EMUI 11.0 Security Technical White Paper

Issue 1.0

Date 2020-11-30

HUAWEI

Contents

Note: * indicates a feature not supported by all devices. Supported features vary depending on device models or market characteristics in difference countries. For more information, refer to specific product descriptions.

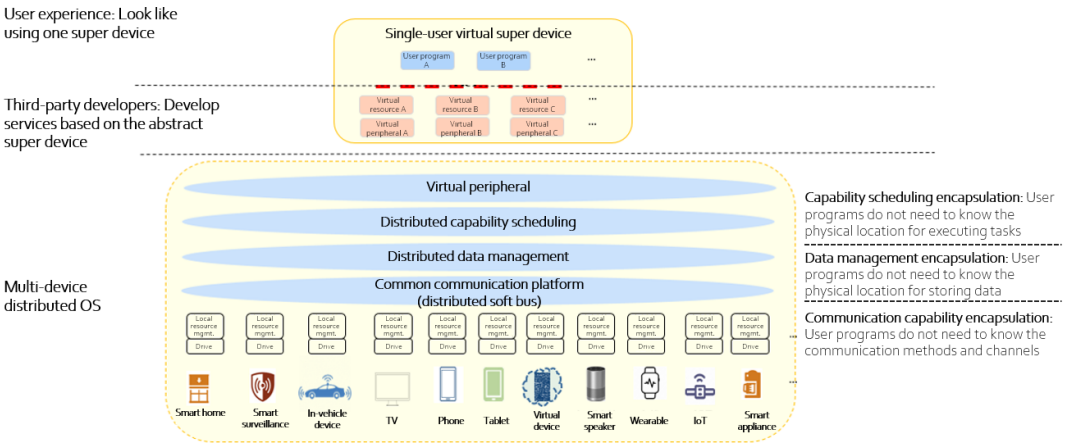
Figures

1 Overview

Introduction

The Emotion UI (EMUI) is an operating system developed by Huawei for smart mobile products. It adheres to the "design with empathy" philosophy, featuring a simplified design with a unified language and consistent operations.

EMUI 11.0 is the next step in human-device interaction, extending user experience from a single device to multiple devices and securely connecting the multiple devices. It provides a unified, distributed cross-device development platform, which seamlessly integrates multiple smart devices available in the distributed, all-scenario smart life, allowing these smart devices to transform into one super device for consumers. This systematically improves experience for consumers and working efficiency for developers in multi-device environments.



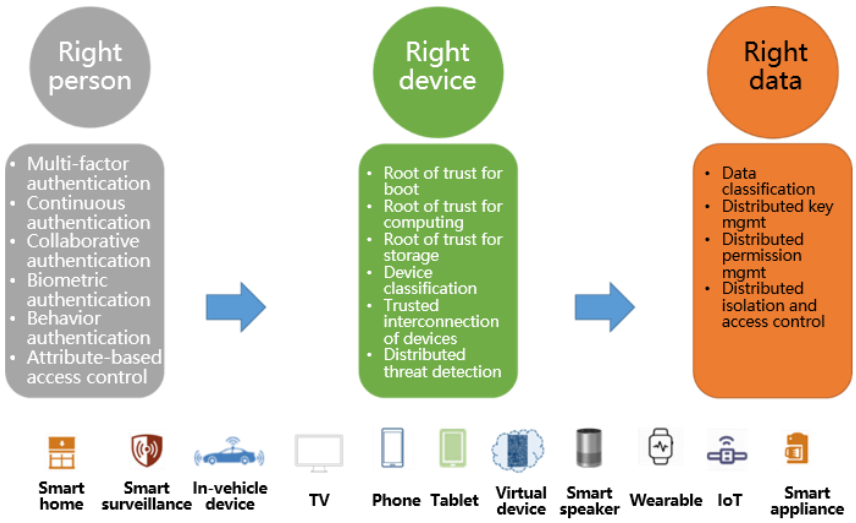
Huawei's distributed, all-scenario smart life solution benefits both developers and consumers.

- Benefits to developers:
  - **Unified development platform**: The hardware differences are shielded so that different devices can transform into a super device, saving developers from repeatedly developing apps for multiple devices.
- Benefits to consumers:
  - **Hardware collaboration**: Multiple types of smart devices can work together and share hardware resources, providing high-quality cross-device experience.

- **Ecosystem sharing**: The software service can be provided by the device at the most appropriate distance, delivering the most suitable capabilities, and allowing the most convenient interaction with users. Consumers can enjoy seamless experience without needing to know which device is providing the software service; as long as the software service is installed on one device, it can be used on multiple devices.

EMUI Security

In terms of device security, Huawei always adheres to the principle of allowing the right person to access the right data through the right devices, thereby providing consistent security assurance for data throughout the lifecycle.



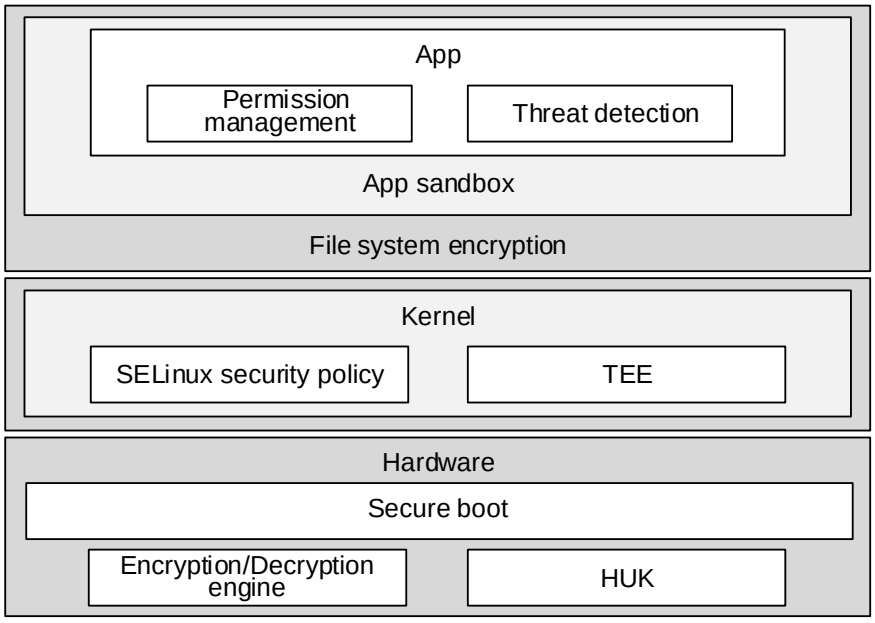
Security is a systematic project. EMUI provides end-to-end security protection from hardware, systems, apps, and the cloud (as shown in Figure 1-1), including security and privacy protection for the hardware chips, Trusted Execution Environment (TEE), system kernel, data, apps, network, payment, cloud services, and device management.

EMUI provides a secure boot mechanism from underlying hardware chips to prevent the EMUI read-only memory (ROM) image from being tampered with. The ROM image can only run on a device after passing signature verification. This ensures secure boot for the bootloader, recovery, and kernel images, and prevents tampering and malicious code implantation by attackers during the boot process, thereby ensuring security from hardware chips to EMUI system boot.

To ensure data security, EMUI encrypts user data using a hardware unique key (HUK) and a user lock screen password. Data files from various apps are stored in the sandboxes of the corresponding apps, preventing files from one app from being accessed by another. The data erasure function is provided to permanently erase data during device recycling or factory restoration, thereby preventing unauthorized data restoration. EMUI also allows cloud services to help users back up and synchronize data to ensure data security.

For app security, in addition to mechanisms such as security sandbox and permission management, EMUI pre-installs Phone Manager to provide virus scanning, block and filter, traffic management, notification management, and other functions. Utilizing these functions, EMUI can automatically detect viruses and Trojans within apps, and provide fine-grained permission, traffic, and notification management functions.

Figure 1-1 EMUI security architecture



As mobile Internet continues to develop, smart mobile devices have become primary network access devices, which store large amounts of user data including personal user information. In addition, an increasing number of apps from unverifiable sources are installed on these devices where malicious apps may infringe upon user privacy or steal user assets. As a result, privacy issues are becoming more prominent and arise as major concerns of consumers.

Huawei always attaches great importance to product security and user privacy protection.

In smart interconnection scenarios of the distributed, all-scenario smart life solution, Huawei products are facing new security challenges:

1. How to ensure that each node of the distributed device is secure and trusted, as well as working properly
2. How to interconnect and transform different user devices into one distributed virtual device in a secure and trusted manner while allowing these devices to identify/recognize the same user
3. How to ensure consistent security protection for the user data storage and use on each node and data transmission between nodes of the distributed device

This document describes the security technologies and functions of the EMUI 11.0 system, and enables security practitioners to understand the specific implementation of EMUI security. It also enables EMUI developers to integrate the security capabilities provided by the EMUI platform with developer programs to ensure the privacy and security of consumer data.

This document contains the following chapters:

- Hardware security: secure boot, hardware encryption/decryption engine and random number generator (RNG), HUK, device group key, device attestation, secure element, and secure storage
- TEE: secure OS, security capability, security capability openness, and TUI
- System security: integrity protection covering verified boot, Huawei Kernel Integrity Protection (HKIP), and EMUI Integrity Measurement Architecture (EIMA); kernel

security covering system access control and kernel address space layout randomization (KASLR); identity authentication; system software update

- Data security: lock screen password protection, secure storage of short data, Huawei Universal Keystore (HUKS), secure erasure, and password vault
- App security: app release security detection, app signature, app sandbox, runtime memory protection, secure input, app threat detection, artificial intelligence (AI) security protection, and malicious website detection
- Network and communication security: virtual private network (VPN), Transport Layer Security (TLS), Wi-Fi security, protection against fake towers, and device interconnection security
- Trusted interconnection of the distributed system: interconnection security between devices logged-in with the same HUAWEI ID; IoT device interconnection security
- Collaborative user identity authentication of the distributed system: multi-device collaborative authentication of facial information
- Payment security: Huawei Pay, secure keys, and short message service (SMS) verification code protection
- Internet cloud service security: HUAWEI ID, account protection, HUAWEI ID message, MyCloud, HUAWEI ID-based key, and MyCloud backup
- Device management: Find My Phone, activation lock, and Mobile Device Management (MDM) Application Programming Interface (API)
- Privacy protection: permission management, audio/video recording reminder, location access, device identifier system, differential privacy, and privacy statement

EMUI is applied to products running a variety of hardware chip platforms. As such, security implementation may differ depending on hardware and chips. For the specifications relating to a particular device, refer to its product manual.

2 Hardware Security

EMUI adopts security capabilities based on hardware chips, and delivers overall security with secure software solutions. Hardware chip security is the core of the EMUI security system. This chapter describes Huawei device hardware chip security, including the following security features:

- Secure boot
- Hardware encryption/decryption engine and RNG
- HUK
- Device group key
- Device attestation
- Secure element
- Secure storage

Secure Boot

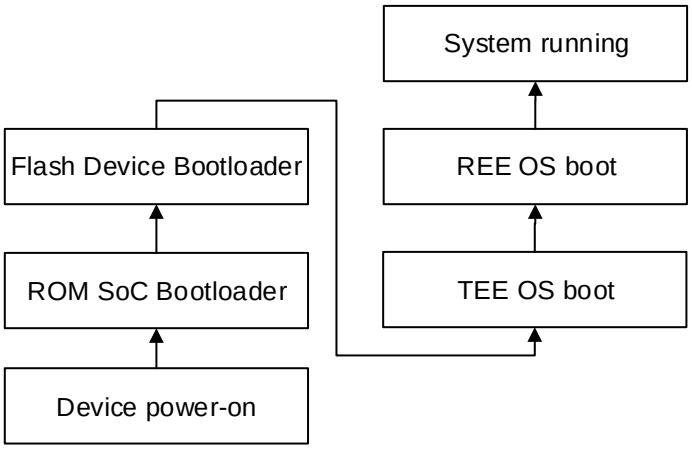
Secure boot prevents the loading and running of unauthorized software during device boot. The boot program uses a public key to verify the digital signatures of software, ensuring trustworthiness and integrity. Only image files that pass the signature verification can be loaded. These files include bootloader, kernel, and baseband firmware image files. If the signature verification fails during boot, the boot process is terminated.

When a device is started, a boot program in the chip, known as the ROM SoC Bootloader, is executed first. This code snippet is written into the ROM inside the chip during manufacturing and is not modifiable after delivery. It is the root of trust for device boot.

The ROM SoC Bootloader performs basic system initialization and then loads the Flash Device Bootloader from the flash storage chip. The ROM SoC Bootloader uses the public key hash in the eFuse space (using the fuse technique and cannot be changed once the fuse blows) of the main chip to verify the public key, and then uses the public key to verify the digital signature of the Flash Device Bootloader image. The Flash Device Bootloader is executed once verification is successful. The Flash Device Bootloader then loads, verifies, and executes the next image file. A similar process is repeated until the entire system is booted, eventually establishing a chain of trust, and thereby preventing unauthorized programs from being loaded during the boot process.

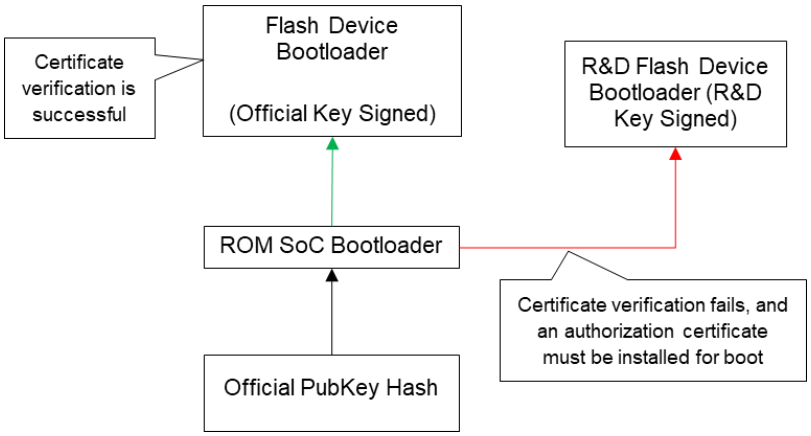
The images used during some boot processes are encrypted.

