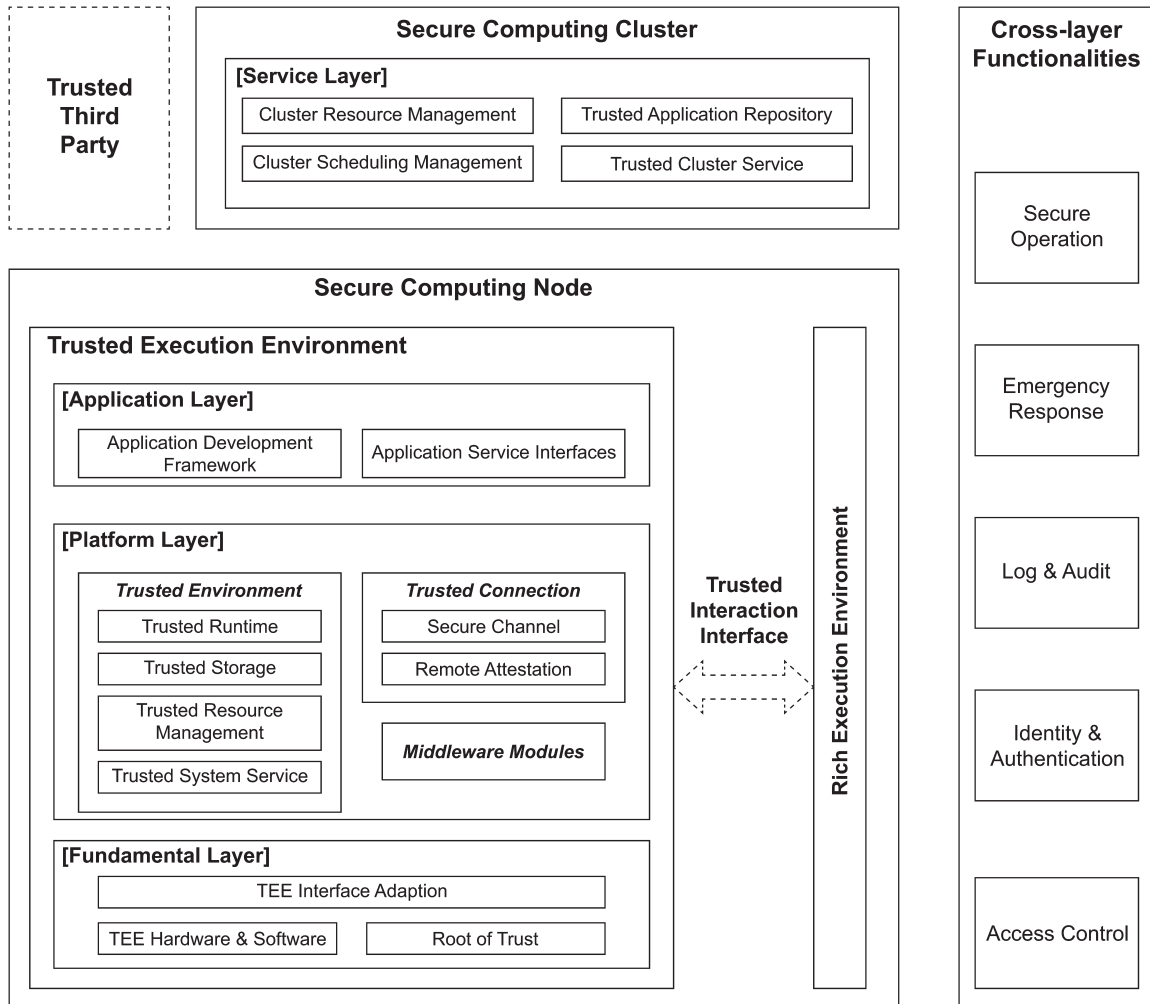
LibOS	library operating system
RA	remote attestation
REE	rich execution environment
ROT	root of trust
TA	trusted application
TEE	trusted execution environment
TEEOS	trusted execution environment operating system
TTP	trusted third party

### 4. Technical framework

#### 4.1 Architecture framework

The architecture framework of the secure computing system based on trusted execution environment (TEE) is described in [Figure 1](#).

