MagicOS 8.0 Security Technical White Paper

Released: 2024-01-10

HONOR

Trademarks and Permissions

HONOR and other HONOR trademarks are trademarks of Honor Device Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between HONOR and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Honor Device Co., Ltd.

Address: Suite 3401, Unit A, Building 6, Shum Yip SkyPark, No. 8089, Hongli West Road, Xiangmihu Street, Futian District, Shenzhen.

URL: https://www.hihonor.com

Customer service: 4006966666

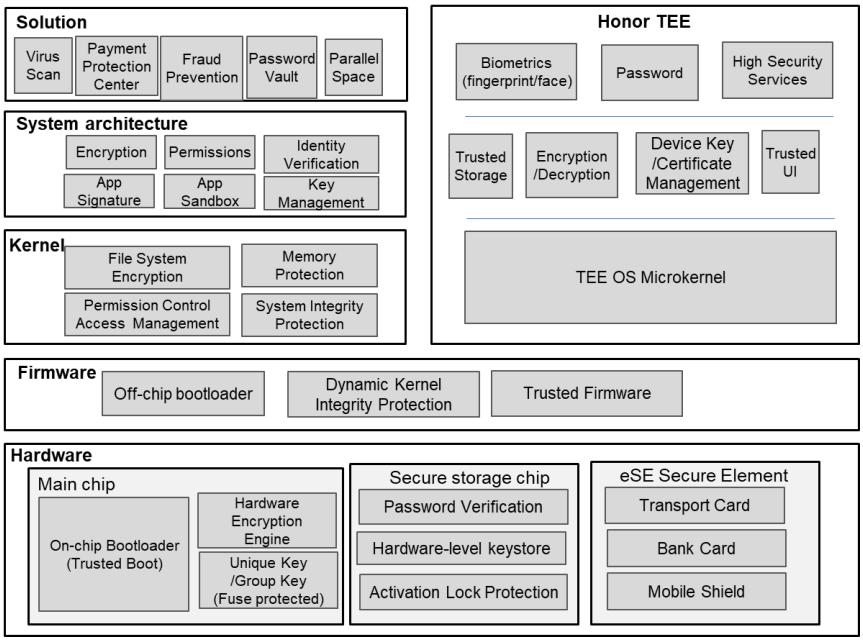
Contents

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.

Overview

Introduction

MagicOS is a secure, data-centric, and chip-based platform that combines hardware and software to provide a complete solution for user data security and privacy protection needs (Overall architecture shown below). It provides end-to-end security protection (hardware, system, apps, and the cloud). The security and privacy protection encompass chips, trusted execution environments, system kernels, data, apps, networking, connectivity and services.



MagicOS security architecture

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

To ensure data security, MagicOS encrypts user data using a hardware unique key (HUK) and a user lock screen password. Data files from various apps are stored in the sandboxes of the corresponding apps, preventing files

from one app from being accessed by another. The data erasure function is provided to permanently erase data during device recycling or factory restoration, thereby preventing unauthorized data restoration. MagicOS 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, MagicOS pre-installs System Manager to provide virus scanning, block and filter, traffic management, notification management, and other functions. Utilizing these functions, MagicOS can automatically detect viruses and Trojans within apps, and provide fine-grained permission, traffic, and notification management functions.

This document contains the following chapters:

Hardware security: secure boot, hardware encryption/decryption engine and random number generator (RNG), HUK, device group key, secure element.

TEE: secure OS, trusted storage services, encryption and decryption, device attestation, etc.

System security: integrity protection covering HONOR Kernel Integrity Protection (HKIP), Integrity Measurement Architecture (IMA), and system software update; kernel security covering system access control and kernel address space layout randomization (KASLR); identity authentication.

Data security: Universal Key Library, lock screen password protection, data encryption, secure erasure, and password vault.

App security: app signature verification, app sandbox, runtime protection, secure input, app threat detection, malicious website detection, and SMS verification protection.

Network and communication security: virtual private network (VPN), Transport Layer Security (TLS), Wi-Fi security, and protection against fake base stations.

Interconnection security: interconnection security between devices logged-in with the same HONOR ID; IoT device interconnection security.

Service security: HONOR ID, Find Device & Activation Lock, Payment protection center.

Privacy protection: permission management, privacy access history, audio/video recording reminder, location access, device identifier system, differential privacy, and privacy statement.

MagicOS 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.

Hardware security

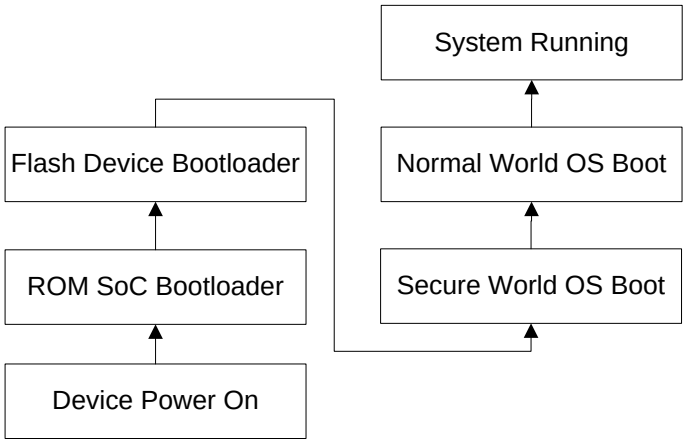
MagicOS adopts security capabilities based on hardware chips and delivers overall security with secure software solutions. Hardware chip security is the core of the MagicOS security system. This chapter describes HONOR device hardware chip security, including the following security features:

Secure Boot

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

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

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



A.1.1.1.1.a.i.1 Secure Boot

Products running MagicOS supports the Verified Boot feature. When accessing a read-only system partition with Verified Boot protection enabled, the system verifies the integrity of the accessed area using the integrity protection information generated when building the read-only partition image. This feature helps prevent malware from permanently residing on the system partition and ensures that the user boots the device in the same state as the last time it was used.