Figure 2-1 Secure boot



EMUI 11.0 secure boot supports the isolation between commercial and R&D signature keys. Commercial software editions use official commercial signature keys for signing and verification. In the meanwhile, R&D software editions use R&D signature keys for signing, and such editions cannot be booted on commercial mobile phones, thereby preventing any impact.

Figure 2-2 Separation of commercial and R&D signature keys



Hardware Encryption/Decryption Engine and RNG

To meet the requirements of high-performance encryption/decryption and key protection, EMUI utilizes the hardware security engine to perform operations such as data encryption/decryption and key derivation. The chip provides a high-performance hardware encryption/decryption acceleration engine which supports the following algorithms and functions:

- 3DES
- AES128 and AES256
- SHA1 and SHA256
- HMAC-SHA1 and HMAC-SHA256

- RSA1024 and RSA2048
- ECDSA-P256 and ECDH-P256
- CTR_DRBG RNG compliant with NIST SP800-90A and hardware entropy source compliant with NIST SP800-90B

HUK

An HUK is a unique identifier in a chip. It can only be used by the hardware encryption/decryption engine for key derivation and varies depending on the chip. The HUK provides a device-unique key for EMUI. It is applied to lock screen password protection, file system encryption, and other functions.

Device Group Key

A device group key is an identifier in a chip. It can only be used by the hardware encryption/decryption engine for key derivation and is the same across devices of the same type. The device group key enables EMUI to derive the same key for the same type of devices. It is applied to image encryption and other functions.

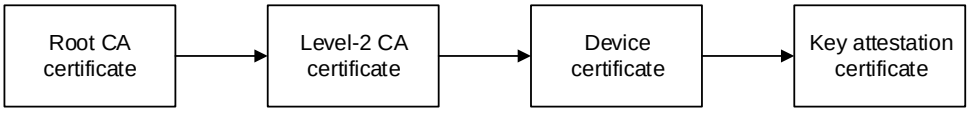
Device Attestation

To ensure that EMUI devices are trustworthy, Huawei has preset device certificates and public and private key pairs in the production line. The device certificate and public and private key pair vary by device and are used to uniquely identify a specific device. Certificates and keys are written into iTrustee of EMUI and then encrypted for storage. Services cannot directly access the certificates or keys, and can only access them through the unique proprietary interface provided by Huawei's unified key management service.

A device certificate is issued by the Huawei public key infrastructure (PKI) system and contains a three-level certificate chain.

If the validity of devices, users, or accounts needs to be verified for services with high security requirements (such as payment and account services), a service certificate can be obtained from the device certificate and provided for the service entity to verify the certificate chain before the services can be executed, ensuring that only trusted devices are allowed to operate corresponding services.

A service certificate is obtained from a device certificate in iTrustee of a mobile phone. A service certificate contains a four-level certificate chain, as shown in the following figure. After the four-level certificate chain passes verification and the signature of the last level of certificate passes verification, the device is considered valid and is allowed to perform the corresponding service.



Secure Element*

A secure element is a subsystem that provides a secure execution and storage environment. On EMUI, a secure element is used to address insufficient mobile payment security.

Huawei developed the Integrated Secure Element (inSE) security solution, which integrates security chips into processors. When compared with software security solutions and other separated chip security solutions, the inSE provides both software and hardware protection through System-on-a-Chip (SoC) level security design and software algorithms. This solution not only delivers software security protection capabilities, but also defends against physical attacks. It provides improved protection and fundamentally ensures the security of mobile phones.

The inSE has received the China Financial National Rising Authentication (CFNR) Technology Certification of Mobile Financial Service – Chip Security, China UnionPay's Certification of Card Chip Security Specifications, and a Certificate for Commercial Cipher Product Models. In addition, the inSE has obtained the EMVCo chip security certification and can be used for international mobile payment and mobile financial services.

*Note: This function is available only for certain chip models in China.

Secure Storage*

The secure storage function is a security function implemented by the secure file system (SFS) provided by iTrustee®. This function enables the secure storage of keys, certificates, personal privacy data, fingerprint templates, and more.

A trusted application (TA) running in iTrustee® uses a secure storage API to encrypt and store data in the SFS. The encrypted data is accessible only to the TA.

The AES256 hardware encryption/decryption used by the secure storage function is compatible with the GlobalPlatform (GP) TEE standard. Secure storage keys are derived by the HUK and not sent outside of the TrustZone. Data encrypted using the keys cannot be decrypted outside of the TrustZone.

EMUI also provides a flash-based replay protected memory block (RPMB) to prevent system data from unauthorized deletion and access. The RPMB is directly managed by iTrustee and bound with the keys derived by the HUK. Only iTrustee can access the RPMB-protected data, and the Rich Execution Environment (REE) does not provide any interface for accessing the RPMB. The RPMB uses built-in counters, keys, and the HMAC verification mechanism to defend against replay attacks and prevent data from being maliciously overwritten or tampered with.

*Note: This function is available only for certain chip models in China.

Mobile Security Processor*

A mobile security processor (MSP) is a highly secure operating environment built into Kirin chips. EMUI uses MSPs in TEE to implement services such as lock screen password protection and verification, file encryption, biometric feature protection and identification, and key management. MSPs guarantee basic EMUI security capabilities through hardware.

*Note: This function is available only for certain chip models in China.

3 TEE

This chapter describes the TEE of devices. The Huawei iTrustee is a secure OS that provides a TEE in compliance with GP TEE specifications. It is independently developed by Huawei based on the HarmonyOS's formal microkernel, and features high security, performance, scalability, and stability.

iTrustee Secure OS

The iTrustee secure OS provides a TEE based on TrustZone technology. TrustZone enables hardware-level security and balances performance, security, and cost. This technology allows CPUs to operate in a TEE or an REE. Special instructions are used to switch a CPU between the TEE and REE, in order to provide hardware isolation. A TEE protects and isolates hardware resources, such as memory and peripherals. End-to-end security is achieved by protecting the execution process, key confidentiality, data integrity, and access permissions, which prevents malware attacks from an REE.

Microkernel

The Huawei iTrustee secure OS utilizes microkernel technology, which simplifies kernel functions and adopts a modular design to implement more system services outside the kernel. The microkernel provides only the most basic services, with system services remaining in user mode for most of the time. On-demand scaling improves system performance and reduces the attack surface. Fine-grained permission design is enhanced, allowing the iTrustee secure OS to have the following advantages:

- Good scalability: A unified security kernel is built for distributed devices, allowing heterogeneous devices to support various services, such as multi-core, on-demand concurrency, and large- and small-core scheduling.
- Easy to implement and debug: Stable underlying library interfaces facilitate application development and porting, as well as supporting the development of the security service ecosystem.

Formal Verification

iTrustee is the first of its kind to use formal verification, significantly improving the system security level of the TEE kernel, and thus rebuilding trustworthiness and security. Formal verification uses mathematical theorems to verify system correctness (without vulnerabilities) from the source. Conventional verification methods (such as function verification and attack simulation) apply only to limited scenarios, while formal verification can use data models to verify all software running paths. This process verifies the correctness of core modules, core

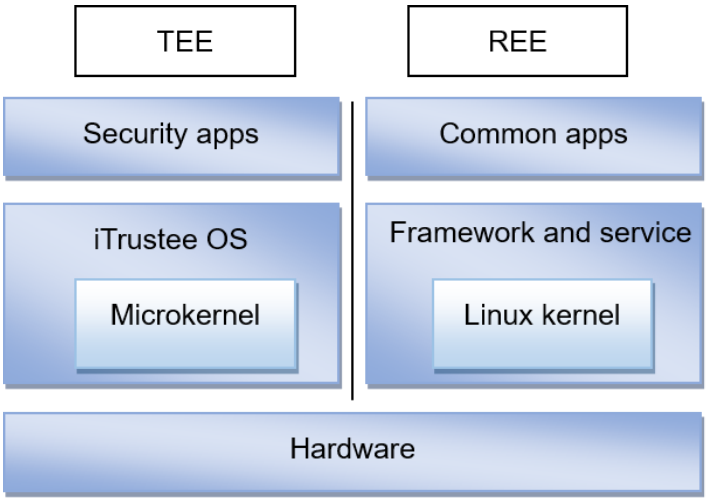
APIs, and high-level mechanisms, such as process isolation and permission management, preventing data race and memory access errors.

The iTrustee secure OS microkernel has obtained the CC EAL5+ certification (the highest security level), thanks to its use of formal verification. iTrustee is constantly working toward a TEE without vulnerabilities to provide higher security assurance for products.

In addition, iTrustee implements comprehensive security hardening for REE-side systems, channels, and authentication as well as TEE-side systems, and uses kill-chain-based security defense techniques to enhance system security, such as image anti-reverse engineering, system anti-intrusion, and data anti-damage. For example, anti-reverse engineering is used to prevent attacks in the intrusion preparation phase by encrypting images. Image encryption is enabled in the chip delivery phase to prevent reverse engineering attacks. Anti-intrusion encrypts authentication information and strictly authenticates REE-TEE communication sessions to ensure that TEE data from the REE is intact and trusted. Anti-attack uses control flow protection, stack canary, and other techniques to defend against common kernel vulnerability exploits.

iTrustee also builds proactive defense capabilities to identify abnormal program behavior and REE-side system exceptions, enabling security responses to be made in advance and protecting sensitive information.

The following figure shows the iTrustee secure OS architecture.



The iTrustee secure OS ensures the safe running of security apps by providing a TEE, thereby safeguarding security services. Major security services are as follows:

Content protection	Applies to the digital rights management (DRM) field to ensure security and anti-copy during playback.
Bring your own device (BYOD)	Applies when enterprises require mobile platforms with higher security. For example, secure storage is required to store user login passwords. iTrustee ensures the security of user login passwords and prevents malicious programs from stealing user passwords.

Mobile payment	Ensures the security of input information and can be used together with Near Field Communication (NFC). iTrustee protects user input information against theft from malicious programs.
Application logic protection	Protects critical application logic from being stolen or tampered with.

As an example, iTrustee provides a TEE to protect the security of services on Huawei mobile phones, such as fingerprint payment, 3D face recognition, Huawei Pay, mobile POS, smart vehicle key, secure key, SkyTone, and electronic identification (eID).

The iTrustee secure OS supports the following basic security capabilities:

Trusted Storage Service

Trusted storage service enables the storing of critical information and ensures data confidentiality and integrity. Trusted storage supports device binding and isolation between different security apps. Each security app can only access its own storage content and cannot open, delete, or tamper with the storage content of other apps.

Trusted storage of iTrustee is classified into two types: SFS and RPMB. An SFS stores ciphertext to a specific secure storage partition, and an RPMB stores ciphertext to a specific storage area of the embedded multimedia card (eMMC). The RPMB supports anti-deletion and anti-rollback.

Encryption/Decryption Service

iTrustee supports multiple symmetric and asymmetric encryption and decryption algorithms, as well as key derivation algorithms. It supports the same key derived on a chip platform, HUK, keys derived of hardware based on secure elements, and international standard cryptographic algorithms. It also provides support for third-party development of service TAs that store and use keys, and complies with GP TEE specifications. To improve security, key generation and calculation in iTrustee is implemented by independent hardware chips. Keys are stored in a separate secure storage chip or in a secure storage space that is strictly encrypted. Users can develop TAs based on service needs to use the trusted key service.

Trusted Display and Input (TUI)

In app environments in the REE, the displayed payment amounts or input passwords may be hijacked by malicious apps. To counter such threats, iTrustee provides the Trusted UI (TUI) display technology (compliant with GP standards) that disables screenshots to protect content displayed by TAs, and prohibits access from the REE side. In this way, the TUI prevents the hijacking and tampering of displayed data and input by malicious apps, so that such apps cannot view information on the screen or access the touchscreen.

The TUI ensures that the information displayed to users is not intercepted, modified, or obstructed by any software in the REE or unauthorized apps in the TEE. Displayed information is not transferred to the REE, and permission control is used to ensure that only authorized TEE apps can access the information. In the TUI, preset images or texts are displayed to indicate the secure display and input state.

The TUI supports basic controls such as PNG images, texts, buttons, and text entry boxes, display of Chinese characters, English letters, symbols, and digits in the same size,

customized UI, randomized keypad keys, and various controls and window management. In addition, the UI is consistent in style with EMUI of devices.

Trusted Time

iTrustee provides trusted reference time, which cannot be modified by malicious TA or REE apps.

iTrustee supports the following advanced security capabilities:

- **Multi-core and multi-thread capabilities:** Multiple tasks can be created for security services and run on multiple CPUs, greatly improving the computing power of iTrustee. For example, the 3D face TA utilizes the multi-thread architecture and can run concurrently on multiple CPUs, ensuring the security of 3D face recognition throughout the process. Facial data, facial detection, facial data storage, 3D face recognition algorithm, and facial attribute extraction and comparison are all within the TrustZone.
- **Basic function library and math library:** Standard C libraries are supported, which provide approximate Portable Operating System Interface (POSIX) APIs.
- **AI capability:** The neural-network processing unit (NPU) library is integrated, which provides the convolutional neural network (CNN) computing power. 3D face recognition uses the NPU capability in iTrustee.
- **Device security service:** Unique device identifiers are provided, as well as REE health status information, and more.

iTrustee provides developers with TEE platform capabilities, diversified APIs, comprehensive SDKs, and relevant reference manuals and reference design. It also provides security certificate management, app signature, security app lifecycle management, and application release services. A unified developer UI is available in the HUAWEI DevEco Studio development environment.

Third-party TAs can be developed and debugged in iTrustee.