<sup>8</sup>*IEEE Standards Dictionary Online* is available at: <http://dictionary.ieee.org>. An IEEE Account is required for access to the dictionary, and one can be created at no charge on the dictionary sign-in page.



**Figure 1—TEE-based secure computing system architecture framework**

The TEE-based secure computing system is a system to protect the confidentiality and integrity of data and code. It consists of fundamental layer, platform layer, application layer, service layer, trusted third party, and cross-layer functionalities. The TEE-based secure computing system usually can be used independently or jointly with other technologies, such as secure multi-party computing, federated learning, and blockchain.

The core component of the TEE-based secure computing system is the secure computing node. Within this node, there are trusted execution environment and untrusted rich execution environment. These two execution environments communicate with each other through the trusted interaction interface.

In most use cases, a distributed secure computing cluster is required to provide high availability.

## 4.2 Layered functionalities

Each layer cooperates with other layers to realize the full stack secure computing as follows:

- a) The fundamental layer provides the basic hardware trustworthiness. The root of trust serves as the trustworthy source of the entire TEE. The TEE hardware and software provide the function of isolation, memory encryption, and other underlying capabilities such as remote attestation and so on.

The TEE compatibility interface is to provide abstract and unified interface for the platform layer on different TEE implementations.

- b) The platform layer provides trusted application runtime environment. The trusted runtime environment can be based on the TEE OS, library operating system (LibOS), or software development kit. The trusted connection mainly contains remote attestation and secure channel modules, which can be used to establish secure communication between TEE nodes and other TEE or non-TEE nodes. The middleware modules mainly provide trusted functions to simplify and standardize the development of trusted applications.
- c) The application layer is optional, and provides external service capabilities. The development framework simplifies the application development process. The application service interfaces define how one application interacts with the other applications.
- d) The service layer provides basic cluster service capabilities for distributed computing nodes, among which cluster scheduling management is responsible for application scheduling within the cluster nodes, while cluster resource management is about the management of cluster resources (especially enclave memory of each node). The trusted application repository is mainly used to save and manage trusted applications and trusted cluster services provide auxiliary services for trusted applications.
- e) Possible involvement of the trusted third party includes trusted platform provision, trusted application review and signing, and trusted audit.
- f) The cross-layer functionalities include secure operation, emergency response, audit logging, identity authentication, access control, and data security. It can be integrated into each secure computing node, or used as an independent service to centrally manage multiple computing nodes.

## 5. Functional components

### 5.1 Fundamental layer

#### 5.1.1 Root of trust

Root of trust provides the trustworthy source of the entire TEE. Usually it is a hardware-protected secret saved inside the CPU or other trusted hardware components. The root of trust provides the foundation of trustworthiness and is used to derive various keys to protect the TEE memory and data security during the computing task execution.

The root of trust should have three main properties as follows:

- a) Confidentiality: The root of trust shall be implanted into a specified hardware region, and the hardware must enforce that anyone cannot steal the root of trust.
- b) Integrity: It shall ensure that anyone cannot tamper or replace the root of trust.
- c) Verifiability: The root of trust shall represent a TEE environment. Verifiers can verify it, e.g., by verifying a report signed by the root of trust, and determine whether the report is generated from a genuine TEE.

#### 5.1.2 TEE hardware and software

The TEE hardware and software (within trusted computing base) that make up the TEE work together to provide the basic TEE functionalities, including the following:

- a) Environment isolation: TEE hardware and software shall construct multiple TEE environments and enforce strong isolation between different environments. Each TEE environment's CPU and memory

resources shall not be accessed or modified by other TEE environments or privileged system software (e.g., the operating system and the virtual machine manager).

- b) Environment measurement: TEE hardware and software shall measure the whole TEE environment, including the platform security status and the trusted application in memory. They should generate a verifiable measurement with the help of the root of trust. Verifiers can verify the measurement and use it during the remote attestation.
- c) Memory encryption: TEE hardware and software can encrypt all the TEE memory so that attackers cannot directly steal sensitive data by physically accessing the memory.
- d) Memory integrity protection: TEE hardware and software can check the integrity of encrypted memory so that attackers do not modify or replace sensitive data.
- e) Secure key generation: TEE hardware and software can provide secure key generation based on the root of trust. The TEE hardware and software enforces the trust and privacy of these keys.