Hardware encryption and decryption engine and random number generator

To meet the requirements of high-performance encryption/decryption and key protection, MagicOS utilizes the hardware security engine to perform operations such as data encryption/decryption and key derivation. The chip provides a high-performance hardware encryption and decryption acceleration engine that supports the following main algorithms and functions (including but not limited to).

3DES, AES128, AES256

SHA1, SHA256

HMAC-SHA1, HMAC-SHA256

RSA1024, RSA2048, RSA3072, RSA4096

ECDSA-P256, ECDH-P256

ED25519, X25519

SM2 and SM4 algorithms

CTR_DRBG RNG compliant with NIST SP800-90A and hardware entropy source compliant with NIST SP800-90B

Hardware Unique key (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 MagicOS. 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 MagicOS to derive the same key for the same type of devices.

StrongBox*

StrongBox is a hardware-based key management function which offers better defense capability on side channel attacks and semi-invasive attacks. Applications can protect their keys in a more secure way through the interfaces provided by StrongBox. HONOR products use Secure Elements, and independent secure storage chip to support StrongBox, delivering more secure key management measures and stronger protection measures on digital identity authentication services, such as passkey.

Secure Element*

The Secure Element is a chip that provides secure execution, data storage protection and industry security certification to meet the security requirements of mobile payments. Secure Elements have independent memory, persistent storage media, encryption/decryption logic circuits, processors, and software systems to protection applications and data against external attacks. HONOR products also use secure elements to ensure the security of payment-related functions. The secure elements have passed CC EAL6+ (hardware) and EAL5+ (software) security certifications and international standards such as EMVco.

Mobile Shield*

The Mobile Shield function supported by HONOR products uses Secure Element to support the mobile certificate service of banks, combining the traditional USB plug-in U shield with the mobile phone and turning it into the mobile shield to provide financial-grade hardware protection for electronic payments.

When the user opens Mobile Shield, MagicOS's Trusted Service Management Platform (TSM) will act as the manager of the Secure Element, and the functional modules on the phone will communicate with the Secure Element

by establishing a Secure Channel Protocol (SCP) and creating a trusted, independent, and secure operation space within the Secure Element. The banking application will then generate key pairs and certificates for transactions in this secure space and requires the user to set up PIN protection.

When using Mobile Shield, the user first enters the PIN code for authentication through the trusted UI interface. Then the Secure Element will digitally sign the user's transaction request using the private key generated during the creation process. The bank transaction system performs signature verification when processing the transaction request to ensure the security of the transaction.

When the user logs out of (closes) Mobile Shield, the system destroys the key pairs stored in the Secure Element and makes sure they cannot be recovered.

From the generation of the certificate and the public and private keys to the destruction of the certificate, the private key will be stored in the Secure Element during the whole life cycle to guarantee the security of the certificate keys.

Independent secure storage chip*

Some of HONOR's products use a secure storage chip that is independent of the main chip. The chip meets the hardware-grade CC EAL5+ security standard and has independent memory, permanent storage media, processor, hardware encryption and decryption engine, and software system to further enhance the protection of lock screen password verification, biometric data, and activation lock data to protect users' data security.

The HTEE uses Secure Channel Protocol (SCP) and shared key pairs for secure communication between the controller program and the secure storage chip. The key pair is pre-set during the production of the device using the Hardware Unique Key (HUK) of the main chip and injected into the secure storage chip through the HTEE to achieve a one-to-one binding between the secure storage chip and the main chip, avoiding the security risks associated with chip replacement and removal.

The communication keys used by the secure channel protocol are derived at the establishment of the communication by randomly generated factors and the key pair to prevent compromise.