4 System Security

System security aims to ensure that EMUI devices leverage the security capabilities of hardware chips to provide basic hardware-based software security capabilities for apps running on the EMUI system. EMUI builds system security capabilities primarily from the following aspects:

- Integrity protection: This is the basis of system security, ensuring that trusted system software provided by vendors is running at initial device operation. In addition, Huawei Kernel Integrity Protection (HKIP) and EMUI Integrity Measurement Architecture (EIMA) are used to ensure that the kernel is not maliciously compromised during runtime and that any compromised system is promptly detected.
- System software update: When the system becomes faulty or maliciously compromised, the minimum system can be used to perform security update of the system software. Only authentic system software can be used for device updates.
- Kernel vulnerability anti-exploitation: At runtime, the system faces the risk of malicious exploitation of kernel vulnerabilities. If the kernel is compromised, the system cannot provide basic protection for upper-layer apps, and confidential data of apps may be disclosed. For this reason, multiple kernel vulnerability anti-exploitation technologies are needed in different scenarios. For example, Kernel Address Space Layout Randomization (KASLR) can ensure that vulnerabilities are not discovered at the kernel's runtime. Even if vulnerabilities are discovered, exploitation can be prevented using Privileged Access Never (PAN)/Privileged eXecute Never (PXN) and Control Flow Integrity (CFI).
- Kernel attack detection: This technology can accurately detect exploitation of kernel vulnerabilities and malicious attacks on systems and promptly notify users to take proper mitigations.
- Mandatory access control (MAC): After a secure and trusted system kernel base is built using the preceding four technologies, MAC can be used for the kernel, defining policy rules for different apps in the system to properly use different resources, ensuring that the entire system provides basic security capabilities for upper-layer apps.
- Identity Authentication: EMUI provides two biometric identification capabilities: fingerprint recognition and facial recognition. That is, EMUI uses the unique physiological features (fingerprint and facial features) to authenticate personal identities. These capabilities can be applied to identity authentication scenarios such as device unlocking, payment, and app login.

Integrity Protection

Secure Boot and Verified Boot

EMUI supports secure boot and verified boot functions. Secure boot uses the signature verification mechanism, which ensures that the system image is provided by the vendor and cannot be maliciously tampered with. When a read-only system partition with verified boot enabled is accessed, the system uses the integrity protection information generated when the read-only partition image is built to verify the integrity of the accessed partition. This feature helps prevent malicious software from permanently residing in system partitions and ensures that the device has the same status at startup as when it was last used.

HKIP*

Although secure boot and verified boot ensure the authenticity and integrity of software during startup, vulnerabilities in authentic code may still be exploited by attackers. HKIP uses the hypervisor mode (EL2) provided by the ARMv8 processor to protect the kernel, preventing key system registers, page tables, and code from being tampered with. This protects system integrity and prevents privilege escalation during system runtime. HKIP protects not only static data such as code and read-only data segments, but also some dynamic data using the write-rare protection mechanism. HKIP uses this mechanism to secure kernel data that is read most of the time but rarely modified. Even if attackers exploit vulnerabilities to write the memory at the kernel level, they cannot modify the protected data.

Currently, HKIP supports the following security protection mechanisms:

- Code snippets of the kernel and driver module cannot be tampered with.
- Read-only data of the kernel and driver module cannot be tampered with.
- Non-code snippets of the kernel cannot be executed.
- Critical dynamic kernel data cannot be tampered with.
- Critical system register settings cannot be tampered with.

*Note: This function is available only for certain HiSilicon chip models in China.

EIMA

The EIMA measures and detects the integrity of critical code and resource files of the system, and provides a system integrity measurement framework. This framework offers a unified service for measuring the integrity of critical system components or processes, and addresses runtime measurement as well as dynamic measurement of user-mode processes. This detects whether protected processes have been maliciously tampered with so that handling policies can be provided. The integrity measurement framework consists of three parts:

1. Baseline extraction

   The goal of baseline extraction is to generate static baseline metrics for software programs to be protected. Target files are hashed to generate baseline metrics. Two generation modes are available:

   - Offline generation: Baseline metrics are calculated during the build process, and are protected by a private key signature and built into the software image version.
   - Runtime generation: It is assumed that secure boot can ensure the validity of files. Baseline metrics are generated when target programs are loaded for the first time.

2. Static measurement

   The integrity of a file means that its content or attributes have not been modified. From a cryptography point of view, the hash value of a file can be used to detect whether the file

has been tampered with. Therefore, the hash values of measured objects are collected to determine the integrity of programs or data instances during memory loading.

3. Runtime measurement

   In the measurement evaluation phase, the baseline metrics are compared with the measurement data collected during system operation to determine whether the programs running are consistent with the baseline metrics. The integrity check result is provided, and service-specific decision makers then determine subsequent handling policies.

System Software Update

EMUI supports over the air (OTA) update in order to quickly fix any possible vulnerabilities. The signature of an update package is verified during system software updates. Only verified update packages are considered authentic and can be installed.

In addition, the EMUI provides software update control. At the beginning of OTA update and after a software package is downloaded, EMUI applies for update authorization by sending the digest information of the device identifier, the version number and hash value of the update package, and the device upgrade token to the OTA server. The OTA server verifies the digest before authorization. If the digest verification succeeds, the OTA server signs the digest and returns it to the device. The upgrade can be implemented only after the device passes the signature verification. If the device fails the signature verification, an upgrade failure is displayed to prevent unauthorized software updates, especially updates using vulnerable software.

EMUI periodically releases security patches. After the system is upgraded, required security patches are automatically updated to ensure the security of the EMUI system. For more information about software security updates, visit https://consumer.huawei.com/en/support/bulletin/.

Kernel Vulnerability Anti-exploitation

KASLR

In a code reuse attack, a specific jump address must be determined for reused code. KASLR enables the address mapped to the kernel image to have an offset relative to the link address, and this offset address is randomly generated upon each boot. As a result, the virtual address mapped to the kernel image varies with each boot. KASLR enables unpredictable address space layout, and makes it more difficult to launch code reuse attacks, thereby enhancing the security of the system kernel.

PAN/PXN

EMUI supports ARMv8-based PAN and PXN for security protection of kernels. These technologies prevent the kernel from accessing user space data and executing user space code.

Using some kernel attack methods, an attacker tampers with the data pointer in the data structure used by some kernels so that it points to the data structure that the attacker prepared in user mode, which launches an attack by affecting kernel behavior. PAN prevents the kernel from accessing user-mode data, thereby preventing such attacks.

Using some kernel attack methods, an attacker tampers with the code pointer in the data structure used by some kernels so that it points to privilege escalation code in user mode, and then triggers execution of the privilege escalation code by using a system call. PXN prevents the kernel from directly executing user-mode code, thereby preventing such attacks.

CFI

Return-oriented programming (ROP) and jump-oriented programming (JOP) are attack means to redirect program control flows to the code snippets of existing programs by exploiting program vulnerabilities. Attackers combine these code snippets to implement complete attack behavior.

A common method for implementing ROP/JOP attacks is to exploit a program vulnerability to overwrite a function pointer stored in memory. Therefore, a targeted check can be performed. CFI adds additional checks to confirm that control flows stay within the preset scope, in order to mitigate ROP/JOP attacks. If undefined behavior is detected in a program, the program execution is discarded. Although CFI cannot prevent attackers from exploiting known vulnerabilities or even rewriting function pointers, it can strictly limit the scope of targets that can be effectively called, making it more difficult for attackers to exploit vulnerabilities.

EMUI uses Clang CFI and stack protection technologies to reduce ROP/JOP attack threats to the kernel.

- CFI adds a check before each indirect branch to verify the validity of the target address and prevent an indirect branch from jumping to an arbitrary code location.
- The compiler supports link-time optimization (LTO) to determine all valid call targets for each indirect branch.
- Kernel modules can be loaded at runtime. Cross dynamic shared object (cross-DSO) can be enabled in compilation so that each kernel module contains information about valid local branch targets and the kernel looks up information from the correct module based on the target address and the modules' memory layout.
- EMUI checks the stack layout when the function runs to the end and exits to prevent attackers from exploiting the overflow vulnerability to modify the return address.

Kernel Attack Detection

On the basis of the attack characteristics and threat model of kernel vulnerabilities, EMUI monitors and collects app behavior in the kernel and determines whether the behavior matches the threat model. It can detect kernel vulnerability exploit attacks in real time and block such an attack. Once EMUI detects a risky app with malicious behavior, it immediately generates a warning that prompts users to handle the risky app and synchronizes the warning to the cloud. Huawei is the first in the industry to implement this function in mobile systems, which enhances Huawei's detection and defense capabilities and competitiveness in the security of EMUI systems.

MAC

EMUI supports the SELinux feature. When a device is started, MAC policies are loaded to the system kernel and cannot be dynamically changed. This feature applies MAC to all processes when they access resources such as directories, files, and device nodes, and applies root-capability-based MAC to local processes with the root permission. This prevents malicious processes from reading and writing protected data or attacking other processes and limits the system impact of processes that are maliciously tampered with to a local scale, providing strong support for the security defense of upper-layer apps.

EMUI also supports the secure computing (seccomp) feature that restricts the system calls that can be invoked by upper-layer application processes based on the rule files in the read-only file systems, preventing malicious apps from using sensitive system calls to compromise the system.

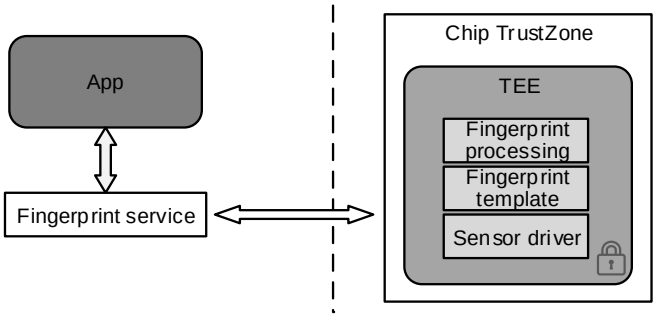
Identity Authentication

Fingerprint Recognition

EMUI provides two fingerprint recognition modes: capacitive and optical. Both modes have similar recognition capabilities (recognition rate and anti-counterfeiting rate). Capacitive fingerprint recognition is applicable to devices with external fingerprint sensors, while optical fingerprint recognition is applicable to devices with under-display fingerprint sensors.

The following figure shows EMUI's fingerprint recognition security framework.

Figure 4-1 Fingerprint recognition security framework



EMUI establishes a secure channel between a fingerprint sensor and iTrustee. Fingerprint information is transmitted to iTrustee through this secure channel, and the REE cannot obtain the information. EMUI collects fingerprint image information, extracts features, detects live fingers, and compares features in iTrustee, and performs security isolation based on the TrustZone. The REE fingerprint framework is only responsible for fingerprint authentication initiation and authentication result data, and does not involve fingerprint data.

Fingerprint feature data is stored in the iTrustee secure storage, and data encryption and integrity protection are implemented using high-strength cryptographic algorithms. The key for encrypting fingerprint data cannot be obtained externally, ensuring that fingerprint data is not leaked. No external third-party app can obtain fingerprint data or transfer such data outside of iTrustee. EMUI does not send or back up any fingerprint data to any external storage media including the cloud.

EMUI's fingerprint recognition supports the anti-brute force cracking mechanism. If the fingerprints of a user fail to be identified five consecutive times in the screen-on state, fingerprint recognition will be disabled for 30 seconds. In the screen-off state, fingerprint recognition is disabled for 30 seconds after 10 consecutive failed fingerprint recognition attempts. If a user fails fingerprint recognition 20 consecutive times, the user must enter the password to unlock his/her device.

Dirty or damaged fingerprint sensors, dirty or wet fingers, and other external factors may affect the recognition rate, and should be avoided.

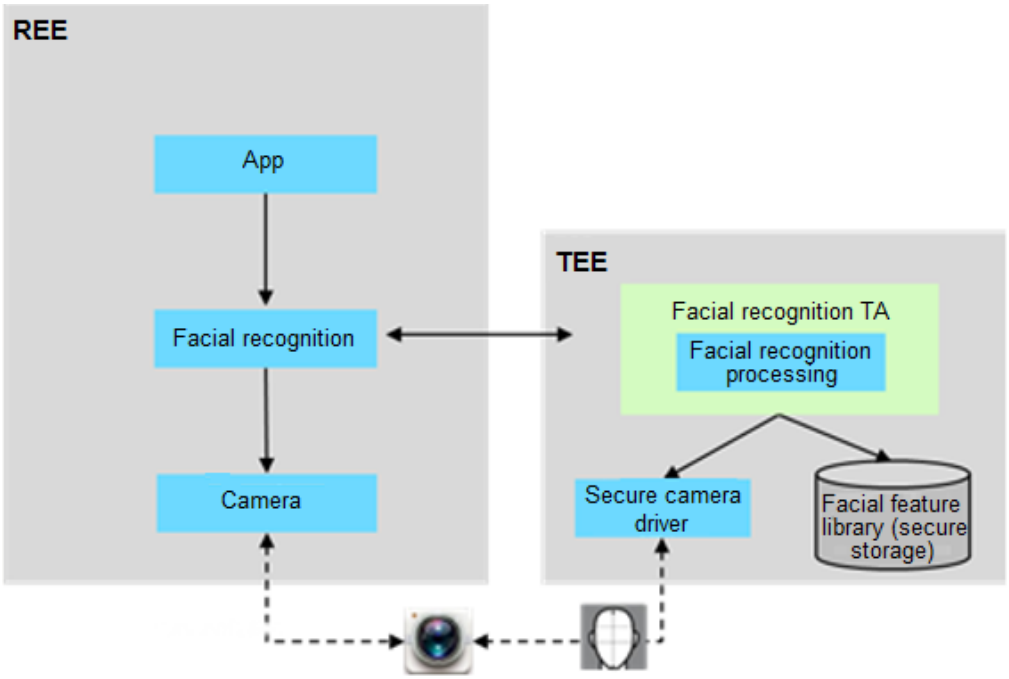
Fingerprint recognition facilitates identity recognition, but users may easily forget their lock screen passwords. Currently, if a user does not use his/her unlock password within 72 hours, the user is compelled to enter the password to unlock the screen, in order to reduce the likelihood of a forgotten password.

Facial Recognition

EMUI provides two types of facial recognition capabilities: 2D and 3D. Only devices with 3D face recognition capabilities can use this technology. The recognition rate and anti-counterfeiting capability of 3D face recognition are better than those of 2D face recognition. 3D face recognition can be applied to payment scenarios, whereas 2D face recognition cannot.

The following figure shows EMUI's facial recognition security framework.