### 5.1.3 TEE interface adaption

The TEE interface adaption realizes TEE abstraction and provides interface to the platform layer. The TEE abstractions are different in different implementations, including architecture level isolation, virtual machine level isolation, and process level isolation. Refer to [Annex A](#) for more information.

The common interfaces provided by TEE should include the following:

- a) TEE instance creation: APIs for creating a TEE instance and allocating the initial CPU and memory resources.
- b) TEE instance measurement: APIs for generating the verifiable measurement of a TEE instance.
- c) TEE instance invocation: APIs for invoking a TEE instance and passing arguments to the instance.
- d) TEE instance exit: APIs for a TEE instance to switch back to the non-TEE environment and pass the return value.
- e) TEE instance destroy: APIs for deleting a TEE instance, include clearing CPU status and memory.
- f) TEE resources management: APIs for managing TEE resources, e.g., resources allocating, releasing, and scheduling.

In order to be compatible with different TEEs, there may be a cross-TEE compatibility layer to help ensure the portability of applications across TEEs.

## 5.2 Platform layer

### 5.2.1 Trusted environment

Trusted environment provides the complete system for trusted applications. Trusted environment includes trusted runtime, trusted storage, trusted resource management, and trusted system service.

#### 5.2.1.1 Trusted runtime

Trusted runtime provides programming model and runtime model for trusted applications. For example, a software development kit is used for partitioned application model, LibOS/TEEOS is used for a stand-alone application model.



### 5.2.1.2 Trusted storage

The trusted storage of the secure computing system includes the following functions:

- a) Secure, stable, efficient, and highly scalable confidential data storage capacity, and guaranteed data confidentiality, integrity, and availability.

NOTE—Availability may be realized by TEE and other related modules.<sup>9</sup>

- b) General storage interface forms, such as file system, database, etc.
- c) At the end of computing task, the system completely and effectively erases the sensitive data, to help ensure that the data has been completely eliminated.

### 5.2.1.3 Trusted resource management

The trusted resource management of the secure computing system includes the following functions:

- a) The memory management of trusted applications
- b) The isolation between trusted applications
- c) The multi-threading computing capabilities to trusted applications
- d) The multi-process computing capability to trusted applications

### 5.2.1.4 Trusted system service

The trusted system service required by trusted application includes the typical operating system service, such as network, IO, dynamic libraries management, timer, and so on.

- a) For SDK-type trusted runtime, the trusted system service is provided by the untrusted operating system.
- b) For LibOS-type trusted runtime, the trusted system service is provided by LibOS together with untrusted operating system.
- c) For TEEOS-type trusted runtime, the trusted system service is provided by the TEEOS itself in the TEE.

## 5.2.2 Trusted connection

Trusted connection is used to establish a secure communication channel between trusted application and applications outside the given TEE.

### 5.2.2.1 Remote attestation

The remote attestation module includes the following functions:

- a) Attest the state and measurement of the trusted application
- b) Attest the trustworthy evidence (such as state and measurement) of the secure computing system

NOTE—In some case, local attestation is used instead of remote attestation within a single secure computing system.

<sup>9</sup>Notes in text, tables, and figures of a standard are given for information only and do not contain requirements needed to implement this standard.

### 5.2.2.2 Secure channel

The secure channel module includes the following functions:

- a) The communication channels for data transmission from non-TEE to TEE, based on one-way remote attestation.
- b) The communication channels for data transmission between different TEE instances, based on two-way remote attestation.
- c) The confidentiality protection and integrity protection of the data transmitted through the channel.

### 5.2.3 Middleware modules

The middleware modules consist of a series of commonly used trusted software development modules. They are usually independent components and compatible with different TEEs, such as unified attestation interface, cryptographic algorithms, etc.

## 5.3 Application layer

The application layer provides complete services based on TA to external clients. The development framework can simplify the development of trusted applications. The application service interfaces are used to provide trusted services to external clients and invoke the other related services.

### 5.3.1 General description

Trusted applications are deployed in the TEE-based secure computing physical server, as shown in Figure 2. A single server can deploy multiple trusted applications. A trusted application may or may not have a binding untrusted part in the REE. Unauthorized untrusted parts are not allowed to call unbound trusted interface in TEE. Correspondingly, the client also can connect to the trusted application directly or via untrusted interface in REE.