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

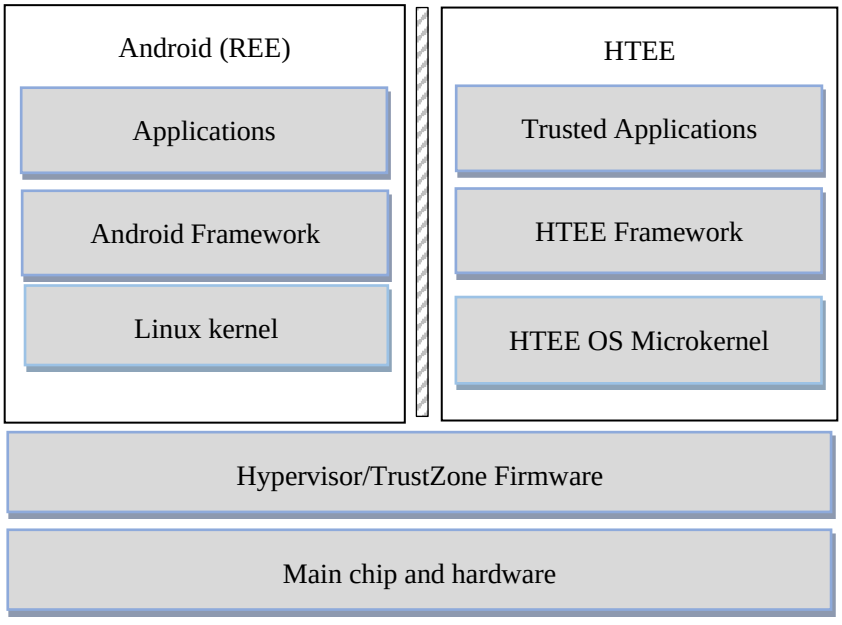
Trusted Execution Environment

HONOR provides a trusted execution environment (TEE) in compliance with Global Platform (GP) TEE specifications. It is a secure OS independently developed by HONOR based on the formal microkernel, and features high security, performance, scalability, and stability.

HONOR Trusted Execution Environment (HTEE)

HTEE is a trusted execution environment implemented by HONOR based on the microkernel technology. It includes a secure OS kernel, framework layer and trusted system core applications, and is built on TrustZone and virtualization 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.

Hypervisor virtualization is a widely used technology. Virtualization allows for the isolation of virtual machines running on the same physical core. This allows multiple execution environments to share the same hardware environment. For chips with ARM architecture, Hypervisor runs at the EL2 exception level. Only software running at the EL2 exception level or higher can access and configure various virtualization features. HTEE runs in a VM, which is isolated from other VMs and communicates with other VMs through Hypervisor.



*Note: Some product models use a TEE provided by the main chip manufacturer that may differ from the HONOR HTEE in function and specification.

Microkernel

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

Good scalability: A unified security kernel is built for distributed devices, allowing heterogeneous devices to support various services, such as multi-core, on-demand concurrency, and large- and small-core scheduling.

Easy to implement and debug: Stable underlying library interfaces facilitate application development and porting, as well as supporting the development of the security service ecosystem.

Formal Verification

HTEE uses formal verification to improve the TEE kernel's system security level significantly, thus building trust and security. Formal verification uses mathematical theorems to verify system correctness (without vulnerabilities) from the source. Conventional verification methods (such as function verification and attack simulation) apply only to limited scenarios, while formal verification can use data models to verify all software running paths. This process verifies the correctness of core modules, core APIs, and high-level mechanisms, such as process isolation and permission management, preventing data race and memory access errors.

HTEE is constantly working toward a TEE without vulnerabilities to provide higher security assurance for products.

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

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

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

Content protection	Applies to the digital rights management (DRM) field to ensure security and anti-copy during playback.
Mobile payment	Ensures the security of input information and can be used together with Near Field Communication (NFC). HTEE protects user input information against theft from malicious programs.
Application logic protection	Protects critical application logic from being stolen or tampered with.

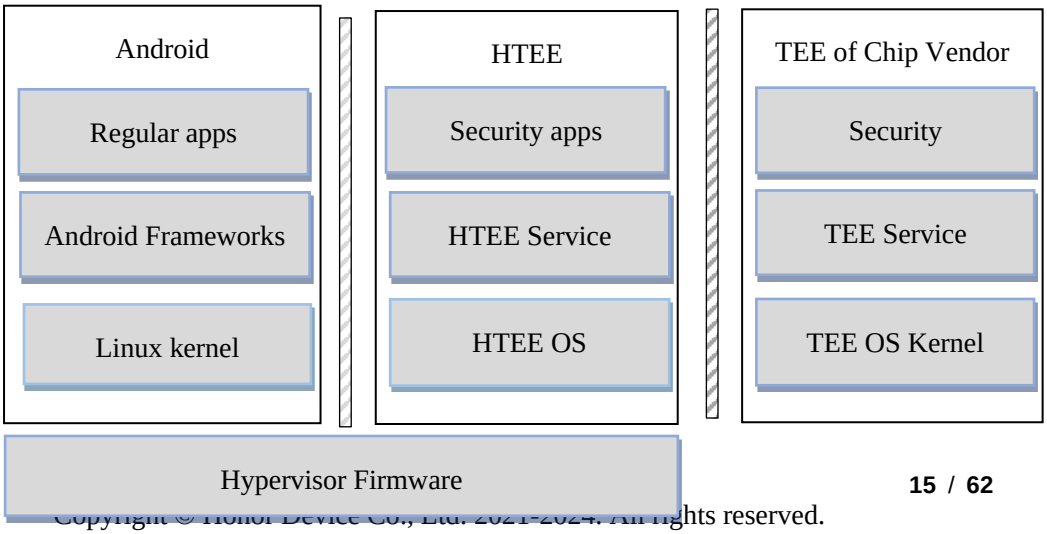
As an example, MagicOS provides a TEE to protect the security of services on HONOR mobile phones, such as fingerprint/face unlock and payment, and secure key management.