Figure 4-2 Facial recognition security framework



EMUI establishes a secure channel between the camera and iTrustee. Face image information is transmitted to iTrustee through this secure channel, and the REE cannot obtain the information. EMUI collects face images, extracts features, detects live faces, and compares features in iTrustee, and performs security isolation based on the TrustZone. The external facial framework is only responsible for facial authentication initiation and authentication result data, and does not involve facial data.

Facial feature data is stored in the iTrustee secure storage, and data encryption/decryption and integrity protection are implemented using high-strength cryptographic algorithms. The key for encrypting facial feature data cannot be obtained externally, ensuring that facial feature data is not leaked. No external third-party app can obtain facial feature data or transfer such data outside of iTrustee. EMUI does not send or back up facial data (either encrypted or unencrypted) to any external storage media including the cloud.

EMUI's facial recognition supports the anti-brute force cracking mechanism. If the face of a user fails to be identified five consecutive times, the user must enter his/her password to unlock the screen.

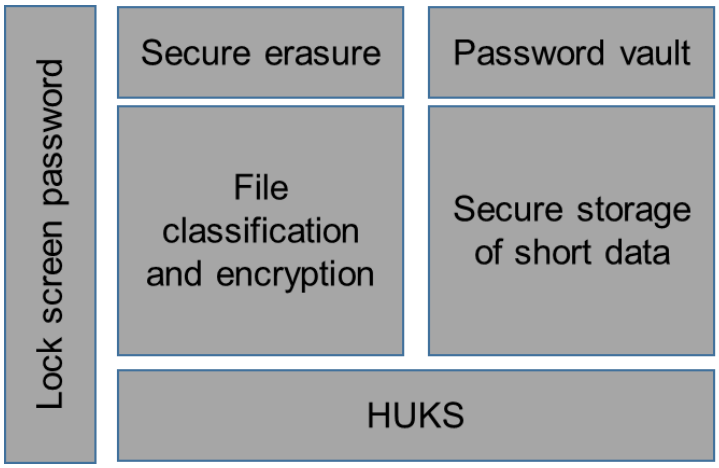
The facial recognition rate is different for twins and siblings who are similar in appearance, as well as children under 13 years of age. Fingerprint recognition or password authentication can be used in such cases.

3D face recognition is a strong biometric authentication, whereas 2D face recognition is weaker. If a 3D face recognition user does not enter his/her unlock password within 72 hours, the user is prompted to enter the password to unlock the screen.

5 Data Security

This chapter describes EMUI data security protection. The EMUI file system is divided into a system partition and a user partition. The system partition is read-only, isolated from the user partition, and inaccessible from common apps. For data stored in the user partition, the system provides file-based data encryption and directory permission management to restrict data access between apps. EMUI provides various mechanisms for critical data in the user partition to ensure the secure storage, use, and destruction of highly sensitive user data. Such mechanisms include lock screen password protection, secure storage of short data, secure erasure, and password vault. In addition, EMUI provides app developers with HUKS framework capabilities, enabling them to securely use keys to protect confidential data in apps.

Figure 5-1 Data security architecture



HUKS

The HUKS is a key and certificate management system based on the J2EE Connector Architecture/Java Cryptography Extension (JCA/JCE) architecture in the EMUI, and provides Keystore and Crypto APIs for apps. It provides key management, symmetric/asymmetric encryption and decryption, certificate management, and other functions. The HUKS enables EMUI app developers to manage keys and certificates throughout their lifecycles and call encryption and decryption algorithms. It provides device authenticity verification based on device certificates. The cloud server can authenticate the EMUI devices through certificate

authentication. In combination with biometric authentication, the HUKS can provide services such as login and payment with iTrustee security for payment apps.

HUKS keys and certificates are stored in iTrustee. All keys are encrypted by the HUK using AES_256_GCM and then stored in the file system. When a key is used, it is decrypted in iTrustee and also used in the calculation of encryption and decryption keys. A plaintext key is always stored in iTrustee, and key encryption and decryption are protected by iTrustee.

The HUKS strictly controls access to keys in order to prevent unauthorized access. A key can be accessed only by the app that generated the key. When a key is generated, the HUKS records the app's identity information such as the UID, signature, and package name. An EMUI app shall pass identity authentication before accessing a key. EMUI apps can use biometric authentication functions (such as fingerprint and facial recognition) to enhance access control for keys. The HUKS allows key access and operations only after confirming the biometric authentication result.

The HUKS also provides a key attestation function. With this function, a Huawei device certificate injected by the production line can be used to authenticate keys in iTrustee. Each device has a unique device certificate. The HUKS also provides an ID attestation function, which offers trusted device identifier authentication capabilities for the cloud, covering device identifiers such as the SN and IMEI. The HUKS allows EMUI apps to apply for certificates from the certificate authority (CA) server through protocols such as the Certificate Management Protocol (CMP) and Simple Certificate Enrollment Protocol (SCEP). In addition, the HUKS integrates the Standard Of auThentication with fingERprint (SOTER) framework to provide SOTER-based biometric authentication for EMUI apps.

Lock Screen Password Protection

EMUI allows lock screen passwords with six digits (default), four digits, an unfixed number of (four or more) digits, an unfixed number of (four or more) hybrid characters, and patterns. After a user sets a lock screen password, the password can be used to unlock the device and provide entropy for the file system encryption key. This means that even if an attacker obtains a device, the attacker cannot access data protected by the lock screen password entropy without a screen lock password.

EMUI increases the password attempt interval upon input of each incorrect password to prevent password brute forcing. A longer password and more character types indicate longer time needed to attempt all combinations.

Lock screen passwords are protected using the HUK. When a user creates or modifies a lock screen password, or unlocks the screen using the lock screen password for verification, the lock screen password is processed in iTrustee. This means that brute force cracking attempts can only be made on attacked devices. If a lock screen password contains six digits and letters, it will take 8 years to attempt all possible combinations using brute force cracking, even if the attempt interval increase is not considered.

Even if the system beyond iTrustee is compromised, the lock screen password will still remain protected.

Lock screen password protection is powerful. A lost password may lead to data loss on the phone. As such, it is recommended that users properly store their password and back up their data.

In cases where a mobile phone is lost, EMUI provides data encryption protection for the user file system, preventing unauthorized users from launching physical attacks (for example, directly reading the flash memory) to obtain device data and cause user data breaches.

Data Classification and Hierarchical Encryption

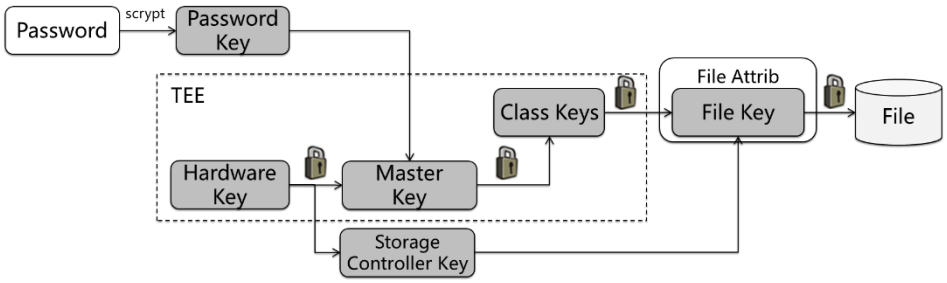
File System Classification and Hierarchical Encryption

Since EMUI 5.0, the kernel's encryption file system module and hardware encryption/decryption engine are used to deliver file-level encryption through the XTS-AES 256 algorithm.

To ensure the security of user data and application experience, EMUI provides the following data encryption solutions:

- Credential Encryption (CE), Sub-Enhanced Credential Encryption (SECE), and Enhanced Credential Encryption (ECE) that work with the device lock screen password: Apps use this type of data encryption solution by default. In this type of solution, class keys are relevant to the lock screen password and are protected by using both the lock screen password and HUK. Specifically,
  - CE: CE-protected data is accessible only after an EMUI device is unlocked for the first time.
  - SECE: Enhances CE. When a device is locked, SECE-protected files cannot be opened, but files can be created and written. For example, email attachments can be downloaded and written in the background.
  - ECE: Enhances SECE. When a device is locked, ECE-protected files cannot be opened or created until the device is unlocked.
- Device encryption (DE) that is irrelevant to the lock screen password: DE-protected data, such as wallpapers, alarm clocks, and ringtones, can be accessed after a device is powered on, independent of whether the device is locked or not. DE-based class keys are protected using the HUK and irrelevant to the lock screen password.
- Non-encryption (NE): Data is not encrypted, which is rarely used. Such data includes OTA upgrade packages.

Figure 5-2 File encryption levels



Secure Storage of Short Data

Some apps may process short sensitive data, such as user passwords and authentication credentials. It is complex to store this type of data in a file system. Such short data can be stored in the secure storage.

Encrypted short data (ciphertext) is protected using the HUK and app identity. Decryption and encryption are performed in the TEE, and the key for encrypting data is stored in the TEE. A single piece of ciphertext is protected in AES_256_CCM mode, and batch ciphertext is protected in AES_256_CBC mode.

Two types of short data can be stored in the secure storage:

- Sensitive data: sensitive data of critical assets, such as user passwords.
- Authentication credentials: authentication credentials or tokens, which are usually the credentials for an app to use a service. For example, when an app connects to a server, the token is used for session validation.

The secure storage service verifies the signature, package name, user identity (UID), and other information of the app that queries the stored data, in order to verify the access permission and ensure access security.

Secure Erasure

Normal factory restore operations cannot ensure that all data stored on physical storage is completely deleted. While logical addresses are usually deleted for efficiency, this method does not clear the physical address space, and the data can often be restored.

In factory restoration, EMUI erases stored data securely. An overwrite command is sent to the physical storage to erase the data. Erased data is all 0s or all 1s. This ensures that sensitive user data cannot be restored using software or hardware means, and protects data security if devices are resold or abandoned. The following compatibility definition document (CDD) requirement is met: [C-0-3] MUST delete the data in such a way that will satisfy relevant industry standards such as NIST SP800-88.

Password Vault

An ever-increasing number of apps are becoming available, and logins to these apps require user names and passwords, which can be forgotten at any time. A password vault is provided to store user app login information (user names and passwords) and associate the login information with relevant face IDs, touch fingerprints, or lock screen passwords so that the password vault automatically fills in a user's user name and password for login.

The password vault (supported only by Huawei HiSilicon platform-based devices of EMUI 9.0 and later) stores encrypted app accounts and passwords in the SQLite database of the file system on a device, providing hardware-level encryption and storage capabilities. The passwords are encrypted using AES_256_CCM. The encryption key is protected by iTrustee, and encryption/decryption is always performed in iTrustee.

Currently, the password vault does not provide cloud synchronization or backup capabilities. The account and password data stored in the password vault can be encrypted and transferred between Huawei devices that support the password vault through Phone Clone (password vault clone is available only to Huawei devices that support the PKI certificate). Alternatively, users can restore encrypted data stored on a PC back to the device that previously possessed the data.

The password vault data transmitted in the Phone Clone process is encrypted using AES_256_CBC. The encryption key is obtained through key exchange using the asymmetric key generated by two phones in iTrustee. Key exchange is performed in iTrustee, which also protects the obtained clone encryption key. Encryption and decryption are performed in the REE, facilitating the quick execution of the clone operation for password vault data.

The password vault data transmitted in the PC-based backup process is also encrypted using AES_256_CBC, and the encryption key is derived from the HUK. A device's backup data on a PC cannot be restored on other devices.

6 App Security

This chapter focuses on the security mechanisms for apps on EMUI. Apps can be obtained from various channels, which can sometimes result in users downloading malicious apps. If not properly handled, malicious apps may compromise the security and stability of the system and present security risks to personal user data, and even personal property.

EMUI provides a complete set of app security solutions to enable a secure environment for apps:

- When apps are released in the Huawei AppGallery, security detection is performed to ensure that malicious apps are accurately identified. In addition, convenient security detection services are provided for developers to ensure the security of app releases.
- During app installation, the signature verification mechanism prevents apps from being maliciously tampered with.
- When an app is running, app sandbox, runtime memory protection, secure input, and other mechanisms are used to prevent data generated in the app from being maliciously read by unauthorized apps, in order to prevent user data breaches.
- The EMUI system provides various functions to ensure a secure environment for apps. These functions include static app threat detection, AI-based app threat detection, and malicious website detection.

App Release Security Detection

The Huawei AppGallery uses the SecDroid security detection platform to perform strict security tests on each app to be released. The sandbox environment is used to analyze vulnerabilities, viruses, ads, malicious behavior, and privacy for each app in order to ensure a secure app release. Convenient security detection services are also provided to developers.

- Vulnerability analysis
  - Static vulnerability analysis: allows static scanning and analysis of Android packages (APKs) for potential vulnerabilities. It detects component security, data security, traffic consumption, insecure command execution, password autocomplete, service enabling, WebView security, and sensitive behavior, and covers dozens of analysis monitoring points.
  - Dynamic vulnerability analysis: dynamically monitors APKs running in the sandbox, and analyzes security vulnerabilities in the APKs based on captured dynamic run logs.
- Virus analysis: uses the SecDroid virus analysis engine, as well as antivirus engines from well-known antivirus engine vendors, such as Avast, ANTIY, 360, and Tencent, to comprehensively detect viruses in APKs.

- Ad analysis: uses dynamic sandbox execution technology and static feature analysis technology of the SecDroid to detect third-party ads in apps based on the dynamic and static rule features of ad software development kits (SDKs).
- Malicious behavior analysis: uses dynamic sandbox execution technology and static feature analysis technology of the SecDroid to detect and analyze sensitive app behavior.
- Privacy analysis:
  - Static privacy analysis: uses data flow tracking technology, analyzes static data flows of APKs, and monitors pollution sources and breach points to identify the complete path along which private data (such as phone numbers, SMS messages, and locations) is breached.
  - Dynamic privacy analysis: dynamically monitors APKs running in the sandbox, and analyzes privacy data collection and transmission in the APKs based on captured dynamic run logs.

Signature Verification During App Installation

Only apps with complete signatures can be installed in EMUI. App signatures can be used to verify the integrity and source legitimacy of apps. The system verifies the signature of an app to check whether it has been tampered with before installing the app. Apps that fail verification cannot be installed.

