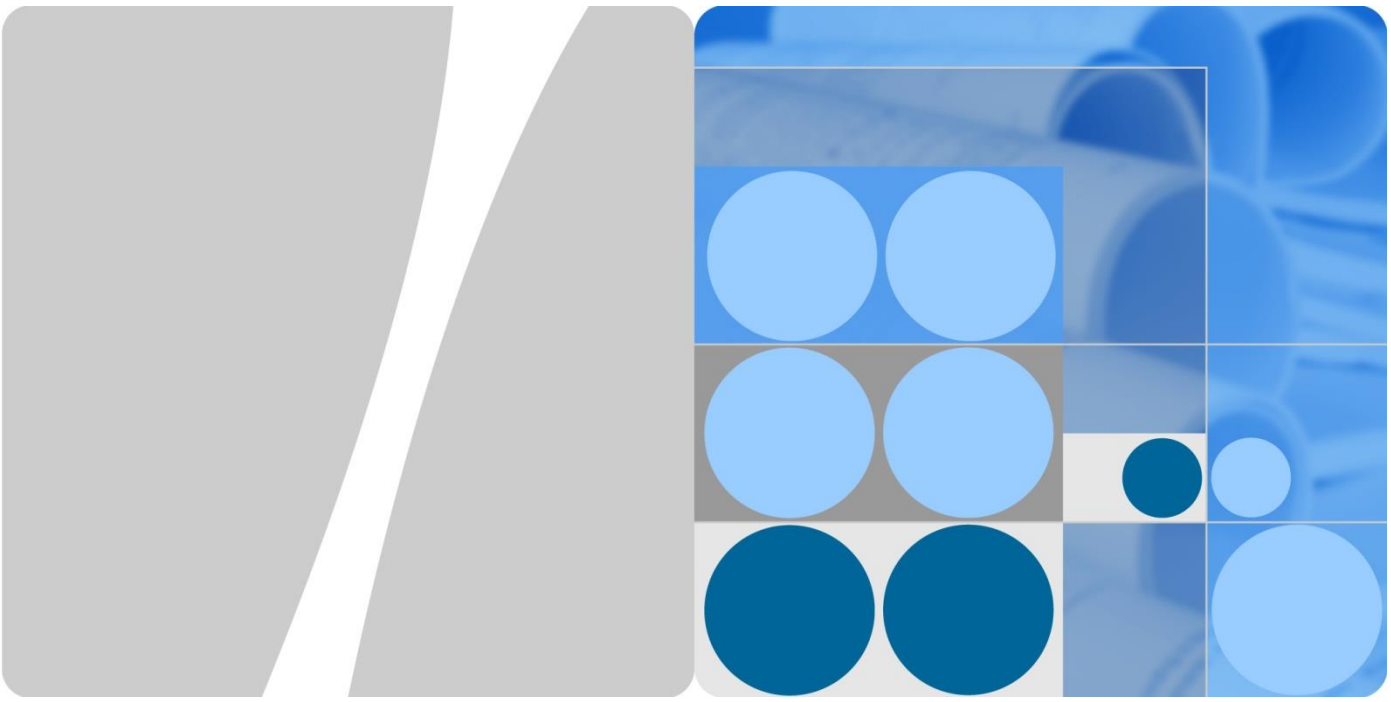


EMUI 10.0 Security Technical White Paper

Issue V1.0

Date 2019-08-30

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

As mobile Internet continues to develop, mobile smart devices have become primary network access devices, which store large amounts of user data including personal user information. In addition, increasing numbers of apps originating from unverifiable sources are installed on these devices. As a result, privacy and security are ever-present concerns.

Mobile apps can come from various channels, with some arriving pre-installed by vendors while others can be acquired from third parties. Through this process, it is possible for users to download malicious apps, which may infringe upon users' privacy or steal users' property, presenting a wide range of potential security risks.

Huawei attaches a great deal of importance to the security of mobile smart devices, providing comprehensive security assurance while also delivering a high-level user experience. This white paper systematically describes the security and privacy protection solutions offered by the Emotion UI (EMUI).

The EMUI is a Huawei-developed mobile device operating system for smart mobile apps. It 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.

Security is a systematic project. The 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.

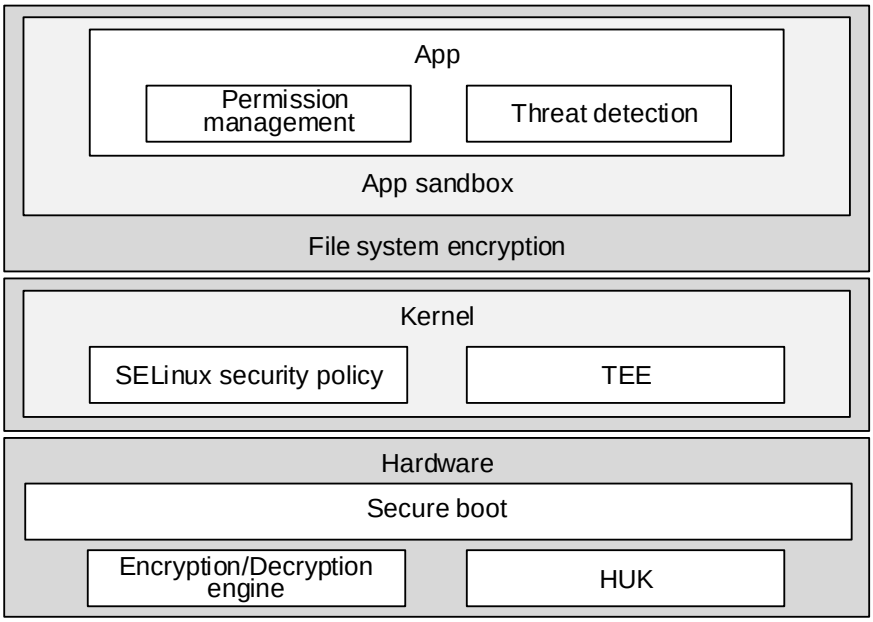
The EMUI provides a secure boot mechanism from underlying hardware chips to prevent the EMUI read-only memory (ROM) image from being tampered with. The ROM image can only run on a device after passing signature verification. This ensures secure boot for the bootloader, recovery, and kernel images, and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