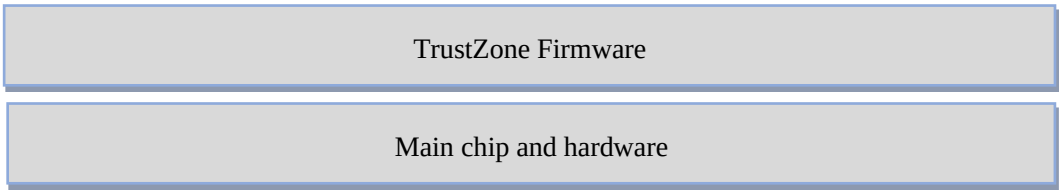
Dual TEE system

Creating a dual TEE system in some products by combining Main chip manufacturer TEE and HONOR HTEE (Based on Trustzone and Hypervisor virtualization). Thus, building trustworthy system security capabilities and a friendly security ecosystem.

SoC chip vendor TEE continues to build a hardware security framework based on the chip's own core capabilities of secure boot, secure hardware (RPMB, hardware encryption engine), and key management, providing hardware-level security capabilities such as trusted storage, encryption and decryption, and file encryption.

With the ability to develop, deploy manage, and maintain security applications throughout their entire life cycles and basic capabilities such as formal verification, multi-core multi-threading, and encryption, HTEE can achieve rapid integration of existing security services and continuous expansion.





*Note: the dual TEE feature is only available for certain chip models.

HTEE supports the following basic security capabilities:

Trusted Storage Service

Trusted storage of HTEE secure OS 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 NAND Flash. The RPMB supports anti-deletion and anti-rollback. Trusted Storage supports device binding and isolation between different security applications. Each security application can only access its own stored content and cannot open, delete or tamper with the data of other applications.

The Secure File System (SFS) based on HTEE Secure OS provides critical data storage protection capabilities. It can be used to store information such as keys, certificates, personal privacy data, and fingerprint templates to ensure confidentiality, integrity, atomicity, isolation, etc.

The TA (Trusted Application) running in HTEE can encrypt data and store it in the secure file system through the secure storage API. Encrypted data can only be accessed by the TA itself and not by external applications.

The secure storage uses AES256 encryption/decryption and is compatible with the GP TEE standard specification. The key of the secure storage is derived through the unique key of the device and does not go out of the TEE secure zone. Data encrypted by the key cannot be decrypted outside the secure zone.

MagicOS further provides the Flash-based RPMB (Replay Protected Memory Block) storage feature to protect critical system data from unauthorized deletion and access. RPMB is directly managed by HTEE for security, the access authentication key of RPMB is bound with the unique key of the device. As an API is not provided at REE side, only HTEE can access the protect contents of the RPMB partition. RPMB data is protected by built-in counters/keys and HMAC verification to prevent replay attacks and ensure that the data is not maliciously overwritten or tampered with.

Encryption/Decryption Service

HTEE supports multiple symmetric and asymmetric encryption and decryption algorithms, as well as key derivation algorithms. It supports the

same key derived on a chip platform, HUK, keys derived of hardware based on secure elements, and international standard cryptographic algorithms. It also provides support for third-party development of service TAs that store and use keys, and complies with GP TEE specifications.

To improve security, key generation and calculation in HTEE 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.

Post-Quantum Cryptography

In response to the data security threats brought by the rapid development of quantum computer technology, the National Institute of Standards and Technology (NIST) released the first batch of post-quantum cryptography algorithm FIPS drafts in 2023: CRYSTALS-KYBER (FIPS 203), CRYSTALS-Dilithium (FIPS 204) and SPHINCS+ (FIPS 205).

In order to ensure the security of user data and prevent the existing data against the potential “Store Now, Decrypt Later” (SNDL) attacks that quantum computing heralds, MagicOS 8.0 has started post-quantum cryptography migration and introduced post-quantum public key encapsulation mechanism algorithm (CRYSTALS-Kyber) and post-quantum digital signature algorithm (CRYSTALS-Dilithium) to encrypt and protect digital signatures for some key data of users.

*Note: This feature is only available on some product models released in the China market.

Device Attestation

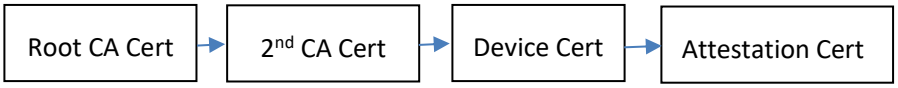
To ensure that MagicOS devices are trusted, device certificates and public-private key pairs are already provisioned during the production stage. Device certificates and public-private key pairs are different for each device and are used to identify the device. The certificates and keys are written to TEE and then encrypted and stored. This information can only be accessed through the HONOR key management service.

Device certificates are issued by the HONOR PKI system and contains a three-level certificate chain as follows.

if the device, user or account needs to be verified for services with high security requirements such as payment and account management, the

corresponding certificates (also called Attestation certificates) can be issued by the device certificate and private key to form a certificate chain. Operations can only be executed after verification, ensuring that only trusted devices can carry out the corresponding operations.

The service certificate is issued in the TEE through the device certificate, which contains a four-stage certificate chain as follows. An operation can only proceed if device legitimacy is proven by simultaneously passing the four-stage certification chain and the signature verification.