The system also verifies app signatures before updating pre-installed or user-installed apps. Such an app can only be updated when the signature of the target version is the same as the existing signature. This prevents malicious apps from taking the place of existing ones.

App Sandbox

EMUI provides an app sandbox mechanism. This mechanism enables all apps to run in isolation within the sandbox in order to ensure runtime security. When an app is installed, the system allocates a private storage directory to the app which cannot be accessed by other apps, ensuring static data security. Sandbox isolation technology protects the system and apps from malicious attacks.

The system allocates a unique UID to each app and builds an app sandbox based on UIDs. The sandbox provides multiple kernel access control mechanisms, such as discretionary access control (DAC) and MAC, to restrict apps from accessing files and resources outside the sandbox. By default, all apps are sandboxed. To access information outside the sandbox, an app needs to use services provided by the system or open interfaces of other apps and obtain required permissions. The system will prevent access if an app does not have required permissions.

Apps with the same signature can share a UID, and share code and data in the same sandbox.

Runtime Memory Protection

Malicious apps usually obtain memory addresses by viewing the memory if the allocated memory addresses are relatively fixed during app operation. EMUI provides ASLR and DEP to address this issue. ASLR is a security technique used to prevent the exploit of buffer overflow vulnerabilities. It randomizes the layout of linear areas such as heaps, stacks, and shared libraries, making it harder for attackers to predict target addresses and preventing them from locating attack code, which leads to reduced overflow attacks. ASLR denies attackers the opportunity to take advantage of memory vulnerabilities. DEP marks specific memory areas as non-executable. This helps prevent attacks exploiting memory vulnerabilities.

Secure Input*

EMUI provides secure input when users are entering passwords. Once secure input is enabled, the system will automatically switch to secure input when a user enters a password. Secure input and common input are managed separately. To safeguard user passwords, secure input does not remember or predict any entered passwords. It cannot connect to the Internet or collect user passwords. After secure input is enabled, screen recording cannot be performed in the backend, and no third-party apps can capture screenshots.

*Note: Third-party input methods will be used in some bank APKs, and secure input does not take effect in such cases.

App Threat Detection

Security risks may exist in apps as a result of unknown third parties, and downloading apps from unverified sources can introduce malicious threats. EMUI can check whether app sources are legitimate during app installation. By default, apps from unknown third parties cannot be installed. It is recommended that default security settings be retained to prevent unnecessary risks.

EMUI has an industry-leading built-in antivirus engine, which is used to detect viruses in user-installed apps. The antivirus engine supports local and online virus scanning and removal, to ensure that app risks are identified regardless of whether user devices are connected to the Internet. The antivirus engine can scan viruses during app installation and in the backend.

Once a virus is detected, a risk warning is reported to the user, prompting them to handle the virus.

AI Security Protection*

EMUI provides a hardware-based AI computing platform for device security protection. It has a built-in industry-leading AI antivirus engine encompassing a security defense-oriented AI model built upon deep learning and training. EMUI monitors the behavior of unknown app software in real time to identify new viruses, new variants of existing viruses, and dynamic loading of malicious programs, and runs the AI model on devices to analyze the behavior sequence of unknown software. This quickly and effectively detects threats and improves app threat detection capabilities. Once a malicious app is detected using AI security defense, the system will generate a warning immediately to prompt the user to handle the app.

*Note: This function is available only for certain chip models in China.

Malicious Website Detection*

EMUI can detect phishing websites, or websites with malicious threats, when a Huawei browser is used or text messages are sent. When a Huawei browser is used to browse a malicious page, EMUI checks the website so that the Huawei browser can block access and warn the user of security risks. This function can also identify malicious website URLs in received text messages and warn users of security risks.

*Note: This function is available only for certain chip models in China.

7 Network and Communication Security

Secure connections are needed when devices connect to the network. Otherwise, they may connect to or be connected to malicious sites and leak data. This chapter focuses on EMUI's security mechanisms for network connection and transmission, and security protection that EMUI provides for device communication, and device interconnection for data transmission.

VPN

A VPN enables a user to establish a secure private network using public network links for secure data transmission. EMUI supports the following VPN protocols and authentication modes:

- Point-to-Point Tunneling Protocol (PPTP), supporting Microsoft Challenge Handshake Authentication Protocol version 2 (MS-CHAP v2) password and RSA SecurID for user authentication as well as Microsoft Point-to-Point Encryption (MPPE)
- Layer Two Tunneling Protocol (L2TP)/IP Security (IPsec), supporting MS-CHAPv2 password, pre-shared key (PSK), and certificate authentication
- Internet Key Exchange version 2 (IKEv2)/IPsec, supporting shared key, RSA certificate, Elliptic Curve Digital Signature Algorithm (ECDSA) certificate, Extensible Authentication Protocol MS-CHAPv2 (EAP-MSCHAPv2), or EAP Transport Layer Security (EAP-TLS) for authentication
- IPsec Xauth PSK, IPsec Xauth RSA certificate authentication, and IPsec Hybrid RSA certificate authentication
- Cisco IPsec, using password, RSA SecurID, or CRYPTOCard for user authentication EMUI supports the following VPN functions:

For networks based on certificate authentication, IT policies use VPN configuration description files to specify the domains that require VPN connections.

A VPN can be configured per app, for more accurate VPN connection.

A VPN can remain enabled. A user does not need to enable the VPN manually after connecting to the network.

The VPN function can be enabled or disabled for devices managed by the MDM solution, thereby ensuring data security within an organization.

TLS

Devices support TLS v1.0, v1.1, v1.2, and v1.3. They also support SSL/TLS through a third-party OpenSSL protocol stack.

TLS is a security protocol that protects data and data integrity during communication. Application-layer protocols can run transparently over TLS. TLS is responsible for the authentication and key exchange required for creating encrypted channels. Data transmitted using application-layer protocols is encrypted when passing through TLS. This ensures the communication stays private.

A device enables TLS v1.3 by default for all TLS connections. Compared with TLS v1.2, TLS v1.3 improves performance and security (for example, by removing weak and rarely used algorithms). The TLS v1.3 encryption suite is not user-defined, and after TLS v1.3 is enabled, the supported encryption suite remains enabled and ignores any operations that attempt to disable it.

Wi-Fi Security*

EMUI provides multiple authentication modes for users requiring different levels of security. Such authentication modes include Wi-Fi Protected Access (WPA)/Wi-Fi Protected Access 2 (WPA2) PSK, Wi-Fi Protected Access 3 (WPA3) for some products, 802.1x EAP, and WLAN Authentication and Privacy Infrastructure (WAPI).

To prevent an EMUI device from being tracked and enhance user privacy protection, the device uses a random MAC address to scan the network before connecting to Wi-Fi.

From EMUI 10.0, the device uses a random MAC address by default when connecting to Wi-Fi (supported by some products as it depends on chip capabilities). If a user trusts the target network, the user can manually change the setting and use the MAC address of the device for connection.

In addition, devices also support the Wi-Fi hotspot function, which is disabled by default. Wi-Fi hotspot, once enabled, supports WPA2 PSK authentication to ensure the connections are secure.

Public Wi-Fi may be convenient, but at the same time, it may be used illegally to steal users' private data and perform phishing. This can undermine a user's privacy and even result in financial losses. EMUI provides a Wi-Fi threat detection engine for access points. It detects Wi-Fi hotspots before connection. If any security risks are detected, it will prompt users so that they can take measures to ensure the connection is secure.

*Note: This function is only available in China.

Protection Against Fake Towers*

Unauthorized users can obtain user location and identity information by deploying fake towers, or send advertisements and fraud messages to users, which not only seriously interferes with a user's normal communication, but can also result in financial losses. EMUI provides chip-level protection against fake towers with its HiSilicon chips (not supported on other chip platforms). It compares and analyzes network parameter characteristics for access and reselection of fake GSM/LTE towers and network parameter characteristics of normal towers, and rejects the residence and access of identified fake towers. (Fake LTE towers can only be identified by some chip platforms.) In addition to decoding system messages, the device can identify fake towers through combined process characteristics such as fake tower attack without authentication redirection. This prevents a device from camping on or accessing cells with such characteristics.

*Note: This function is only available in China.

8 Distributed Security

Device Interconnection Security

To ensure user data flows securely between devices in a distributed scenario, the devices must be trusted by each other, that is, they must have established a trust relationship, and be able to establish a secure channel after the trust relationship is verified.

The trust relationship can be established between EMUI devices under the same HUAWEI ID or between EMUI devices and IoT devices.

Interconnection Security for EMUI Devices Under the Same HUAWEI ID

EMUI provides authentication services for devices that are logged in with the same HUAWEI ID. Each EMUI device that is logged in with a HUAWEI ID generates a public-private key pair using elliptic curve cryptography as the device identifier, and applies for public key authentication from the Huawei Cloud server. In the device interconnection service, devices with the same HUAWEI ID that have passed public key authentication can authenticate each other and exchange their identity public keys to verify whether each other is a trusted device. Based on the identity public-private key pair, devices logged in with the same HUAWEI ID can exchange session keys and establish a secure communication channel. Bogus devices and devices not registered under this HUAWEI ID cannot be mutually authenticated.

Networking Service for EMUI Devices Under the Same HUAWEI ID

The device trust relationship verification service supports trusted networking of EMUI devices that are logged in with the same HUAWEI ID, including mobile phones, tablets, large-screen devices, and PC. When the trusted networking service is enabled on an EMUI device, the device authentication service performs identity authentication on each nearby device that is logged in with the same HUAWEI ID, and exchanges the session key between devices.

When a user enables services such as family album or HUAWEI Vision for continuing playing a video on another EMUI device under the same HUAWEI ID, device authentication and session key exchange are implemented for the devices based on the trusted device networking service, and the exchanged session key is used to encrypt data transmitted between the devices.

IoT Device Interconnection Security

EMUI supports P2P trust relationships between devices that do not have a HUAWEI ID login UI on themselves (such as AI speakers, smart home devices, and wearable devices) and EMUI

devices (such as mobile phones and tablets), and allow devices that have established a trust relationship to establish secure connections for E2E encryption and transmission of user data.

IoT Service Identifiers of EMUI Devices

An EMUI device generates different identifiers for different IoT device management services to isolate these services. The identifier can be used for authentication and communication between an EMUI device and an IoT device. Similar to the device identifier generated when a HUAWEI ID is used to log in to the device, the IoT service identifier is also an Ed25519 public-private key pair. The key pair is generated using elliptic curve cryptography in iTrustee of the EMUI device, and the plaintext private key is not transmitted out of iTrustee.

IoT Device Identifiers

An IoT device can generate its own device identifier for communicating with EMUI devices. It also uses elliptic curve cryptography to generate an Ed25519 public-private key pair, and stores its private key locally. Each time the device is restored to factory settings, the public-private key pair will be reset.

The identifier can be used for secure communication between an EMUI device and an IoT device. After both devices authenticate the service identifier or device identifier, they can exchange session keys and establish a secure communication channel.

P2P Trusted Binding Between Devices

An EMUI device and an IoT device establish a P2P trust relationship by exchanging the EMUI device's IoT service identifier and the IoT device identifier.

During this process, the user needs to enter or scan the PIN provided by the IoT device on the EMUI device. PIN is either dynamically generated if the IoT device has a screen, or preset by the manufacturer if it does not have a screen. A PIN can be a 6-digit number or a QR code. The EMUI and IoT devices then use the Password-Authenticated Key Exchange (PAKE) protocol for authentication and session key exchange, thereby protecting the integrity of the exchanged identifiers.

On an EMUI device, the peer's identity public key is stored in iTrustee. This ensures that the trust relationship with the communication peer end cannot be tampered with.

Communication Security Between EMUI Devices and IoT Devices

When an EMUI device and an IoT device communicate with each other after establishing a trust relationship, they both verify that the peer end is a bound device, and exchange the session key by using the locally stored identity public key of the peer.

When Huawei Share OneHop, collaborative facial recognition, or multi-screen collaboration is used to share data between a mobile phone and a Huawei PC, large-screen device, or tablet, a P2P trust relationship can be established through trusted binding. For encryption of the transmitted data, the session key obtained during authentication is used.

When AI Life is used, the mobile phone will be connected to a Huawei-certified third-party sensitive IoT accessory, where a P2P secure connection is established through trusted binding. For E2E encryption of the transmitted data, the session key obtained during authentication is used.

Collaborative User Identity Authentication of the Distributed System

EMUI builds distributed identity authentication capabilities for trusted devices that comprise a distributed system. This breaks the boundaries between devices and provides flexible identity authentication based on user operations and service requirements. When users operate

multiple devices under the same HUAWEI ID on the same LAN, they can select the most easily accessible device among those with the same security level for access and identity authentication.

Collaborative user identity authentication (referred to as collaborative authentication) is applicable only to trusted devices that comprise a distributed system. EMUI generates an Ed25519 public-private key pair in iTrustee for the trusted devices involved in collaborative authentication, which is used as the collaborative identity to sign and verify collaboration scheduling information and data. The collaborative identity is securely transmitted from iTrustee of one device to iTrustee of other devices in the same networking in end-to-end encryption.

When two EMUI devices are interconnected, EMUI authenticates whether they are trusted. If they are, it establishes a secure data transmission channel between them, which is used for transmitting encrypted scheduling information of the collaborative authentication service. The collaborative authentication services of both devices use the collaborative identity to sign the data sent out from the local device and verify the data received from the peer device, ensuring that collaborative scheduling data for identity authentication is not tampered with during transmission.

Distributed Permission Management

The EMUI provides a permission management mechanism, which is designed to allow or restrict apps' access to APIs and resources in distributed scenarios. By default, no permissions are granted to apps, and access to protected APIs or resources is restricted to ensure security of such APIs and resources.

In distributed scenarios, when distributed apps on different trusted devices collaborate with each other, the distributed permission management system performs strict access control on the apps to ensure that the apps can access distributed resources or capabilities in the right way both locally and across devices.