To ensure data security, the 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 directories of the corresponding apps, meaning files from one app cannot be accessed by another. The data erasure function is provided to permanently erase data during device recycling or factory restoration, in order to prevent unauthorized data restoration. The 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, the EMUI pre-installs Phone Manager to provide virus scanning, block and filter, traffic management, notification management, and other functions. Utilizing these

functions, the 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



This document describes the security technologies and functions of the EMUI 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, secure storage, and trusted UI (TUI)
- TEE: secure OS, security capability, and security capability openness
- 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, malicious website detection, HiAIKit, and HiHealth Kit
- Network and communication security: virtual private network (VPN), Transport Layer Security (TLS), Wi-Fi security, protection against fake towers, and device interconnection security
- 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

2 Hardware Security

The 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
- TUI

Secure Boot

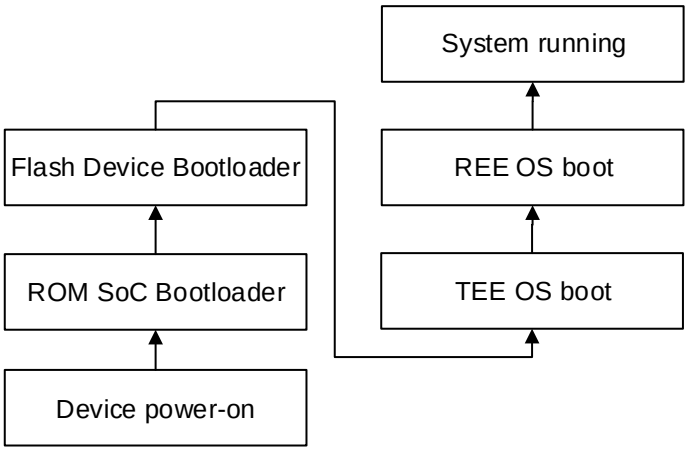
Secure boot prevents the loading and running of unauthorized apps during boot. The boot program uses a public key to verify the digital signatures of software, ensuring trustworthiness and integrity. Only image files that pass the signature verification can be loaded. These files include bootloader, 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 (fuse blowout) 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, thereby ensuring a trust chain transfer and preventing unauthorized programs from being loaded during the boot process.

The images used during some boot processes are encrypted.

Figure 2-1 Secure boot



Hardware Encryption/Decryption Engine and RNG

To meet the requirements of high-performance encryption/decryption and key protection, the 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 10.0. 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 10.0 to derive the same key for the same type of devices. It is applied to image encryption and other functions.

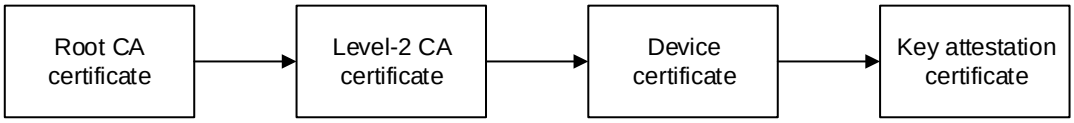
Device Attestation

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 the TEE of the 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 the TEE 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 10.0, 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.

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.

The 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 the TEE and bound with the keys derived by the HUK. Only the TEE 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.

TUI*

In app environments in the REE, the displayed payment amounts or input passwords may be hijacked by malicious apps. The iTrustee® secure OS provides the TUI display technology (compliant with GP standards) that can disable 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.

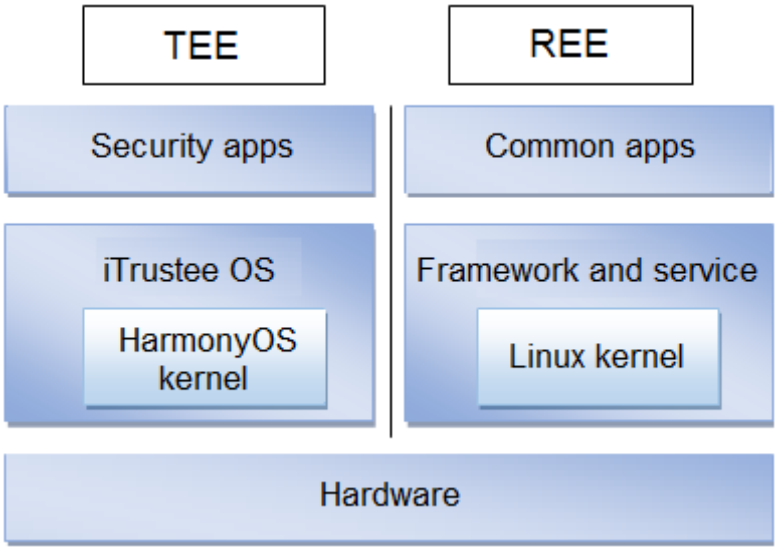
3 TEE

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 Introduction

The iTrustee 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. The Huawei iTrustee secure OS utilizes HarmonyOS microkernel technology and improves TEE kernel security using a formal method.

The HarmonyOS microkernel architecture 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 and maximum isolation improve system performance and reduce the attack surface. In addition, fine-grained permission design is enhanced to provide more powerful security features and lower latency. Formal verification is an effective means to verify system correctness (without vulnerabilities) from the source by using mathematical theorems. The iTrustee first uses formal verification to significantly improve the system security level. The correctness of core modules, core APIs, and high-level mechanisms such as process isolation and permission management is verified to prevent data race and memory access errors, thereby rebuilding trustworthiness and security. See the following figure.