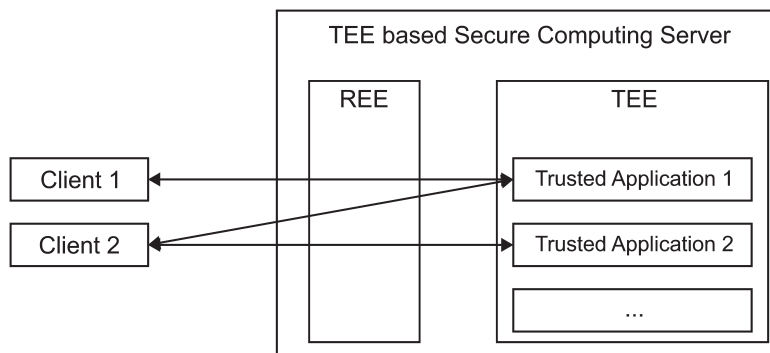


Figure 2—Overview of the trusted application

### 5.3.2 Application development framework

Application development framework creates a single unified enclaving abstraction for developer to build TEE-based trusted applications. For application development framework, the following requirements shall be met:

- a) It should provide function calls to manage the lifecycle of an enclave within the application.

- b) It should abstract secure execution mechanisms and provide APIs for defining call-ins and call-outs and the data marshalling associated with them.
- c) It should provide APIs of generating and verifying enclave measurement.
- d) It may include cryptographic libraries to provide the necessary cryptographic support within a TA.
- e) It may provide system primitives exposed by enclave runtime, such as thread and memory management.
- f) It should provide functions to support persistence of secrets.
- g) It may provide a hardware abstraction layer to make applications portable across different TEEs with no code changes.

### 5.3.3 Application service interfaces

The application service interfaces are used to expose trusted applications and invoke services from others. For application service interfaces, the following requirements shall be met:

- a) All service interfaces shall provide services based on the secure channel, and the encryption key of the secure channel should be kept inside TEE.
- b) The service interface should respond normally only after the client's identity is authenticated.
- c) If the service interface provides asynchronous computing task calls, the service interface should also provide an interface for terminating computing tasks. When certain constraints are met (for example, when most clients reach a consensus), the authenticated client can call this interface to actively stop the computing task.
- d) The service interface may adopt access control to differentiate the access rights of different clients. Access control may include a control policy.
- e) The control policy of access may satisfy the principle of least privilege and least leakage, and a multi-level security strategy could be established.

## 5.4 Service layer

The service layer provides distributed secure computing capabilities. It should provide cluster resource management service, cluster scheduling management service, trusted application repository service, and trusted cluster service.

### 5.4.1 Cluster resource management

Cluster resource management shall include the following functions:

- a) It shall support the management of secure computing nodes and the allocation of trusted resources.
- b) It shall support the monitoring of secure computing nodes, trusted resources, trusted application instances, etc.

### 5.4.2 Cluster scheduling management

Cluster scheduling management shall include the following functions:

- a) It shall schedule the trusted application to secure computing nodes and management the application life cycle.
- b) It shall support role and authorization management for cluster users.

### 5.4.3 Trusted application repository

Trusted application repository shall include the following functions:

- a) It shall support the management of source code, library files, executable, or container image of trusted applications.
- b) It shall support the acquisition, modification, and submission of trusted applications.
- c) It shall support authentication and authorization management of accessing the trusted applications.

### 5.4.4 Trusted cluster service

For trusted application, some auxiliary services should be needed, including, but not limited to, the following auxiliary services:

- a) Trusted business secret key management service should be provided.
- b) Trusted application configuration management services should be provided.

## 5.5 Cross-layer functions

### 5.5.1 Secure operation

Secure operation shall include the following functions:

- a) It shall be able to monitor and manage the operating status of the system environment and components.
- b) It should be able to monitor the occupied resources condition.
- c) It shall support operation and maintenance of underlying infrastructure.
- d) It shall support system upgrading without user perception.
- e) It shall be able to monitor the execution status of secure computing tasks.
- f) It should be able to record the user's operations and send alarm messages to the system administrator in case of abnormal scenarios.