Only apps with the distributed access permission are allowed to access resources or capabilities of trusted distributed devices. When a distributed app calls the resources or capabilities of a distributed virtual device, the distributed permission management system first checks whether the app has been granted distributed permission. If not, the distributed app cannot access the resources or capabilities of the virtual device. When an app attempts to access restricted resources or capabilities, the distributed permission management system verifies whether the app has the relevant permissions. If not, the system either denies access or prompts users to conduct dynamic authorization. The app can call distributed resources or capabilities through the distributed application framework over a secure channel only after it is authorized.

On the peer device, the distributed permission management system also maintains authorization information of distributed apps. When a distributed app accesses the resources or capabilities of the peer device, the permission management system strictly controls access on the app. Only permitted apps can access the relevant resources or capabilities.

Any permission changes in a running distributed app are synchronized in real time to the virtual device to ensure real-time and effective access control.

9 Advanced Security

This chapter describes security protection for Huawei Pay and other mobile payment apps. For third-party payment apps, EMUI can identify malicious apps, isolate the payment environment for protection, and encrypt verification codes to ensure payment security.

Huawei Pay

Using Huawei Pay, users can make payments on supported Huawei devices in a convenient, secure, and confidential way. Huawei Pay has enhanced security in both hardware and software design.

Huawei Pay Components

- Secure element: is a chip that has received industry certification and recognition. It complies with digital payment requirements in the finance industry.
- NFC controller: processes NFC protocols and supports communication between the app processor and secure element and between the secure element and POS terminal.
- Huawei Pay app: refers to "Wallet" on devices that support Huawei Pay. This app enables users to add and manage credit and debit cards and make payments. Users can also query their payment cards and other information about the card issuers.
- Huawei Pay server: manages the status of bank cards in Huawei Pay and the device card number stored in the secure element. The server communicates with devices and payment network servers at the same time.

How Huawei Pay Uses the Secure Element

Encrypted bank card data is sent from a payment network or card issuer to the secure element. The data is stored in the secure element and protected by the security functions provided by the secure element. During a transaction, a device directly communicates with the secure element using a dedicated hardware bus through the NFC controller.

How Huawei Pay Uses the NFC Controller

The NFC controller functions as the gateway to the secure element and ensures that all contactless payments are conducted through POS terminals in close proximity to payment devices. The controller marks the payment requests from devices as contactless transactions.

Once a cardholder authorizes payment through fingerprint or password authentication, the controller sends the contactless response prepared by the secure element to the NFC chip. In this manner, detailed payment authorization information for contactless transactions is saved only in the local NFC chip and will not be disclosed to the app processor.

Bank Card Binding

When a user adds a bank card to Huawei Pay, Huawei securely sends the payment card information and other information about the user account and device to the card issuer. The card issuer then determines whether to allow the user to add the card to Huawei Pay.

Huawei Pay uses commands invoked on the server to send and receive packets exchanged with the card issuer or network. The card issuer or network uses these commands to verify, approve, and add payment cards to Huawei Pay. The sessions between clients and servers are encrypted using TLS.

Adding Bank Cards to Huawei Pay

To manually add a payment card, users must enter their name, card number, card expiration date, and card verification value (CVV) code. Users can enter this information in the Wallet app or use the camera function to input the information. If the camera is used to capture payment card information, the Wallet app will attempt to fill in the card number. After all information is entered, the information except the CVV code is verified. The information will be transmitted to the card issuer for verification through the security control. Huawei will not save or use the information such as the CVV code.

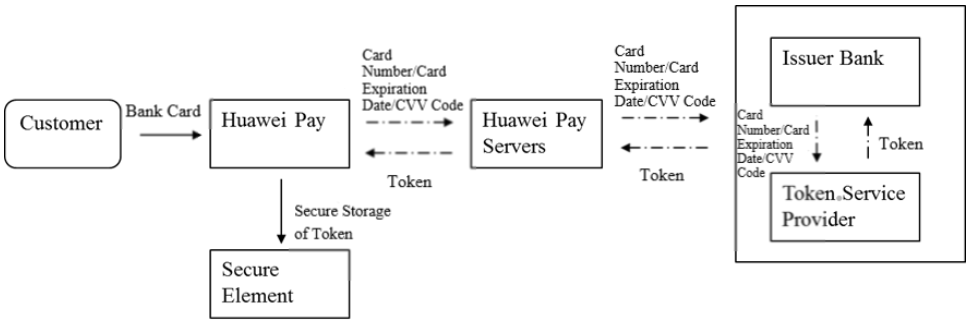
If any terms and conditions are returned by the card issuer for the payment card confirmation process, Huawei downloads the terms and conditions and displays them on the user's device.

If the user accepts the terms and conditions, Huawei sends the accepted clauses and CVV code to the card issuer and carries out the binding process. The card issuer determines whether to allow the user to add the payment card to Huawei Pay according to the user's device information, such as the name, device model, Huawei mobile phone to which Huawei Pay is bound, and approximate location when the user adds the payment card (if GPS is enabled).

The following operations are performed in the binding process:

- The mobile phone downloads the credential file representing the bank card.
- The mobile phone binds the payment card to the secure element.

To ensure the data security and privacy of cardholders, both international organizations and the People's Bank of China have issued relevant standards stipulating that bank card information stored on devices must be replaced with a token. In other words, when the user adds a bank card to Huawei Pay, the card information will be transmitted to the card issuer through the security control measures provided by the issuer. The issuer will then send back an authorized token rather than the actual card number to Huawei Pay. Therefore, the card number stored in the mobile phone is not the actual one. The binding process also requires real-name verification by Huawei and the card issuer to ensure that the HUAWEI ID and bank card belong to the same user.



Additional Verification

Card issuers determine whether to perform additional verification on bank cards. Depending on the functions supported by card issuers, users can select text message verification for additional verification.

Users can select the contact information archived by their card issuers to obtain text message notification and enter the received verification code in the Wallet app.

Payment Authorization

The secure element permits payment only after receiving authorization from the Huawei Pay-capable mobile phone and determining that the user has passed fingerprint or device password authentication.

Fingerprint authentication, if available, is the default authentication mode for payment. Users can use the password instead of fingerprint at any time. If fingerprint authentication fails once, the system automatically prompts the user to enter the password.

Using Huawei Pay for Contactless Payment

If a Huawei mobile phone is powered on and detects an NFC signal, it displays related bank cards. The user can access the Huawei Pay app and select a bank card, or use a specific fingerprint sensor to invoke the payment page when the device is locked.

If the user is not authenticated, no payment information will be sent. After the user is authenticated, the device card number and dynamic security code dedicated for transaction are used during payment.

Suspending, Removing, and Erasing Payment Cards

Card issuers or payment networks can suspend the payment function of Huawei Pay payment cards or remove the cards from devices even if the devices are not connected to cellular or Wi-Fi networks.

Payment with Biometric Features

Huawei Pay users can authenticate payments with fingerprints and facial information, which are stored securely on the device and will not be synchronized to the Huawei Cloud. In addition, the payment information is protected by digital certificate signatures.

International Authoritative Financial Certification

Huawei Pay has obtained the international PCI-DSS certification as well as VISA PCI-CP and CDCVM security certifications, and complies with authoritative security standards in the payment industry.

Transportation Card

After users add a transportation card to HUAWEI Wallet on their Huawei mobile phone, the transportation card company loads their transportation card app to the Secure Element (SE) of the mobile phone over the air, associates the app with a Supplementary Security Domain (SSD), and then downloads and stores personal data of the transportation card into the transportation card app in the SE. Then the associated SSD can provide security assurance for the personal data. After a transportation card is set up in HUAWEI Wallet, users can top up, query information such as the card number and balance, move the transportation card to the cloud, download it from the cloud to the mobile phone, and return the transportation card to get a refund.

Set Up a Transportation Card

After a user adds a transportation card to HUAWEI Wallet and pays the service fee, the mobile phone initiates a card registration request. Under the protection of the Secure Channel Protocol (SCP) of the Issuer Security Domain (ISD), Huawei's Trusted Service Manager (TSM), namely, the Secure Element Issuer (SEI) TSM, creates an independent SSD for the transportation card, and converts operations on the transportation card app to the Application Protocol Data Unit (APDU) instructions according to the GlobalPlatform Card (GP Card) specifications. Under the protection of the ISD SCP, the mobile phone downloads the APDU instructions to the security chip, instantiates the transportation card app, and transfers the card instance to the previously created SSD. The transportation card company's TSM, namely, the Service Provider (SP) TSM, manages the SSD key. The SP TSM uses the SSD key to provide SCP encryption protection while downloading personal data, such as the card key (one key for one card), to the transportation card app in the SE. The transportation card is then set up on the mobile phone.

Top Up a Transportation Card

After a user adds money to the transportation card, the mobile phone initiates a balance top-up request. Once the SP TSM has confirmed that the payment has gone through, it sends a top-up initialization instruction to the card app in the SE, initiating a random number challenge. After receiving the challenge, the card app uses its key to calculate and return the calculation result, which the SP TSM verifies by using the card key. If the verification is successful, the SP TSM considers the card valid. Then, the SP TSM uses the card key to perform another calculation and encapsulates the calculation result into the card app in the SE downloaded in the top-up instruction. The card also verifies the calculation result. If the verification is successful, the card considers the SP TSM valid. Then, the card adds the top-up amount to the card balance. The card key is stored in both the card app in the SE and the hardware encryptor of the SP TSM, with hardware-level security. It is not available to any third party and only the SP TSM of the transportation card company can complete the top-up.

Swipe a Transportation Card

The NFC controller of the mobile phone allows contactless communication between the transportation card app in the SE and the card reader of the transportation company. After the transportation card app and the card reader are mutually authenticated, the card app deducts an amount from the balance as instructed by the card reader.

Move a Transportation Card to the Cloud

If a user temporarily does not need a transportation card already set up in HUAWEI Wallet, the user can move it from the mobile phone to the cloud. Relevant card data will be stored in the SP TSM. When the card data is backed up to the cloud, the SP TSM delivers a migration instruction to the card app in the SE. The card app then obtains data as instructed, encrypts it and adds the message authentication code (MAC) in the card, and then returns the data to the SP TSM. Upon receipt, the SP TSM verifies the MAC, decrypts the data, and stores the decrypted data. The card data is encrypted and the MAC is added within the card, ensuring confidentiality and integrity during data transmission.

Return a Transportation Card

If a user no longer needs a transportation card, the user can initiate a card return request in HUAWEI Wallet. During this process, the SP TSM obtains the card balance, and then the SEI TSM deletes the card from the SE. The SP TSM returns the card balance to the user's bank card used for the initial payment.

Door Key

After users add a door key to HUAWEI Wallet on their Huawei mobile phone, the door key is loaded to the SE of the mobile phone over the air, and an asymmetric key pair is generated in

the security chip. The key administrator (property or campus management personnel) uses the public key in the security chip to encrypt the key and personal data of the card for the door key and delivers it to the SE. The SE then decrypts the data, stores it, and ensures that the card key is not replicable or accessible.

Set Up a Door Key

After a user adds a door key to HUAWEI Wallet or a service app, the mobile phone initiates a key registration request. Under the protection of the ISD SCP, the SEI TSM (Huawei side) converts operations on the card app for the door key into APDU instructions according to the GP Card specifications, downloads the APDU instructions to the security chip, and instantiates the card app for the door key. During the instantiation, an asymmetric key pair is generated for encrypting personal data and transferred from HUAWEI Wallet to the service app or service backend, which then uses the public key of the card app instance to encrypt the card key and personal data. The SP TSM (Huawei side) then converts the encrypted data into APDU instructions according to the GP Card specifications, and downloads the APDU instructions to the security chip under the protection of SCP. The card app for the door key then uses the private key in the security chip to decrypt the card key and personal data.

Swipe a Door Key

The NFC controller of the mobile phone allows contactless communication between the card app for the door key in the SE and the card reader of the property or campus management company. After the card app for the door key and the card reader are mutually authenticated, the card app returns the private ciphertext data as instructed by the card reader.

Secure Keys*

Second-generation U key (such as USB key and audio key) is the main network transaction security solution for banks. Because a U key is external security hardware, it is prone to damage and loss, is inconvenient to carry, and has a low use rate and poor user experience. For apps with a mobile payment function, the main security strategy is to bind with mobile phones during transactions through bank payment channels. The transactions are confirmed through SMS messages and pose high security risks. Users are therefore concerned that their money may be stolen during payment. Huawei secure keys are combined with an independent internal secure element. The secure element is an authenticated chip widely accepted in the industry and supports banks' mobile phone certificate services. Huawei secure keys combine traditional plug-in U keys with phones to form portable secure keys in order to provide finance-level hardware protection for electronic payment.

When a user enables secure keys, EMUI's Trusted Service Manager (TSM) establishes a Secure Channel Protocol (SCP) channel with the secure element to create a trusted, independent, and secure running space within the secure element. The bank app then generates an independent public and private key pair and a certificate in the secure space, and requires the user to enter the personal identification number (PIN) on the TUI to protect the generated key data.

When using secure keys, the user's identity is authenticated on the TUI first, and then the secure element signs the transaction request of the user with the private key generated during the enabling process. When processing the transaction request, the bank verifies and signs the transaction.

When a user deregisters (disables) secure keys, the system directly destroys the public and private key pair stored in the secure element. This operation is irreversible.

The private key is stored in the secure element throughout the entire lifecycle, from public and private certificate key generation to certificate destruction, and is therefore secure.

Viewing Secure Keys Apps

Secure keys can check app package names and signatures. Only official apps will appear on the management screen to avoid fake and malicious apps. One-click query and management of secure keys apps is allowed using the Huawei Wallet APK setting interface.

Secure Keys Switch

The operating system has a switch to avoid background programs and apps from maliciously invoking the bank certificate. Turning on or off the switch is the same as inserting or removing a traditional USB key. When the switch is turned off, no certificate-related business can be conducted. Therefore, secure keys can offer a similar experience of being able to control hardware security.

eID*

Electronic identity (eID) is an ID card app jointly developed by Huawei and the Third Research Institute of the Ministry of Public Security. eID can be used in the same way as a physical ID card in scenarios recognized by the Ministry of Public Security. It also applies for online identity authentication based on the encrypted information provided by the Ministry of Public Security without disclosing the plaintext ID card information of users. In addition, users can swipe eID for public transportation services if the card reader supports this function. eID also provides authentication interfaces for other third-party mobile phone apps, offering quick and trusted identity authentication.