Trusted Display and Input (TUI)

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

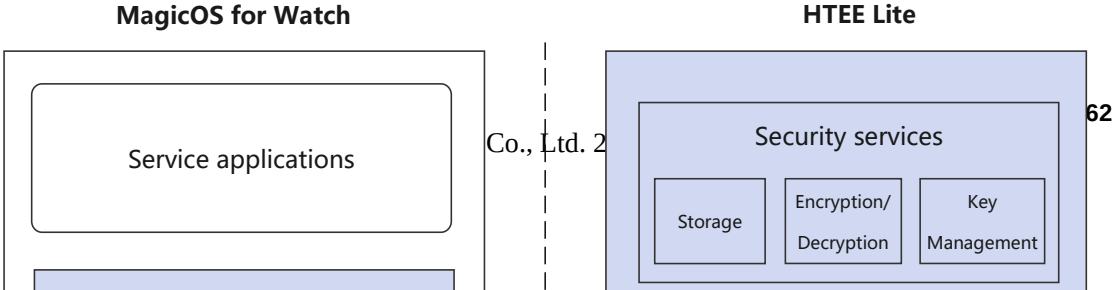
The TUI ensures that the information displayed to users is not intercepted, modified, or obstructed by any software in the REE or unauthorized apps in the TEE. Displayed information is not transferred to the REE, and permission control is used to ensure that only authorized TEE apps can access the information.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

The TUI supports basic controls such as PNG images, texts, buttons, and text input boxes, display of Chinese characters, English letters, symbols, and digits in the same size, customized UI, randomized keypad, and various controls and window management. In addition, the UI is consistent in style with MagicOS.

*Note: The TUI feature is available only for certain chip models.

HTEE Lite*

HTEE Lite is the trusted execution environment HONOR provides for watch products. Leveraging the TrustZone technology for Arm Cortex-M to provide an independent secure world with hardware isolation, HTEE Lite applies to operations requiring high security, such as payment and key information storage.



System Security

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

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

Integrity Protection

HONOR Kernel Integrity Protection (HKIP)

Although secure boot and verified boot ensure the authenticity and integrity of software during startup, vulnerabilities in authentic code may still be exploited by attackers. HKIP uses the hypervisor mode provided by the ARMv8 processor to protect the kernel, preventing key system registers,

page tables, and code from being tampered with. This protects system integrity and prevents privilege escalation during system runtime. HKIP protects not only static data such as code and read-only data segments, but also some dynamic data using the write-rare protection mechanism. HKIP uses this mechanism to secure kernel data that is read most of the time but rarely modified. Even if attackers exploit vulnerabilities to write the memory at the kernel level, they cannot modify the protected data.

Currently, HKIP supports the following security protection mechanisms:

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

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

Integrity Measurement Architecture

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

1. Baseline extraction

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

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

2. Static measurement

The integrity of a file means that its content or attributes have not been modified. From a cryptography point of view, the hash value of a file can be used to detect whether the file has been tampered with. Therefore, the hash values of measured objects are collected to determine the integrity of programs or data instances during memory loading.

3. Runtime measurement

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

*Notes: This feature is only available on products using the MTK chip platform.

System Software Update

MagicOS supports over the air (OTA) update to quickly fix some defects or deliver some new features and services. The security protection process in system software updates is as follows.

The signature of an update package is verified during system software updates. Only verified update packages are considered authentic and can be installed.

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

MagicOS periodically releases security patches. After the system is upgraded, required security patches are automatically updated to ensure the security of the MagicOS system. For more information about software security updates, visit https://www.hihonor.com/cn/support/bulletin/

Kernel Vulnerability Anti-exploitation

Kernel Address Space Layout Randomization (KASLR)

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

Privileged Access/eXecute Never (PAN/PXN)

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

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

Using some kernel attack methods, an attacker can tamper with the code pointer in some data structures used by kernel so that the pointer can be redirected to the privilege escalation code in user mode and executed by using system call. PXN prevents the kernel from directly executing user-mode code, thereby preventing such attacks.

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.

MagicOS uses Clang CFI, stack protection technologies, and PA/BTI 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.
- MagicOS checks the stack layout when the function runs to the end and exits to prevent attackers from exploiting the overflow vulnerability to modify the return address.

- Pointer Authentication (PA)/Branch Target Identification (BTI) are hardware-based ROP/JOP attack mitigation measures. PA provides signature and signature verification for pointers to ensure the integrity of pointers. BTI restricts the target of function jumps to ensure the integrity of jump targets.

  *Note: PA/BTI is chip-dependent and is only available on some products.

Mandatory Access Control (MAC)

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

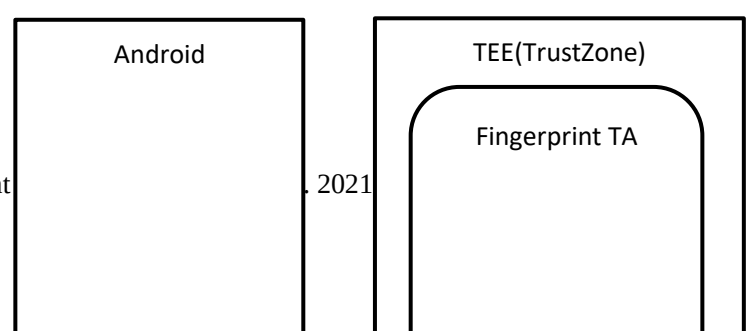
MagicOS also supports the secure computing (seccomp) feature that restricts the system calls that can be invoked by upper-layer application processes based on the rule files in the read-only file systems, preventing malicious apps from using sensitive system calls to compromise the system.