### 5.5.2 Emergency response

Emergency response shall include the following functions:

- a) It shall set up the disposal method for exceptional events.
- b) It shall support timely reporting mechanism for emergencies.
- c) It shall support abnormal problems locking through automatic monitoring and analysis, and quick repairing.

### 5.5.3 Log and audit

Log and audit shall include the following functions:

- a) The log shall record user's operation and system operation.

- b) It shall support the log audit, learn the operation status, and provide necessary information for subsequent problem investigation and evidence collection.
- c) The log shall be saved at least over six months.

#### 5.5.4 Identity and authentication

Identity and authentication shall include the following functions:

- a) All participants shall conduct identity authentication before communicating.
- b) Identity authentication shall be carried out for the access of sensitive resources and key computing links in the computing process to help ensure the legitimacy and non-repudiation of operation behavior.
- c) The identity authentication function shall support at least one authentication method, including, but not limited to, the following methods:
  - 1) Remote attestation report
  - 2) Password
  - 3) Certificate
  - 4) Token
- d) The higher security authentication shall contain either hardware-based identity authentication or two-factor authentication.

#### 5.5.5 Access control

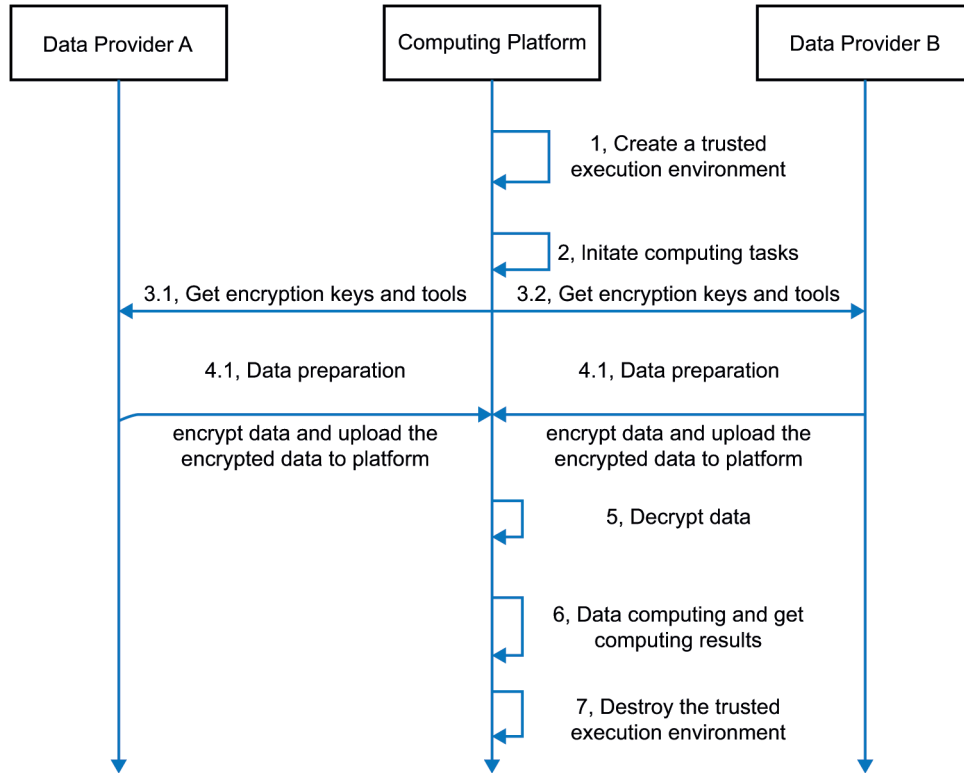
Access control shall include the following functions:

- a) It shall support role-based access control for sensitive resource and data acquisition.
- b) Role management shall be carried out, including, but not limited to, identity authentication, role addition, role deletion, role password setting, etc.

### 6. Reference process of secure computing

The procedure of secure computing is performed as shown in [Figure 3](#) and described in the following steps:

- Step 1: Create a trusted execution environment on the computing platform.
- Step 2: Initiate computing tasks on the computing platform.
- Step 3: Data providers get encryption keys and tools from the computing platform.
- Step 4: Data preparation, including data providers encrypting data and uploading encrypted data to the computing platform.
- Step 5: Decrypt data.
- Step 6: Data computing and get computing results.
- Step 7: Destroy the trusted execution environment.