The iTrustee 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 prevent video theft and interception 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. The 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). The iTrustee protects user input information against theft from malicious programs.
Application logic protection	The iTrustee protects critical application logic from being stolen or tampered with.

As an example, the 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).

Security Capability

The operation memory of the REE and TEE uses Advanced RISC Machine (ARM)-based TrustZone technology to implement hardware isolation.

The iTrustee uses kill-chain-based technical means for security defense, such as anti-reversing, anti-intrusion, and anti-attack, to comprehensively reinforce system security, channel security, and authentication security in the REE, as well as system security in the TEE. For example,

anti-reversing is used to prevent attacks in advance in the intrusion preparation phase by encrypting images. Image encryption is enabled in the chip delivery phase to prevent reverse attacks. Anti-intrusion encrypts authentication information and strictly authenticates CA-TA 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.

The 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. In addition, the formal design methodology is introduced to identify security risks and issues in mathematical and logical ways.

The iTrustee supports the following basic security capabilities:

- **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. Data is stored securely to prevent rollback. Trusted storage of the 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:** The 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 Chinese national 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 the 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:** The iTrustee provides secure display and input to prevent malicious programs from viewing information on a screen or accessing a touchscreen. This function protects passwords or PINs when they are entered, and prevents the exposure of credentials to malicious programs. In addition, apps display security messages to users to prevent them from being snooped on. Basic controls such as PNG images, texts, buttons, and text entry boxes are supported. Chinese characters, English letters, symbols, and digits are displayed in the same size. The UI can be customized, keypad keys are randomized, and various controls and window management are supported. In addition, the UI style is consistent with the EMUI style of devices.

  The TUI feature ensures that the information displayed to users is not intercepted, modified, or covered by any software in the REE or unauthorized apps in the TEE. The TEE sets display parts to a secure state in which only the TEE can access the display parts. Displayed information is not transferred to the REE, and permission control is used to ensure that only authorized TEE apps can access the information. After the TUI is displayed, preset images or texts are displayed to indicate the secure display and input state.
- **Trusted time:** The iTrustee provides trusted reference time, which cannot be modified by malicious TA or REE apps.

The 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 the 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 the iTrustee.
- **Device security service:** Unique device identifiers are provided, as well as REE health status information, and more.

Third-party app developers can develop and debug TAs based on the iTrustee's security capabilities.

Capability Openness

The iTrustee provides TEE platform capabilities for developers and a unified developer UI. The TEE capability framework is standardized to allow both Huawei and third-party apps to make use of Huawei iTrustee security capabilities. Apps are developed once and deployed on multiple devices, supporting the construction of a distributed device security ecosystem.

The iTrustee provides developers with diversified APIs, complete SDKs, and relevant reference manuals and reference design in the HUAWEI DevEco Studio development environment. It also provides security certificate management, app signature, security app lifecycle management, and application release services.

4 System Security

System security aims to ensure that EMUI devices make full use of the security capabilities of hardware chips, and provide an end-to-end security system that integrates software and hardware for apps running on the EMUI system and for consumers who use the EMUI system and apps. The EMUI builds system security capabilities primarily from the following aspects:

- Integrity protection: provides HKIP and an EMUI integrity measurement mechanism during system boot, upgrade, and operation.
- System kernel security: provides system resource isolation and a system access control mechanism.
- System anti-attack capability: provides security defense mechanisms for memory, including KASLR and Privileged Access Never/Privileged Execute Never (PAN/PXN).
- In addition to the identity of the system process identified in the system access control mechanism, the system also verifies the identity of the person who accesses an EMUI device.

Integrity Protection

Verified Boot

The EMUI supports the verified boot function. When a read-only 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, ensuring that the device status at startup is the same as when it was last used.

HKIP*

Although secure boot ensures the validity and integrity of software during startup, vulnerabilities in valid code may still be exploited by attackers.

HKIP uses the hypervisor mode (EL2) provided by the ARMv8 processor to protect the kernel and system registers, preventing the system from being tampered with or injected. This protects system integrity and prevents privilege escalation.

Building on multiple generations of technology iteration and optimization, 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, such as SELinux PolicyDB, cannot be tampered with.
- Critical system register settings cannot be tampered with.

EIMA

The EIMA measures 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 mainly 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 measurement subsystem collects the hash values of measured objects 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 evaluation system provides the integrity check result. Service-specific decision makers then determine subsequent handling policies.

Kernel Security

System Access Control Capability

The EMUI supports the SELinux feature. When a device is started, access control policies are loaded to the system kernel and cannot be dynamically changed. This feature applies mandatory access control (MAC) to all processes when they access resources such as directories, files, and device nodes, and applies root-capability-based mandatory access control 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.

KASLR

In a stack overflow attack, the attacker can bypass data execution prevention (DEP) technology by overwriting the return address of a function. In the exploit code targeted at this type of vulnerability, a specific jump address must be determined. Address space layout

randomization (ASLR) invalidates the return to a fixed address. KASLR allows the kernel image to be loaded to the VMALLOC area.

EMUI 10.0 supports the KASLR mechanism, allowing the address mapped to the kernel image to have an offset relative to the link address. The offset address is randomly generated by the bootloader upon each boot. As a result, the virtual address mapped to the kernel image varies with each boot. KASLR results in unpredictable address space layout, and makes it more difficult to launch code reuse attacks. This further enhances the security of the system kernel.

PAN/PXN

EMUI 10.0 supports PXN and PAN for security protection. These technologies prevent the kernel from executing user space code and accessing user space data.

Using some kernel attack methods, an attacker tampers with the code pointer of a kernel entry 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, defending against such attacks.