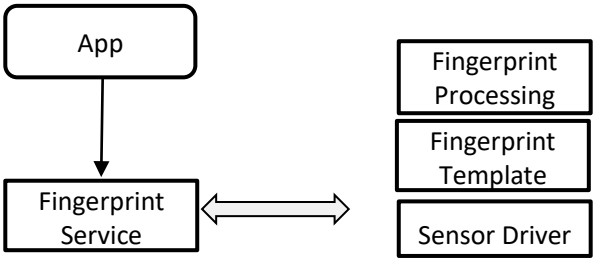
Identity Authentication

Fingerprint Recognition

MagicOS 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 MagicOS's fingerprint recognition security framework.





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

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

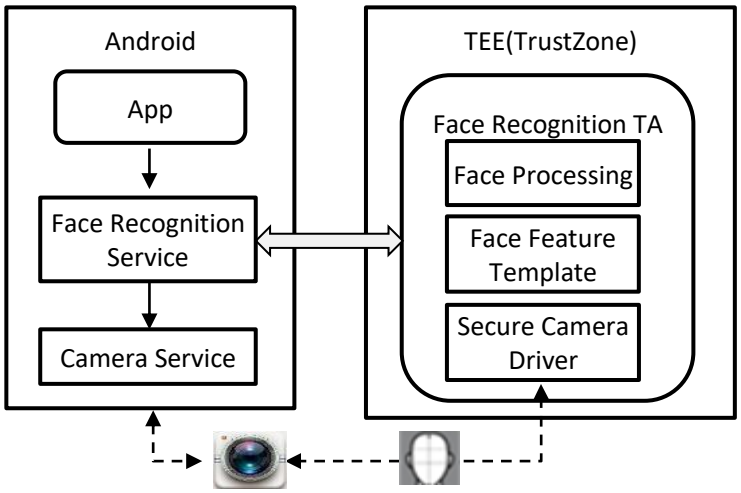
Dirty or damaged fingerprint sensors, dirty or wet fingers, and other external factors may affect the recognition rate, and should be avoided.

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

Facial Recognition*

MagicOS 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.

The following figure shows MagicOS's 3D facial recognition security framework.



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

MagicOS'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.

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

*Note: 3D face recognition is only available on certain product models.

In products that support independent secure storage chip, the face and fingerprint template data recorded by the user in the device are encrypted by a double encryption mechanism, based on the HUK of the main chip and the key stored in the secure storage chip in the HTEE.

Continuous Identity Authentication*

MagicOS provides continuous identity authentication which leverages the AO low-consumption camera and low-consumption NPU capabilities to continuously identify the device owner based on his/her biological

information (such as facial information). The "Smart display" function that uses continuous identity authentication capabilities is turned off by default. Users can turn on this function as needed and must agree to the Privacy Statement before using it. When "Smart display" is turned on and a new banner notification is received when a user's phone is unlocked, the system uses the continuous identity authentication capabilities to check whether the current user is the device owner. The system will show the banner notification content if the current user is the device owner and will hide the banner notification content if the current user is not the device owner, no one or too many people are facing the front camera, or the front camera is occupied.

The biological information used for continuous identity recognition is collected and processed only on the end-side security environment. The biological information is stored in a secure manner, with high-intensity cryptography algorithms used to perform encryption/decryption and integrity protection on the biological information. External devices cannot obtain the key of the encrypted biological information, so as to prevent the biological information against leakage. In addition, external third-party apps cannot obtain or transfer the biological information. MagicOS will not send or back up the encrypted or unencrypted biological information to any external storage medium, including the cloud.

*Note: Not all products support the continuous identity authentication function.

Data Security

This chapter describes MagicOS data security protection. The MagicOS 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. MagicOS 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 critical asset, secure erasure, and password vault. In addition, MagicOS provides app developers with HUKS framework capabilities, which facilitates application developers to store application keys and digital certificates securely, and securely use key encryption to protect confidential data in applications.

HONOR Universal Keystore

HONOR Universal Keystore (HUKS) is a key and certificate management system based on Java Cryptography Architecture/Extension (JCA/JCE) in MagicOS, and provides keystore and crypto APIs for apps, including key management, symmetric/asymmetric encryption and decryption, certificate management, and other functions. It provides device authenticity verification based on device certificates. The cloud server can authenticate MagicOS devices through certificate authentication. In combination with biometric authentication, the HUKS can provide services such as login and payment with TEE security for payment apps.

HUKS managed keys and certificates are stored in the TEE, and all keys are protected by AES_256_GCM encryption based on hardware unique keys. When the key is used, the plaintext of the key is decrypted in the TEE before the data encryption and decryption operation, the plaintext of the key does not leave the TEE and the encryption and decryption process is protected by the TEE.

HUKS enforces strict access control over the use of keys. During key generation, HUKS records the UID (User ID, assigned by the system when the application is installed), signature, package name and other identity information of the application. When the application uses the key, HUKS first verifies the application's identity information and allows the application to use it only after the verification is passed.