**Figure 3—Processing procedure of secure computing**

## 7. Technical and security requirements

### 7.1 Isolation requirements

The isolation requirements of TEE-based secure computing system include the following:

- a) It shall isolate hardware resources into secure resources from normal hardware resources. For example, the memory to store sensitive data is isolated to help ensure that data is not leaked besides breaking TEE, so that the data within the trusted execution environment is unable to be accessed by the operating system and other highly privileged software, and by special means such as memory sniffing.
- b) It shall support the software and hardware isolation between trusted execution environment and untrusted execution environment.
- c) It shall support trusted interaction so that only the specific authorized entities can securely communicate with the isolated programs and securely update the isolated data.
- d) It shall support mutual isolation between different trusted application instances. Each trusted application instance is independent, and shall not access each other without authorization. For a partitioned application, the untrusted part should access the trusted application through strictly defined interfaces.

## 7.2 Interoperability requirements

### 7.2.1 Authentication process

TEEs cooperate to process computing tasks and need mutual authentication. The authentication process shall meet the following requirements, which are applicable to both homogeneous and heterogeneous TEEs:

- a) It should authenticate the TEE environment mutually. Authentication should be based on TEE remote attestation.
- b) It should use the challenge/response to verify the validity of TEE remote attestation report.
- c) It should establish an encrypted channel to transfer the sensitive data.
- d) It may include a session key exchange process in the mutual authentication, and then the session key is used to build the secure channel.

### 7.2.2 Verification

Remote attestation attester generates the unified format of attestation report that can be parsed and verified by others. The verification action can be either integrated into trusted application or executed through an independent attestation service. A verifier may verify the following:

- a) The secure platform state and trusted application attributes
- b) The correctness of the report format, including all required fields
- c) The validity of secure boot, including the version of basic input output system, system kernel, and system services if the report contains the measurements in the secure boot
- d) The validity of the signature signed by the remote attestation service if it's necessary

### 7.2.3 Attributes

Attributes that can be resolved by trusted application and need consistency may include the following:

- a) Code measurement
- b) Signature of the authority
- c) Product ID and software version
- d) The identity of trusted application provider
- e) Data consistency commitment
- f) Tasks initialization parameters

## 7.3 Performance requirements

The secure computing system includes the following performance requirements:

- a) It shall support quick start of secure computing kernel process
- b) It shall support low overhead communications between inter-process in secure computing kernel

## 7.4 Availability requirements

The secure computing system includes the following availability requirements:

- a) Data synchronization and persistence among multiple TEEs in one cluster shall be supported.
- b) The capability of failover and disaster recovery should be supported.
- c) The system should recover automatically including data backup and quick recovery after the failure such as the failure of server, hard disk failure, or network failure.

## 7.5 Data security requirements

The data security requirements for TEE-based secure computing system include the following:

- a) It shall help ensure that data utilization obeys aligned agreement to help avoid data abuse.
- b) It shall support data encryption methods to help prevent sensitive data from being snooped by other data providers, platform, and users.
- c) It shall support the transmission of messages only among specific authorized nodes, and data shall not be accessed by unauthorized users.
- d) It shall support secure transmission that helps ensure transferred data will not be intercepted or tampered with.
- e) It shall support secure storage of sensitive data that helps ensure confidentiality, integrity, and availability of data.
- f) It shall support data authentication methods so that the integrity of computing results can be verified.
- g) It shall support the destruction of data and TEEs.

## 7.6 Cryptography requirements

### 7.6.1 Cryptography algorithms

TEE needs to use cryptographic algorithms to realize security requirements such as secure storage, authentication, etc. To realize these functions, the cryptography algorithm requirements are included as follows:

- a) Symmetric cryptography algorithm shall be supported for data encryption/decryption function.
- b) Message authentication code function shall be supported, to store data in ciphertext, and protect data integrity.
- c) Message digest algorithm (hash algorithm) function shall be supported. It is used to hide data information and may be an important component of other cryptography function.
- d) Asymmetric cryptography algorithm shall be supported to realize data encryption/decryption function and signature/verification function. Asymmetric cryptography algorithm is used to realize identity authentication, share keys of symmetric cryptography, or digital signature.
- e) Key generation function and key exchange function shall be supported.
- f) Random number generating function shall be supported. Random number generating function is used to generate pseudorandom number, which may be used in other cryptography functions.