An attacker can also tamper with the data pointer of a kernel entry so that it points to the data structure prepared by the attacker in user mode, which launches an attack by affecting kernel behavior. PAN prevents the kernel from accessing user-mode data, thereby preventing such attacks.

A processor is required to support PXN and PAN. The ARM supports PAN from the 8.1 instruction set.

Control Flow Integrity (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 10.0 uses Clang CFI technology 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.

Identity Authentication

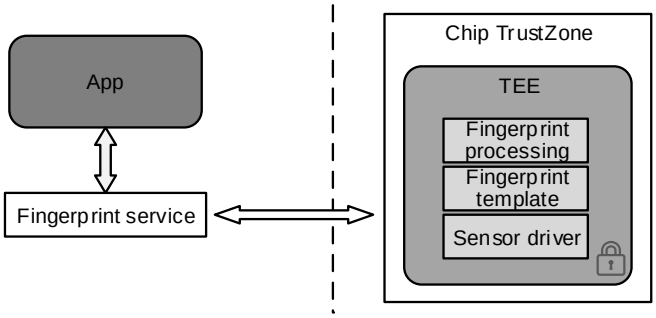
The EMUI provides two biometric identification capabilities: fingerprint recognition and facial recognition. That is, the EMUI uses the unique physiological features (fingerprint and facial features) to authenticate personal identities. These capabilities can be applied to identity authentication scenarios such as device unlocking, payment, and app login.

Fingerprint Recognition

The 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 the EMUI's fingerprint recognition security framework.

Figure 4-1 Fingerprint recognition security framework



The EMUI establishes a secure channel between a fingerprint sensor and the TEE. Fingerprint information is transmitted to the TEE through this secure channel, and the REE cannot obtain the information. The EMUI collects fingerprint image information, extracts features, detects live fingers, and compares features in the TEE, 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 TEE 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 the TEE. The EMUI does not send or back up any fingerprint data to any external storage media including the cloud.

The 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 three days,

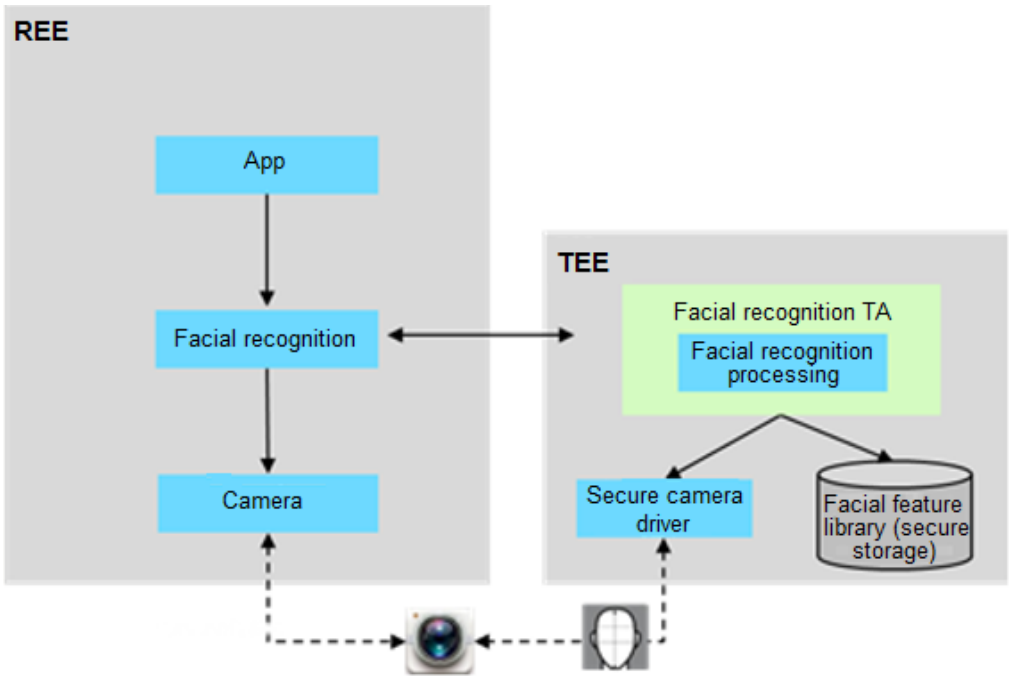
the user is compelled to enter the password to unlock the screen, in order to reduce the likelihood of a forgotten password.

Facial Recognition

The 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 the EMUI's facial recognition security framework.

Figure 4-2 Facial recognition security framework



The EMUI establishes a secure channel between the camera and TEE. Face image information is transmitted to the TEE through this secure channel, and the REE cannot obtain the information. The EMUI collects face images, extracts features, detects live faces, and compares features in the TEE, 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 TEE 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 the TEE. The EMUI does not send or back up facial data (either encrypted or unencrypted) to any external storage media including the cloud.

The 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.

System Software Update

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

In addition, the EMUI provides software update control. At the beginning of OTA upgrade and after a software package is downloaded, EMUI applies for upgrade authorization by sending the digest information of the device identifier, the version number and hash value of the upgrade 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.

The 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/.

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. The 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, the EMUI provides app developers with HUKS framework capabilities, enabling them to securely use keys to protect confidential data in apps.

Lock Screen Password Protection

The 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.

The 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 the TEE. 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.