Users can set up eID in the HUAWEI Wallet app. During this process, they can use the mobile phone NFC to read the physical ID card and enroll facial information. After liveness detection is performed, the mobile phone encrypts the face image and uploads it to the server of the Ministry of Public Security. Once it is verified, the server delivers the eID information to the mobile phone. The collection and encryption of face images are implemented in the iTrustee secure OS, ensuring data security. After eID is set up, the eID information delivered by the server of the Ministry of Public Security is stored in the inSE and is accessible only to specific programs. Any intermediate data, such as face images, will be deleted from the mobile phone after the process is complete.

Huawei mobile devices comply with eID standards and specifications throughout the process, provide full lifecycle management of eIDs, and provide convenient and secure network digital identity services for users. Huawei's eID solution leverages the inSE, security camera, and iTrustee secure OS, thereby providing end-to-end high security protection while users set up, download, use, and deregister eID.

Car Key

Huawei mobile phones support car keys that comply with the Digital Key Specification released by the Car Connectivity Consortium (CCC).

After a car key is set up, users can use an NFC-equipped mobile phone for operations such as opening the car door and starting the engine.

Users can also use the app provided by the car manufacturer to share the car key with their relatives and friends. After being authorized by the car owner, users can download the car's digital key at any time and start the engine. Car owners can withdraw the authorization at any time.

When the car owner sets up the car key in their mobile phone, the EMUI TSM establishes a SCP channel with the SE to create a trusted, independent, and secure running space.

Then, the car owner can request the car manufacturer to help download the car's digital key to the mobile phone by using a TSM. This turns the mobile phone into a car key.

The car's digital key is stored in an industry-certified and recognized independent SE, which has financial-level security.

After a user restores factory settings, the mobile phone automatically disables and deletes the car key, protecting the user's property.

SMS Verification Code Protection*

SMS verification codes have become an important authentication factor for mobile apps. However, if an SMS verification code is intercepted, the user is faced with information breach or economic loss risks. To minimize such risks, EMUI protects SMS verification codes against text message interception by malicious apps.

EMUI has an additional intelligent identification engine for SMS verification codes at the system layer. After identifying an SMS verification code, the engine sends the text message only to the default SMS client set in the EMUI system. If the default SMS client is EMUI's built-in SMS client, the SMS client encrypts the text message with the verification code and filters access to the message. This prevents third-party SMS clients or apps from accessing the message. Even if a third-party SMS client or app directly accesses the SMS database, text messages with verification codes are encrypted, and the client or app cannot decrypt them.

*Note: This function takes effect only when EMUI's built-in SMS client is set as the default SMS client.

10 Internet Cloud Service Security

Huawei has established a series of powerful cloud services to help users use devices more effectively. In terms of design, these Internet services inherit the security objectives promoted by EMUI across the entire platform. Cloud services protect users' personal data stored on the Internet or transferred over the network, defend against threats and network attacks, and prevent malicious or unauthorized access to such information and services. Huawei cloud services use a unified security architecture that ensures user data security without affecting the overall usability of EMUI.

HUAWEI ID

A HUAWEI ID can be used to access all Huawei services, such as MyCloud, AppGallery, HiGame, Huawei Video, and Huawei Music. Ensuring the security of HUAWEI ID and preventing unauthorized access to user accounts are important concerns for users. To achieve this goal, Huawei requires users to use a strong password that is not commonly used and that contains at least eight characters in the form of lowercase and uppercase letters and digits. On this basis, users can add characters and punctuation marks (the maximum password length is 32 characters) to make the password stronger and therefore more secure.

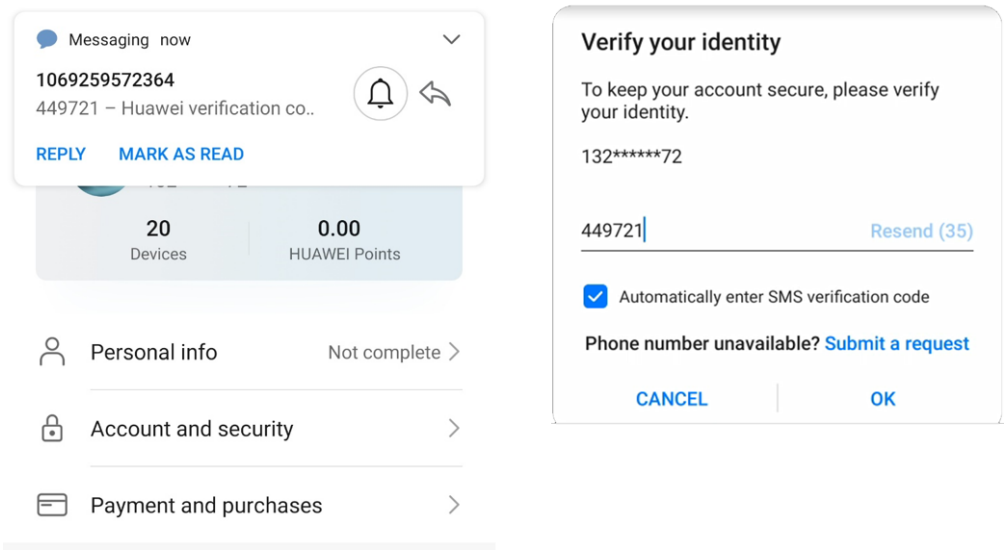
In cases where a user requests a major change to a HUAWEI ID, for example, when the user changes the password or uses the HUAWEI ID on a new device, Huawei will send a text message, email, or notification to the user. If any exception occurs, Huawei will prompt users to immediately change their passwords. Huawei has also adopted various policies and procedures to protect users' HUAWEI IDs. These policies and procedures include limiting the numbers of login and password reset attempts, continuously monitoring fraudulent activities for attack identification, and regularly reviewing existing policies for timely update according to new information that may affect user security.

Account Protection

Two-Factor Authentication

Two-factor authentication is the optimal account protection solution and ensures that the use of HUAWEI IDs is more secure.

Figure 10-1 Account protection

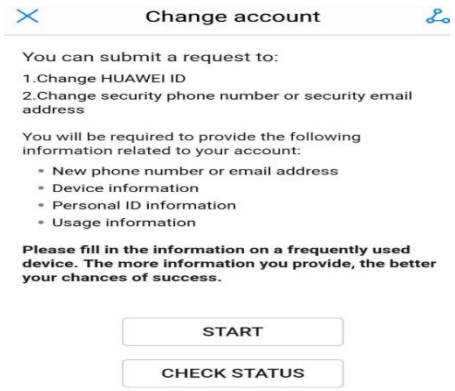


Account protection allows users to log in to their HUAWEI IDs using only their trusted devices. When attempting to log in from a new device, the user must enter the HUAWEI ID password and security verification code, which is automatically sent to the user's trusted phone number or displayed on the user's trusted device. If the new device passes verification, it will become the user's trusted device. For example, if a user has a HUAWEI P40 and wants to log in on a new HUAWEI MatePad Pro, the HUAWEI MatePad Pro will require the user to enter the HUAWEI ID password and the verification code displayed on the HUAWEI P40. This approach helps to enhance the security of HUAWEI IDs and associated HUAWEI ID services (such as MyCloud, AppGallery, Wallet, and HiGame).

Sliding Verification Code

When users log in, reset passwords, or change accounts through WAP and a browser, automated attacks must be prevented. Sliding verification codes are provided to prevent such attacks.

Heuristic Security Authentication



Users can change their phone number, email address, security phone number, or security email address through self-service means if they forget their HUAWEI ID password, want to reset the password, or the phone number or email address bound to the HUAWEI ID is no longer available.

Account Risk Control

Huawei provides an end-to-end risk identification mechanism and confrontation capabilities throughout the lifecycle of accounts. Risk prevention is provided across the entire process of account registration, login, service access, service operation, password reset, and account change. The system identifies fake accounts based on experts' rules, machine learning, and various means such as account operation exceptions, phone number exceptions, email exceptions, network risks, and geographical locations, to prevent malicious attacks on accounts and ensure the security of user assets and data.

HUAWEI ID Message

The HUAWEI ID message function allows Huawei device users to send and receive messages. This function supports texts and attachments, such as photos, contact information, and location information. The information is displayed on all of a user's registered devices so that the user can continue the dialog on any of the devices. Huawei does not record users' information or attachments. In addition, the content is end-to-end encrypted.

MyCloud

MyCloud allows users to store contacts, messages, photo albums, call records, reminders, calendars, browser bookmarks, and other contents, and synchronizes information between the users' devices. Users can log in to their HUAWEI ID to set MyCloud and choose services as required.

When users log out of their HUAWEI ID, related authentication information will be deleted. After obtaining user confirmation, MyCloud will delete all related data to ensure that personal data is not stored on unused devices. Users can log in to their HUAWEI ID on a new authenticated device to restore the MyCloud data.

HUAWEI ID-based Key

Each MyCloud file is divided into different blocks. Each block is encrypted or decrypted using AES128. MyCloud encryption and decryption require HUAWEI ID login. After a user successfully logs in to a HUAWEI ID, MyCloud derives an encryption factor for the HUAWEI ID and sends the factor and block metadata to the hardware encryption and decryption system. MyCloud files are encrypted and decrypted in this system and then sent to

the user's device through secure transmission channels. When stored on MyCloud, the user's data is protected through the key bound to the HUAWEI ID. This means that only the user can read and write the data.

MyCloud Backup

MyCloud backup backs up data (including device settings, app data, photos, and videos on the device) to the cloud only when user devices can access the Internet through Wi-Fi. MyCloud will encrypt and protect backup data.

11 Device Management

This chapter describes the device management function of EMUI. For enterprise users, EMUI provides the MDM function for device configuration and access control. For scenarios where a user loses a mobile phone, EMUI provides functions such as Find My Phone and Activation Lock.

Find My Phone and Activation Lock*

EMUI provides the Find My Phone function. After enabling the function, the user can:

- Locate (including active positioning and automatic location reporting at a lower battery level) a lost mobile phone.
- Ring the phone.
- Lock the phone (including locking the screen, reporting locations and movement tracks, and enabling the power saving mode automatically).
- Erase device data to ensure device data security.

To implement these operations, the user can log in to the cloud service website (cloud.huawei.com) or use the Find My Phone function on another Huawei phone.

EMUI also provides the activation lock function. Enabling Find My Phone will automatically enable activation lock. If an unauthorized user attempts to forcibly erase data from a lost phone, the user is required to log in to the HUAWEI ID to re-activate the phone after it is rebooted. This function enhances phone security by preventing unauthorized users from activating or using the phone.

Uses can choose to unlock activation lock with the lock screen password, if set in the Activate Device page. After the lock screen password is verified, subsequent unlocking operations are performed remotely in the cloud in the same manner as when the activation lock is unlocked by using HUAWEI ID account and password.

*Note: These functions are not supported in the following countries and regions outside China: Iran, Azerbaijan, Belarus, Georgia, Kazakhstan, Kyrgyzstan, Mongolia, Tajikistan, Turkmenistan, Ukraine, Uzbekistan, the United States, Japan, Saint Vincent and the Grenadines, Canada, South Korea, Saint Pierre and Miquelon, Aland Islands, Svalbard, Jan Mayen, Sint Maarten, Guernsey, Caroline Islands, Isle of Man, Jersey, Collectivity of Saint Martin, and Syria

MDM API

EMUI opens up the device management interface SDK of Huawei mobile devices to third parties through the Huawei Developer platform, allowing device configuration and access control for Enterprise Mobility Management (EMM) vendors and application developers. For details about the SDK, visit the Huawei Developer official website: https://developer.huawei.com/consumer/en/doc/system-Guides/3210801

For device management APIs required by enterprise mobile office customers, EMUI grants the customers corresponding use permissions by using the certificate. The enterprise customers can apply for the use permission of device management APIs from Huawei Developer official website.

Huawei issues device management certificates to app developers qualified by Huawei through Huawei open platforms. After the developer integrates the certificate into the developed Android package (APK), the APK can use the authorized APIs on Huawei devices.

When a user installs an APK that has a device management certificate, EMUI analyzes and verifies the certificate. If the certificate passes the verification, the APK obtains all permissions. If the certificate fails the verification, the APK will not have the permission. As a result, invoking the device management APIs fails and a security exception is displayed to ensure security of Huawei devices.

12 Privacy Protection

Huawei believes that privacy is your fundamental right, and you have full control over your privacy. Privacy protection is the cornerstone of product design and is complementary to service experience.

Huawei rigorously adheres to four foundational principles for privacy protection:

- **Transparency:** The way we process personal data has been made transparent. Decisions related to your privacy data are always made by you.
- **User benefit**: Personal data is used for the purpose of optimizing your user experience. And naturally, it is only collected with your consent.
- **Security**: We make use of cutting-edge technology to safeguard your personal data, including techniques such as Differential Privacy, so that user experience is enhanced while privacy protections are bolstered.
- **Legal compliance:** We strictly abide by all privacy-related laws and regulations, in all relevant jurisdictions, and in every respect, starting from product design, development, support service to other related business. Regardless of where you are, we will respect your privacy, and protect it with the utmost care.

This chapter describes EMUI's user privacy protection. Huawei devices may contain user privacy data, such as contacts, short messages, and photos. To protect user privacy, EMUI ensures that pre-installed apps fully meet privacy compliance requirements, and provides app permission management, notification management, location-based service (LBS), and other privacy management functions. To further protect users' privacy, EMUI provides the device identifier system, differential privacy, and other technical privacy protection means.