HUKS supports enhanced key access control using biometric features (fingerprint/face recognition, etc.), and HUKS confirms biometric results before allowing applications to access and operate on the corresponding keys.

The HUKS also provides a key attestation function. A unique device certificate is written to each unit during manufacture. In the TEE, HUKS uses the device key and certificate to issue a key authentication certificate for the application key (see the device attestation chapter for details)

TPM-based key management

On PC devices, the implementation of HUKS is based on the Windows system and TPM-trusted firmware. These functions include key management, data protection, symmetric/asymmetric encryption and decryption, certificate management, and device authentication. With HUKS, MagicOS application developers can invoke the entire key and certificate lifecycle management capability, as well as encryption and decryption algorithms.

HUKS provides hierarchical key management based on hardware root key, and which has strict access controls of using (different type of) keys. The keys owned by different Windows users and different services are isolated from each other, and are protected by hardware unique key (HUK) using AES_256_GCM algorithm. The key management process is protected by the TPM, in which HUKS executes encryption and decryption; also, is invisible to the Windows system to against security attacks.

HUKS provides two forms of business data encryption functions: single key for single device, or single key for single model (GID-based). The encryption process binds invoker firstly, and the data protected by encryption can only be decrypted and used by the invoker later.

The symmetric/asymmetric encryption and decryption function of HUKS provides common cryptographic algorithms based on hardware secure random number, including symmetric AES encryption and decryption, HMAC verification, asymmetric RSA/ECC signature verification, key negotiation, etc.

Lock Screen Password Protection

MagicOS 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.

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

For products with a separate secure storage chip, the protection of the lock screen password verification process is enhanced by carrying out the lock screen password verification and anti-violence cracking mechanism (continuous error count and attempt interval timing) in the secure storage chip. Only after the password verification is passed, the controller program in the HTEE can obtain the material from the secure storage chip for key-derivation and decryption of the encrypted file, thus ensuring the security of user data.

Data Encryption

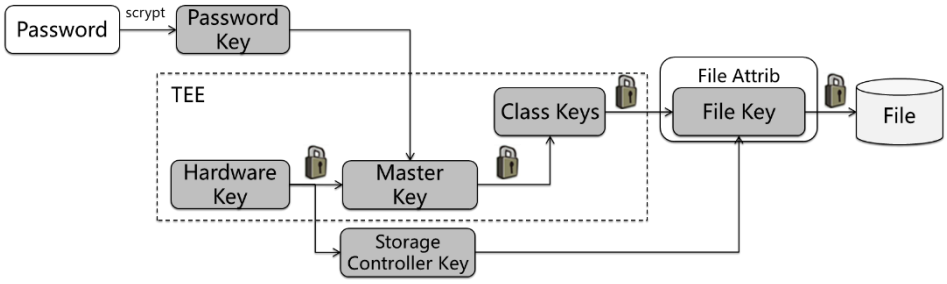
File encryption

MagicOS provide file-based encryption file system, based on the Linux Fscrypt framework and hardware encryption engine, and use XTS-AES 256 algorithm to encrypt data in storage.

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

1. Credential encryption (CE): Apps use this type of data encryption solution by default. In this type of solution, file encryption class keys are linked to the lock screen password and are protected by using both the lock screen password and HUK. CE-protected data is accessible only after an MagicOS device is unlocked for the first time. Since data encryption is linked with the user's lock screen password, the device's data will not be decryptable if the password is forgotten. Therefore, it is recommended that users should protect their passwords and backup their data.

2. Device encryption (DE): DE-protected data, such as wallpapers, alarm clocks, and ringtones, can be accessible after the device is powered on, independent of whether the device is locked or not. DE-based class keys are protected using the HUK and irrelevant to the lock screen password.



File encryption key protection

Secure Storage of Critical Asset

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 data can be stored in the secure storage. The critical asset secure storage service provides security for this data and fine-grained access control to the data.

Encrypted critical asset (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 critical asset can be stored in the secure storage:

Sensitive data: Sensitive data of key assets, e.g., users can save their account numbers and passwords to log in to applications quickly.

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, system assigned UID, and other information of the app that queries the stored data, to verify the access permission and ensure access security.

Secure Erasure

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

In factory restoration, MagicOS 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. Meeting NIST SP800-88 requirements.

Password Vault

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

The password vault 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 TEE, and encryption/decryption is always performed in TEE.

Currently, the account and password data stored in the password vault can be encrypted and transferred between HONOR devices that support the password vault through Device Clone (password vault clone is available only to the devices with HONOR PKI certificate). Alternatively, users can backup and restore password data with a PC backup software.