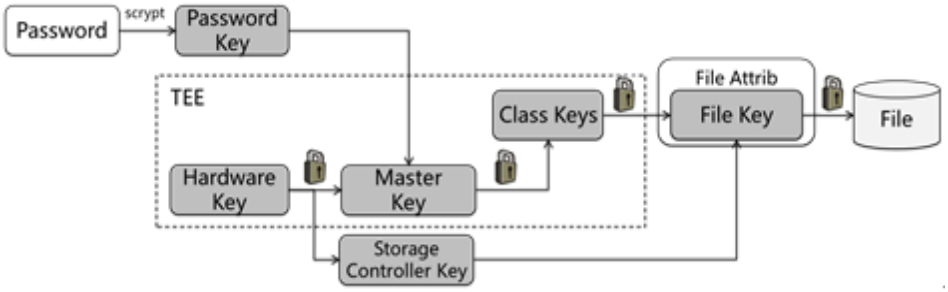
In cases where a mobile phone is lost, the EMUI provides data encryption protection for the user file system, preventing unauthorized users from launching physical attacks (for example, directly reading the flash memory) to obtain device data and cause user data breaches.

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 user data security and positive app experience, the storage area is divided into device encryption (DE), credential encryption (CE), and no encryption (NE) partitions: App data is stored in the CE partition by default to ensure app security.

- CE partition: CE class keys are protected using both the lock screen password and HUK, and can be created only after a device is powered on and the user unlocks the screen using the lock screen password. Such keys are not deleted when the device is locked. CE class keys can be used to protect data such as the gallery, contacts, messages, calendar, call records, and location information.
- DE partition: DE class keys are protected using the HUK and can be accessed after a device is powered on (even without screen unlock). DE class keys can be used to protect data related to WLAN authentication, Bluetooth pairing, alarm, ringtones, and more.
- NE partition: Data in this partition is not encrypted, which is required in rare cases. Such data includes OTA upgrade packages.

Figure 5-1 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.

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 system, 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 validation based on device certificates. The cloud server can authenticate the validity of EMUI devices through certificate authentication. In combination with biometric authentication, the HUKS can provide services such as login and payment with a TEE level of security for payment apps.

HUKS keys and certificates are stored in the TEE. 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 the TEE and also used in the calculation of encryption and decryption keys. A plaintext key is stored in the TEE, and encryption and decryption are protected by the TEE.

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 the TEE. 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.

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, the 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 available for mobile phones, 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 the TEE, and encryption/decryption is always performed in the TEE.

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 negotiation using the asymmetric key generated by two phones in the TEE. Key negotiation is performed in the TEE, 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 the 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.

The 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.
- The EMUI's HiAIKit and HiHealth Kit provide convenient AI and health API kit capabilities for app developers, and adopt corresponding security mechanisms to ensure the security of AI and health data used by developers.

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: scans keys, functions, algorithms, and more to identify common issues such as key leak, dangerous functions, and insecure algorithms. Filter criteria (such as suffix and type) are then set for refined control over scanned objects, in order to determine exact match locations and contexts and highlight matched contents.

App Signature

Only apps with complete signatures can be installed in the 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

The 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. The 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*

The 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. The 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.

The 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*

The 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. The 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.

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

Malicious Website Detection*

The 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, the 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.

HiAIKit

As the basis of smart capability sharing, the HiAIKit provides AI capabilities and a unified AI capability platform for device products. The HiAIKit provides basic AI capabilities such as unified language processing and machine vision processing. The EMUI builds a unified kit of smart capabilities to centrally manage and control AI capabilities. Huawei apps that use the kit capabilities are authenticated by signature. "signatureOrSystem" is added for permission authentication. Permission control is implemented to safeguard personal data and ensure data security. The open capability performs strict boundary verification on user input to prevent malicious attacks and ensure the AI capabilities run as normal.

Data Security

The HiAIKit processes a user's data only after being authorized by the user. User data is stored locally in the device using the sandbox mechanism, which effectively controls data access permissions. The AI model that processes users' personal data is encrypted for storage to ensure confidentiality and integrity. SELinux is used to minimize access permissions for the AI model. The Hypertext Transfer Protocol Secure (HTTPS) is used to transmit data from the device to the cloud server when the device and the cloud server collaborate to complete a requested AI service such as providing an AI capability. The data stored in the cloud service is also encrypted. These measures work to improve the security of data processing.

To improve data and privacy security, universally unique identifiers (UUIDs), software random identifiers irrelevant to hardware devices, are used when devices deal with sensitive services (such as voice services) and interact with third-party services. A UUID can change randomly based on service requirements. For example, when a user withdraws the consent to the privacy statement of a voice processing service, the lifecycle of the UUID ends, and the personal data stored in the cloud server and device will be deleted.

When a user uses HiVoice and enables the Voice Wakeup function, the device will not process the user's voice data flow before wakeup. When the user wakes up the device, the app uses voiceprint verification technology to ensure the consistency between the input voice and the voiceprint stored in the device. When the user has woken up the device and started the voice service, to prevent the device from being intercepted, the interaction ends if there is no voice input within **10s** after wakeup. If the user wants to continue using the voice service, the device needs to be woken up again. When a voice is detected, the user needs to input a voice command within **20s**. Otherwise, the device will not execute the command. This is to prevent the user's phone from being intercepted by a malicious party.

HiHealth Kit

If permitted by the user, the HiHealth Kit can provide fitness and health data for third-party apps or store fitness and health data from third-party apps in the phone. The HiHealth Kit also works directly with wearables or fitness devices, such as compatible heart rate monitors using Bluetooth Low Energy (BLE).

Data Security

The HiHealth Kit can provide fitness and health data for third-party apps, and can store fitness and health data from third-party apps after authorization from the user. The HiHealth Kit grants permissions separately to different third-party apps and different types of fitness and health data, and distinguishes read and write data permissions.

All fitness and health data stored in the HiHealth Kit is encrypted.

Third-Party App Access Security

Third-party apps can invoke the HiHealth Kit API only after being authorized by the user and must comply with the cooperation agreement signed with Huawei Health. For example, third-party apps must provide users with privacy statements. The HiHealth Kit verifies the package names and certificates of third-party apps to ensure that only authorized apps can access fitness and health data, thereby preventing malicious apps from spoofing legitimate apps and obtaining user data.