Permission Management

The EMUI system provides a permission management mechanism designed to allow or restrict apps' access to APIs and resources. By default, no permissions are granted to apps, and access to protected APIs or resources is restricted to ensure security of such APIs and resources. During installation or running, apps request permissions, and users determine whether to grant the permissions. EMUI enables users to allow or deny permissions to an installed app for fine-grained control. The permission management function applies to the following:

- Phone
- SMS
- Contacts
- Call record

- Camera
- Location information
- Microphone
- Calendar
- Body sensor
- Health and fitness
- Storage
- MMS
- Using call transfer (CT)
- Suspended window
- Creating desktop shortcut

Audio/Video Recording Reminder

To prevent malicious apps from obtaining permission to access the microphone or camera through spoofing and recording audio or videos at the backend without users' knowledge to steal users' privacy data, EMUI provides the audio/video recording reminder function. When an app is using a microphone or camera, the system displays a prompt on the notification bar. When the user touches the prompt, the app interface or the app's permission management interface is displayed. The user can also touch the close button to close the app that is recording audio or a video.

Allow Once

In EMUI 11.0, **Allow once** is added to the pop-up window for granting permissions to the camera, microphone, and location. When an app applies for a permission and a user selects **Allow once**, the app can use the permission just this one time. Once the app stops running, the system revokes the permission. The app needs to apply again the next time it wants the relevant permissions.

Location Access

EMUI allows a user to enable or disable location access in **Settings**. After location access is disabled, EMUI also disables the Global Positioning System (GPS), Wi-Fi, Bluetooth, and mobile tower positioning. In this way, users' location positioning is completely disabled, ensuring user privacy.

If an app requires access to location information through LBS, it needs to apply for the location access permission. The user can determine whether to grant the permission (Allow, Always allow, or Deny) to the app based on the application scenario. If the user selects "Allow", the app can access location information but not at the backend. If the user selects "Always allow", the app can access location information during running and at the backend. If the user selects "Deny", the app cannot access location information.

When the user selects "Always allow", the system detects that the app is accessing location information at the backend and will periodically ask the user whether to allow backend access through notification. The system notifies the user only once for each app.

Device Identifier

During system processing, unique identification is required. EMUI provides multiple unique identifiers with different behavior features. The app selects the most appropriate identifier based on different scenarios. These features involve privacy.

Scope

EMUI identifiers have three scopes. Wider scope of an identifier indicates higher risk of being tracked.

- Single App: The ID is only available to the app and cannot be accessed to any other apps.
- App group: The ID is available to a group of apps, such as a group of apps provided by the same app developer.
- Device: All apps installed on the device access a same ID.

Resettability and Durability

The resettability and durability define the lifecycle of identifiers. The longer an identifier is stored, the more vulnerable the user is to long-term tracking. When the app is reinstalled or the identifier is manually reset, the duration is shortened and the risk of being tracked is reduced.

To prevent apps from using device identifiers to track users, EMUI prohibits third-party apps from obtaining permanent device identifiers, such as IMEI, SN, and MAC address.

The EMUI identifier system includes:

ID Type	ID Name	Application Scenario & Scope	Generation Time	Resettability
Random identifier	UUID	Used in scenarios where apps are associated with random identifiers.	A random number is generated each time an identifier is invoked.	The UUID is regenerated each time an identifier is invoked.
User ID	HUAWEI ID	Used for Huawei Cloud service features, AppGallery, MyCloud, Huawei Music, etc.	Generated upon creation of a HUAWEI ID.	Deleted upon deregistration of a HUAWEI ID.
	Open HUAWEI ID	Used by third-party apps to log in to Huawei Cloud service features, AppGallery, MyCloud, Huawei Music, etc.	Generated upon creation of a HUAWEI ID.	Deleted upon deregistration of a HUAWEI ID.
Device ID	Open device identifier (ODID)	Provided for Huawei Developer to prevent data from being correlated between multiple third-party vendors. ODIDs are	The Developer ID is randomly generated during app installation. A different ID is generated each	The ID is regenerated when the app is reinstalled.

ID Type	ID Name	Application Scenario & Scope	Generation Time	Resettability
		assigned based on third-party app signatures.	time the app is installed.	
	Open anonymous identifier (OAID)	Provided for advertisers in advertisement placement scenarios.	Generated when the system is started for the first time or manually reset.	Users can manually reset the OAID.

Differential Privacy

To provide users with a reliable, stable, and energy-efficient software and hardware system that delivers the ultimate experience, Huawei will collect statistical data on the reliability, performance, power consumption, faults, and errors on user devices, as well as data on how user devices and apps are used. User data is sent to Huawei only after users' explicit consent is obtained. EMUI uses differential privacy technology in the User Experience Improvement Program to enhance user experience while protecting the data users share with Huawei. This technology adds random noises to the data so that the real data cannot be recognized. Relevant statistics will only appear if combined with the data from a large number of other users and the randomly added noises average out.

Privacy Statement

EMUI provides an explicit privacy statement and explicitly notifies users to check and confirm the statement in the startup wizard. In addition, users can check the privacy statement in **Settings**. Privacy policies vary in different countries. Therefore, users in different countries are provided with specific privacy statements on EMUI released in the local countries.

Refer to the privacy statement as follows:

https://consumer.huawei.com/en/legal/privacy-policy/worldwide/

13 Security Standards Compliance and Certification

This chapter describes the security standards that Huawei devices comply with and the security certifications obtained in terms of hardware and chips, EMUI platform software, and app software.

Security Standards Compliance

- International cryptographic algorithm standards:

  Cryptographic algorithms complying with the ISO/IEC standards (such as ISO/IEC 18033 series)
- Standards compliance of devices:
  - CCRC-EAL-TR-019-2019 Security Technical Requirements for Operating Systems of Mobile Intelligent Terminals (Evaluation Assurance Level 4 Enhanced)
  - Mobile Device Fundamentals Protection Profile Version 3.1
- Protocol compliance of HUAWEI ID authentication: OAuth 2.0

Security Certification*

Huawei EMUI 11.0 has obtained the following certifications:

Certification Name	Certified Object	Issuing Authority	Description
CC EAL 5+	Microkernel	Netherlands NSCIB	Security assessment of TEE secure OS microkernel
CC EAL2+	iTrustee 5.0 TEE	Netherlands NSCIB	Security assessment of TEE OSs
IT product information security EAL4+	EMUI	China Cybersecurity Review Technology and Certification Center	Security assessment of mobile smart device OS in China

Certification Name	Certified Object	Issuing Authority	Description
ePrivacySeal	EMUI (User Experience Improvement Program – UE, OTA, and AI Voice Celia)	ePrivacy	Privacy assessment of feature compliance with European privacy laws and technical implementation
ISO/IEC 27701	Huawei device software of Huawei Device Co. Ltd.	British Standard Institute (BSI)	Privacy compliance assessment of the establishment, implementation, and maintenance of the corporate privacy information management system
Mobile Finance Technical Service Certification	Kirin 990/980/970 project	China Financial Authentication	Security assessment of security modules of Kirin chips in China

*Note: For details about the security certifications obtained, visit https://consumer.huawei.com/en/privacy/certification/.

14 Digital Copyright Protection

Digital Rights Management (DRM) refers to the technologies that publishers use to control the use of protected objects, such as digital contents (e.g., software, music, and movies) and hardware, and to limit the use of an instance of digital products. DRM is a new technology developed with the explosion of electronic audio and video programs on the Internet. It aims to protect the copyright of digital media, technically prevent illegal copying, or to some extent make copying difficult, where users must be authorized to use digital media.

Currently, DRM includes Widevine, PlayReady, and Verimatrix. Security can be ensured by software, hardware, or enhanced hardware.

Huawei has established ChinaDRM2.0, a proprietary end-to-end DRM solution based on the ChinaDRM technical standard system and the SoC chip certified at the world's highest DRM security level. Huawei's ChinaDRM2.0 has obtained the ChinaDRM ecosystem certification, providing developers with free, secure, and trusted digital copyright protection capabilities on Huawei devices.

ChinaDRM 2.0

Huawei devices provide DRM Kit (ChinaDRM standard), allowing third-party apps (from Baidu, Alibaba, and Tencent in China, and Netflix and Amazon outside China) to build their content copyright protection capabilities based on Huawei device ecosystem.

Huawei devices use the standard Android version and standard Android APIs. All services (except reference apps) run in the vendor environment and TEE. The following figure shows the overall security architecture.

ChinaDRM is integrated into the Android version as a standard DRM component. The ChinaDRM Plugin interconnects with the standard Android DRM framework, which is the adaptation layer for integrating the ChinaDRM into the Android framework. This module calls the ChinaDRM CA module through the ChinaDRM Function API to interact with the ChinaDRM TA, thereby implementing DRM. The ChinaDRM CA interface is an interface layer running in the REE. It is used to communicate with the TEE, process service functions with low security requirements, implement the ChinaDRM Function API, and forward the DRM request of the ChinaDRM Plugin to the ChinaDRM TA. The ChinaDRM TA is a functional TA responsible for ChinaDRM services in the TEE. It implements online certificate download, license parsing, policy processing, and code stream decryption. The unified ChinaDRM client is used for license parsing. The ChinaDRM Key TA burns keys into chips and manages ChinaDRM certificates and keys. It provides the key operation interface.

15 Conclusion

Huawei attaches great importance to users' device security and privacy, and has designed EMUI to provide end-to-end (from underlying chips and systems to apps) security protection capabilities. EMUI constructs a trusted basic architecture for the device based on the chip hardware, and constructs security experience that balance both security and user experience based on enhanced security and strong computing performance of the device hardware.

At the system layer, EMUI improves system security by enhancing kernel security. Based on underlying trusted platform and system security hardening, it provides more secure system control capabilities for the upper layer. At the app layer, EMUI provides app signature, app sandbox, permission management, threat detection, and other functions, and works together with the cloud to ensure security.

While providing security solutions, Huawei also attaches great importance to establishing security process and capabilities, which are vital for implementing security management of products throughout the lifecycle.

Huawei has set up a computer emergency response team (CERT) dedicated to improving product security. Any organization or individual that finds security vulnerabilities in Huawei products can contact Huawei at PSIRT@huawei.com. Huawei PSIRT will reply promptly while organizing internal vulnerability fixing, releasing vulnerability warning, and pushing patches for update. Huawei is sincere in its willingness to jointly build Huawei device security with all stakeholders.

16 Acronyms and Abbreviations

Table 16-1 Acronyms and Abbreviations

Acronym/Abbreviation	Full Name
2D	Two Dimensions
3D	Three Dimensions
3DES	Triple Data Encryption Standard
AES	Advanced Encryption Standard
AI	Artificial Intelligence
API	Application Programming Interface
APK	Android Package
ARM	Advanced RISC Machines
ASLR	Address Space Layout Randomization
BLE	Bluetooth Low Energy
BYOD	Bring Your Own Device
CA	Certificate Authority
CE	Credential Encryption
CERT	Computer Emergency Response Team
CFI	Control Flow Integrity
CNN	Convolutional Neural Network
DE	Device Encryption
DEP	Data Execution Prevention
CMP	Certificate Management Protocol
DAC	Discretionary Access Control
DRM	Digital Rights Management

Acronym/Abbreviation	Full Name
ECB	Electronic Code Book
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic Curve Digital Signature Algorithm
EAP	Extensible Authentication Protocol
EIMA	EMUI Integrity Measurement Architecture
EMM	Enterprise Mobility Management
eMMC	Embedded Multimedia Card
GP	GlobalPlatform
GSM	Global System for Mobile Communications
HKIP	Huawei Kernel Integrity Protection
HMAC	Hash-based message Authentication Code
HOTA	Huawei Over The Air
HUK	Hardware Unique Key
HUKS	Huawei Universal Keystore
ID	Identifier
IMEI	International Mobile Equipment Identity
InSE	Integrated Secure Element
IOT	Internet of Things
IPsec	Internet Protocol Security
IT	Information Technology
JOP	Jump Oriented Programming
L2TP	Layer Two Tunneling Protocol
LKM	Loadable Kernel Module
LSM	Linux Security Module
LTE	Long-Term Evolution
LTO	Link Time Optimization
MAC	Mandatory Access Control
MAC	Media Access Control
MDM	Mobile Device Management
MPPE	Microsoft Point-to-Point Encryption
NFC	Near Field Communication

Acronym/Abbreviation	Full Name
NIST	National Institute of Standards and Technology
NPU	Neural Processing Unit
OS	Operating System
OTA	Over The Air
P2P	Peer to Peer
PAN	Privileged Access Never
PIN	Personal Identification Number
PKI	Public Key Infrastructure
POS	Point of Sales
PPTP	Point-to-Point Tunneling Protocol
PRNG	Pseudo-Random Number Generator
PSK	Pre-Shared Key
PXN	Privileged Execute Never
REE	Rich Execution Environment
ROM	Read-Only Memory
ROP	Return Oriented Programming
RSA	Rivest Shamir Adleman
RPMB	Replay Protected Memory Block
SCEP	Simple Certificate Enrollment Protocol
SD	Secure Digital Memory Card
SDK	Software Development Kit
SELinux	Security-Enhanced Linux
SFS	Secure File System
SHA	Secure Hash Algorithm
SN	Serial Number
SOTER	Standard Of auThentication with fingERprint
SSL	Security Sockets Layer
TA	Trusted Application
TEE	Trusted Execution Environment
TLS	Transport Layer Security
TUI	Trusted User Interface

Acronym/Abbreviation	Full Name
UDID	Unique Device Identifier
UID	User Identifier
UUID	Universally Unique Identifier
VPN	Virtual Private Network
WAPI	WLAN Authentication and Privacy Infrastructure
WEP	Wired Equivalent Privacy
WLAN	Wireless Local Area Network
WPA	Wi-Fi Protected Access
WPS	Wi-Fi Protected Setup

Change History

Date	Description
2020-9-10	Updates for EMUI 11.0: • System security • Data security • Distributed security • Privacy protection
2020-5-11	Updates for EMUI 10.1: Added "Distributed Security".
2019-8-30	Updates for EMUI 10.0: • Hardware security • System security • TEE • App security hardening • Device interconnection security
2018-10-31	Updates for EMUI 9.0: • Facial recognition • eID • Password vault • AI security protection • HUAWEI ID • AI application • Differential privacy
2017-10-31	Updates for EMUI 8.0: • HKIP • File system encryption key protection • SD card encryption • Secure input function enhancement • Device interconnection security • Secure keys • Payment protection center optimization • Code scanning login • Find My Phone function enhancement • Privacy space function enhancement
2017-05-31	Initial release