- g) The post-quantum secure cryptographic algorithms may be supported, e.g., symmetric algorithms with double key length, lattice-based, code-based, hash-based, multivariate, and super singular elliptic curve isogeny cryptography.

### 7.6.2 Cryptography security requirements

The cryptography security requirements are included as follows:

- a) Cryptography algorithm functions should be called through a secure interface or in a secure environment.
- b) The security of the encryption algorithm shall be reviewed regularly, and if necessary, the encryption algorithm with higher cracking difficulty shall be adopted.
- c) It shall have a clear key management scheme and authority control, manage the life cycle of the key, help ensure the security of key generation, as well as storage, distribution, and use of the key.
- d) The digital certificate used shall meet the standard format requirements, the digital certificate shall be within the validity period, and the certificate chain shall be complete and valid.

## Annex A

(informative)

### Introduction of popular TEEs

According to the isolation implementation, TEEs are divided into three categories: architecture level isolation, virtual machine level isolation, and application-level isolation.

For example, an application-level TEE uses virtualization technology to provide application-level isolation, and has the following features:

- a) Verifiable security: The core implementation has been formally verified. The code has been reviewed and certified by authoritative organizations, and it also has financial technology product certification.
- b) Advanced TEE capabilities: Support all TEE capabilities such as isolated execution, remote attestation, memory encryption, and data sealing.
- c) Open software ecosystem: Compatible with a wide variety of existing TEE ecosystems.
- d) Heterogeneous TEE compatibility: Provide heterogeneous TEE interconnection capability.
- e) Support clustered usage: Supporting scalable and highly available cluster capabilities.

## Annex B

(informative)

### Introduction of popular TEE LibOS

Generally, TEE has limitation to directly use operating system services, such as network service and file system service. With the help of library operating system (LibOS), the legacy application can run inside TEE with little or even no modifications, and realize protection of user workloads. For example, a memory-safe, multi-process LibOS has the following salient features:

- a) Efficient multitasking. It offers light-weight LibOS processes that enables all LibOS processes to share the same enclave. Compared to heavy-weight item, per-enclave LibOS processes, light-weight LibOS processes is up to  $1000 \times$  faster on startup and three  $\times$  faster on IPC. In addition, it offers an optional multi-domain software fault isolation scheme to isolate the LibOS processes if needed.<sup>10</sup>
- b) Multiple file system support: It supports various types of file systems, e.g., read-only hashed file systems (for integrity protection), writable encrypted file systems (for confidentiality protection) and untrusted host file systems (for convenient data exchange between the LibOS and the host OS).
- c) Memory safety: It is the first LibOS written in a memory-safe programming language. Thus, it's much less likely to contain low-level, memory-safety bugs and is more trustworthy to host security-critical applications.
- d) Ease-of-use: It provides user-friendly build and command-line tools. Running applications on TEE can be as simple as only typing several shell commands.

---

<sup>10</sup>Available at: <http://www.cse.psu.edu/~gxt29/papers/sfi-final.pdf>.

## Annex C

(informative)






### Bibliography

Bibliographical references are resources that provide additional or helpful material but do not need to be understood or used to implement this standard. Reference to these resources is made for informational use only.

[B1] Coker, G., J. Guttman, P. Loscocco, et al., “Principles of Remote Attestation,” *International Journal of Information Security*, vol. 10, no. 201, pp. 63–81, June 2011.

# RAISING THE WORLD'S STANDARDS

## Connect with us on:

-  **Twitter:** [twitter.com/ieeesa](https://twitter.com/ieeesa)
-  **Facebook:** [facebook.com/ieeesa](https://facebook.com/ieeesa)
-  **LinkedIn:** [linkedin.com/groups/1791118](https://linkedin.com/groups/1791118)
-  **Beyond Standards blog:** [beyondstandards.ieee.org](https://beyondstandards.ieee.org)
-  **YouTube:** [youtube.com/ieeesa](https://youtube.com/ieeesa)

[standards.ieee.org](https://standards.ieee.org)  
Phone: +1 732 981 0060