Users can control access to fitness and health data by third-party apps through the Data Sharing function of the Huawei Health app. Third-party apps must obtain separate read and write access permissions for each fitness or health data point. Users can view and withdraw these access permissions on the Data Share page of the Huawei Health app.

A third-party app that has obtained write permissions cannot read the data it writes unless it also obtains the read permission. If a third-party app obtains the read permission for a certain type of data, it can read data of this type written by all sources.

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 the 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. The 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

The 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 negotiation 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

The 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. The 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. 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. The 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.

Device Interconnection Security

EMUI Device Interconnection Security for the Same HUAWEI ID

The EMUI provides authentication services for devices that are logged in with the same HUAWEI ID. Each EMUI device logged in with a HUAWEI ID generates a public and private key pair using elliptic curve cryptography as the device identifier, and applies to the Huawei Cloud server to authenticate the public key. Devices with the same HUAWEI ID that have passed the authentication can be mutually authenticated as trusted devices in the device interconnection service, and exchange their authenticated public key credentials. As for public key credentials of the peer end, devices with the same HUAWEI ID can perform key negotiation and further secure communication. Devices not registered under this HUAWEI ID will not pass this authentication.

Devices that support Instant Online and Huawei Share can connect to nearby devices that are logged in with the same HUAWEI ID through Bluetooth or Wi-Fi P2P to share Wi-Fi hotspots and files.

If a device has Instant Online enabled, the device will send signals through BLE to connect to devices that are logged in with the same HUAWEI ID. The EMUI device authentication service performs the authentication process using the trusted public keys of both devices to verify whether the peer device is using the same HUAWEI ID. If the authentication is successful, the EMUI device authentication service will provide the key used for creating this hotspot session, so that Instant Online can use the session key to encrypt private hotspot connection information.

When a user enables Huawei Share to share files with nearby devices that are logged in with the same HUAWEI ID, the EMUI device authentication service also verifies whether the devices are under the same HUAWEI ID and then encrypts and transfers files.

IoT Device Interconnection Security

The EMUI provides device management security for users, supports a point-to-point trust relationship between EMUI devices and IoT devices, provides trusted device authentication and end-to-end encryption for data transmission between two devices that have established the trust relationship, and ensures forward security of communication data.

Device Identifier

An EMUI device generates different identifiers for different IoT device management services to isolate different services. A service identifier is based on the Ed25519 public and private key pair which is generated on the security side of the EMUI device, and the plaintext information of private keys is stored in the TEE. An IoT device also generates its own device identifier for communicating with EMUI devices. This identifier is also generated based on the Ed25519 public and private key pair. The private key is stored in the IoT device. Each time the device is restored to factory settings, the public and private key pair will be reset.

The identifier can be used for communication between EMUI devices and between an EMUI device and an IoT device. To ensure secure communication between devices, the identifier is securely transmitted using end-to-end encryption from the TEE environment of one device to the TEE environment of another device.

Point-to-Point Security Binding Between Devices

During the establishment of a point-to-point trust relationship between devices, the user needs to enter the PIN provided by the peer device on the EMUI device. For a device with a screen,

the PIN is dynamically generated. For a device without a screen, the PIN is preset by the device manufacturer. A PIN can be a 6-digit number that is readable to the user or a QR code. Devices use the Password Authenticated Key Exchange (PAKE) protocol for authentication and session key negotiation, and on this basis, protect the integrity of the exchanged public key for authentication to ensure further communication security.

The public key information of the peer end is stored in the TEE side (SFS) on an EMUI device. This ensures that the trust relationship with the communication peer end cannot be tampered with.

Communication Security Between Trusted Devices

During the communication between EMUI devices or between EMUI and IoT devices that have established a trust relationship, the two parties authenticate each other by using the locally stored peer-end public key after completing the security binding process. All communication uses the Curve25519 public and private key pair of each session. The session key negotiation is performed using the Station-to-Station (STS) protocol, and two-way authentication is completed during the negotiation.

When a user uses Huawei Share OneHop to sync files or share the clipboard between a mobile phone and a Huawei PC, a point-to-point secure connection is established through the secure binding process, and transmitted data is encrypted with a session key.

When a user uses AI Life and connects a mobile phone to a Huawei-certified third-party sensitive accessory, a point-to-point secure connection is established through the secure binding process, and transmitted data is encrypted with a session key. In this way, the user can communicate with IoT devices without disclosing any data to Huawei.

8 Payment Security

This chapter describes security protection for Huawei Pay and other mobile payment apps. For third-party payment apps, the 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.

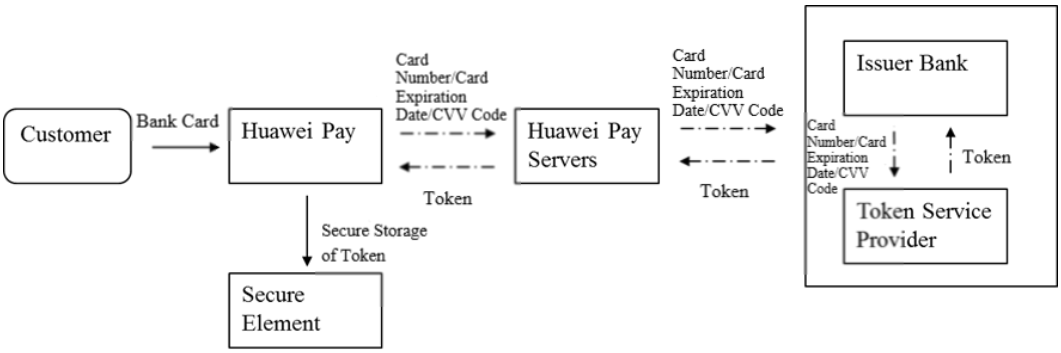
If any terms and conditions are returned by the card issuer for the payment card confirmation process, Huawei downloads the terms and conditions and displays them on the user's device.

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 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.

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, the 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.

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, the EMUI protects SMS verification codes against text message interception by malicious apps.

The 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 the 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 the EMUI's built-in SMS client is set as the default SMS client.)

9 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 the 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 the 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.

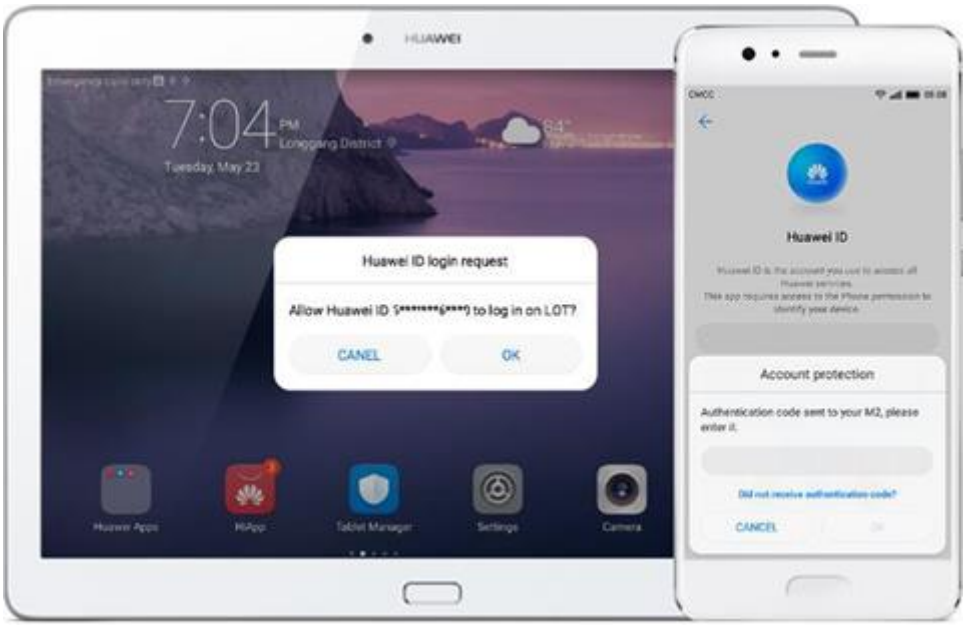
In cases where a user requests a major change to a HUAWEI ID, for example, when the user changes the password or uses the HUAWEI ID on a new device, Huawei will send a text message, email, or notification to the user. If any exception occurs, Huawei will prompt users to immediately change their passwords. 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

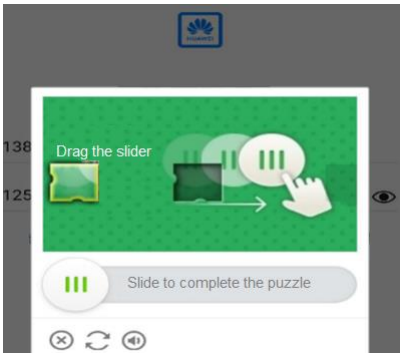
Two-factor authentication is the optimal account protection solution and ensures that the use of HUAWEI IDs is more secure.

Figure 9-1 Account protection



Account protection allows users to log in to their HUAWEI IDs using only their trusted devices. When attempting to log in to a HUAWEI ID from a new device, the user will need to enter the HUAWEI ID password and security verification code. The verification code is automatically displayed on the trusted device or sent to the user's trusted phone number. If the new device passes the verification, it will become the user's trusted device. For example, a user has a HUAWEI M2 and wants to log in to the user's HUAWEI ID on a new HUAWEI P10. The HUAWEI P10 will require the user to enter the HUAWEI ID password and the verification code displayed on the HUAWEI M2. This approach ensures that HUAWEI IDs and HUAWEI ID services (such as MyCloud, AppGallery, Wallet, and HiGame) are more secure.

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.

10 Device Management

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

Find My Phone and Activation Lock (for Mainland China)

The EMUI provides the function of Find My Phone. 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), and erase device data to ensure device data security. The user can implement these operations by logging in to the cloud service website (cloud.huawei.com) or using the Find My Phone function on a Huawei phone. (Note: The function is available only in mainland China.)

In addition, the EMUI provides the function of activation lock. Enabling Find My Phone will also enable the activation lock function on the mobile phone. If an unauthorized user attempts to forcibly erase data from a lost phone, after the phone is rebooted, the user needs to log in to the HUAWEI ID to re-activate the phone. This function ensures that unauthorized users cannot activate or use the phone, thereby protecting security of the phone.

MDM API

The EMUI opens up the device management interface SDK of Huawei phones and tablets 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/30701.

For device management APIs required by enterprise mobile office customers, the EMUI grants the customers corresponding use permissions by verifying a signed enterprise certificate. The enterprises 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 normally invoke the authorized APIs on Huawei devices.

When a user installs an APK that has a device management certificate, the EMUI analyzes and verifies each item of the certificate. After confirming that the signature is correct, the APK passes the authorization and is installed. If the certificate fails the verification, the APK will not have the device management permission. As a result, invoking the device management APIs fails and the developer is prompted with a security exception to ensure security of Huawei devices.

11 Privacy Protection

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

Permission Management

The EMUI operating 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, apps request permissions, and users determine whether to grant the permissions. The 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 call
- Network
- SMS
- Contacts
- Call record
- Camera
- Location data
- Recording
- Wi-Fi
- Bluetooth
- Calendar
- Body sensor
- Health and fitness
- Storage
- Sending multimedia messages
- 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 to steal users' privacy data, the 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.

Location Access

The EMUI allows a user to enable or disable location access in **Settings**. After location access is disabled, the 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. The 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 can’t be accessed to any other apps.
- App group: The ID is available to a group of apps, such as a group apps provided by the same vendor.
- Device: All apps installed on the device access a same ID.

Resettability and Durability

The resettability and durability define the lifecycle of identifiers. A user with a persistent and more reliable identifier is more vulnerable 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, the 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	It can be used by any app.	A random number is generated each time an identifier is invoked.	The UUID is regenerated each time an identifier is invoked.
HUAWEI ID	HUAWEI ID	Used for Huawei Cloud service features, AppGallery, MyCloud, and 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, and 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. Assign IDs based on third-party app signatures.	The Developer ID is randomly generated during app installation. Different IDs are generated for the same app when it is installed twice.	The ID is regenerated when the app is reinstalled.
	Open anonymous identifier (OAID)	Provided for advertisers in the advertisement placement scenario.	Regenerated after being reset.	Manually reset the anonymous device ID.

Differential Privacy

The EMUI uses differential privacy technology to protect information that users share with Huawei while improving user experience. Differential privacy adds random information to data. This prevents Huawei from associating a device with a user when users' data is analyzed. The data patterns appear only when user data is combined with the data from a large number of other users and the randomly added information averages out. These patterns help Huawei understand how users use their devices (so that Huawei can improve related services and products) without collecting personal information.

For example, when collecting information on user experience improvement, the EMUI does not directly upload the original information about the user's operations on the app. Instead, the EMUI generates a digest of the information. In addition, random information is added to the sets of digests. This ensures that it is impossible to associate the data with the user's device.

Privacy Statement

The 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 the EMUI released in the local countries.

Read Huawei Privacy Statement at
http://consumer.huawei.com/minisite/worldwide/privacy-policy/en/index.htm.

12 Conclusion

Huawei attaches great importance to users' device security and privacy, and designs the EMUI to provide an end-to-end (from bottom-layer chips and systems to apps) security protection capability. The EMUI constructs a trustworthy basic architecture for the device based on the chip hardware, and constructs security experience considering both security and user experience based on higher security and good computing performance of the device hardware.

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

While providing security solutions, Huawei also attaches great importance to establishing the security process and security capabilities to implement security management throughout the lifecycle of products.

Huawei has set up a dedicated 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 construct Huawei device security with all stakeholders.

13 Acronyms and Abbreviations

Table 13-1 Acronyms and abbreviations

Acronym/Abbreviation	Full Name
3DES	Triple Data Encryption Algorithm
AES	Advanced Encryption Standard
AI	artificial intelligence
API	Application Programming Interface
ARM	Advanced RISC Machines
ASLR	address space layout randomization
BLE	Bluetooth Low Energy
BYOD	bring your own device
CA	certificate authority
CDD	compatibility definition document
CE	credential encryption
CERT	computer emergency response team
CFI	Control Flow Integrity
CFNR	China Financial National Rising Authentication
CMP	Certificate Management Protocol
CNN	convolutional neural network
CT	call transfer
CVV	card verification value
DAC	discretionary access control
DE	device encryption
DEP	data execution prevention

Acronym/Abbreviation	Full Name
DRM	digital rights management
DSO	dynamic shared object
EAP	Extensible Authentication Protocol
ECDSA	Elliptic Curve Digital Signature Algorithm
eID	electronic identification
EIMA	EMUI Integrity Measurement Architecture
EMM	Enterprise Mobility Management
eMMC	embedded multimedia card
EMUI	Emotion UI
GP	GlobalPlatform
GPS	Global Positioning System
HKIP	Huawei Kernel Integrity Protection
HMAC	Hashed Message Authentication Code
HTTPS	Hypertext Transfer Protocol Secure
HUK	hardware unique key
HUKS	Huawei Universal Keystore
IKEv2	Internet Key Exchange version 2
inSE	Integrated Secure Element
IPsec	IP Security
JCA	J2EE Connector Architecture
JCE	Java Cryptography Extension
JOP	jump-oriented programming
KASLR	kernel address space layout randomization
L2TP	Layer Two Tunneling Protocol
LBS	location-based service
LTO	link-time optimization
MAC	mandatory access control
MDM	Mobile Device Management
MPPE	Microsoft Point-to-Point Encryption
MS-CHAP v2	Microsoft Challenge Handshake Authentication Protocol version 2

Acronym/Abbreviation	Full Name
NE	no encryption
NFC	Near Field Communication
NIST	National Institute of Standards and Technology
NPU	Neural-Network Processing Unit
OAID	open anonymous identifier
ODID	open device identifier
OTA	over the air
PAKE	Password Authenticated Key Exchange
PAN	Privileged Access Never
PIN	personal identification number
PKI	public key infrastructure
POSIX	Portable Operating System Interface
PPTP	Point-to-Point Tunneling Protocol
PSK	pre-shared key
PXN	Privileged Execute Never
REE	Rich Execution Environment
RNG	random number generator
ROM	read-only memory
ROP	return-oriented programming
RSA	Rivest Shamir Adleman
RPMB	replay protected memory block
SCEP	Simple Certificate Enrollment Protocol
SCP	Secure Channel Protocol
SD	Secure Digital Memory Card
SDK	software development kit
SELinux	Security-Enhanced Linux
SFS	secure file system
SHA	Secure Hash Algorithm
SMS	short message service
SoC	System-on-a-Chip
SOTER	Standard Of auThentication with fingERprint

Acronym/Abbreviation	Full Name
SSL	Secure Sockets Layer
STS	Station-to-Station
TA	trusted application
TEE	Trusted Execution Environment
TLS	Transport Layer Security
TSM	Trusted Service Manager
TUI	trusted user interface
UID	user identity
UUID	universally unique identifier
VPN	virtual private network
WAPI	WLAN Authentication and Privacy Infrastructure
WLAN	Wireless Local Area Network
WPA	Wi-Fi Protected Access

Change History

Date	Description
2019-08-30	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	Released the first version.