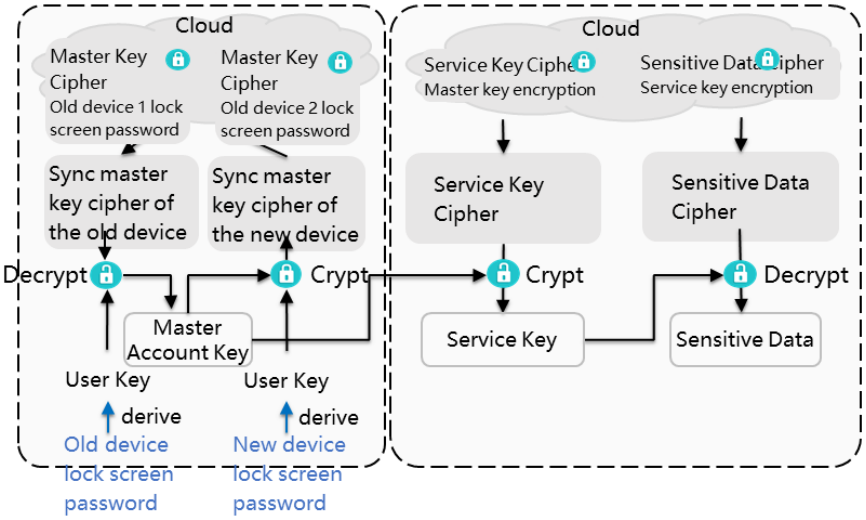
The password vault data transmitted in the Device Clone process is encrypted using AES_256_CBC. The encryption key is obtained through key exchange using the asymmetric key generated by two phones in TEE. Key exchange is performed in TEE, which also protects the obtained clone encryption key. Encryption 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 hardware unique key (HUK). A device's backup data on a PC cannot be restored on other devices.

Starting with MagicOS 6.0, you can automatically sync your account and password to other devices logged in to your HONOR ID via HONOR Cloud. Your information will be encrypted and unreadable by others as well as

HONOR. HONOR protects your information with end-to-end encryption to provide the highest level of data security. Your data is protected by a key generated from information unique to your device and a device password that only you know. Neither other persons nor HONOR can access or read this data, and it is encrypted in transit or storage.

Three-layer key system to protect account passwords on the terminal side



App Security

This chapter focuses on the security mechanisms for apps on MagicOS. 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.

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

During app installation, the signature verification mechanism prevents apps from being maliciously tampered with. The system will perform threat detection when an app is installed (e.g., virus, malware scan) and alerts the user if a risk is identified.

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, to prevent user data breaches.

Application Signature Verification

Only apps with complete signatures can be installed in MagicOS. 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.

MagicOS supports the following three Android V1/V2/V3 application signature verification methods:

Android V1 signature format - a JAR package-based signature scheme. Since V1 signature scheme does not protect the ZIP metadata of APK, using V1 signature alone on the latest Android systems is not recommended.

Android V2 signature format - a full-file signing scheme introduced after Android 7.0, is able to discover all changes made to protected parts of the APK (including ZIP central directory, end of central directory record and file data), thus helping to speed up verification and enhance integrity assurance. When using application signature scheme V2, an APK signature chunk will be inserted in the APK file. V2 signature and signer identity information will be stored in the application signature scheme V2 chunk. This can prevent attackers from forging V2 signature into V1 signature for verification.

Android V3 signature format - a signature scheme introduced after Android version 9 that supports application key rotation. It enables applications to change their signature keys during application updates and upgrades.

New signature formats are backward compatible in V1/V2/V3. MagicOS verify app signatures according to API level information and markers in the signature chunk. Apps intended for Android 11 (API level 30) must be signed using signature scheme v2 or higher.

App Sandbox

MagicOS provides a sandbox mechanism. This mechanism enables all apps to run in isolation within the sandbox 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 the app sandbox based on UID. 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.

Mandatory partitioned storage

To allow users to better manage their files and reduce clutter, apps intended for MagicOS 4.0 (Android API level 29) and higher are given partitioned access to external storage (i.e., partitioned storage) by default. Each app can only access specific directories on external storage allocated by the system, as well as specific types of media files created by this app. To ensure data security, app-specific directories will not be accessible to other apps.

Runtime Protection

Malicious apps usually obtain memory addresses by viewing the memory if the allocated memory addresses are relatively fixed during app operation. MagicOS provides Address space layout randomization (ASLR) and secure computing mode (Seccomp) 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.

Seccomp can control the scope of system calls that can be executed by app processes and prevent processes not within scope from executing. This can effectively prevent the attacks that are executed against the process due to the vulnerability of some system calls.

App lock

App Lock protects the app entrance and prevents private information in the app from being disclosed MagicOS users can enable app lock by going to Settings > Security > App Lock, then set the app lock password and select the apps that need to be locked. After App lock is enabled, users will have to be authenticated (using password, fingerprint, face, etc.) before launching a locked app. When an app is locked, its thumbnail in the recent tasks list is also protected to prevent snooping.

Secure Input*

MagicOS 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 apps for entering passwords and secure input does not take effect in such cases.

Virus scan

Security threats may exist in apps as a result of unknown third parties and downloading apps from unverified sources can introduce malicious threats. MagicOS 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.

MagicOS 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.

Block spam advertising*

Spam ads frequently pop up on the home screen, lock screen, and other interfaces, seriously affecting the user experience. MagicOS provides spam ad identification feature by evaluating the usage characteristics of3rd apps, to identify spam ad applications, block its background pop-up window, and prohibit the app from popping up the ad interface on the home screen and lock screen.

Fraud prevention*

Telcom fraud causes customer a huge amount of property losses. On one hand, MagicOS identifies and blocks risky calls, applications, and URLs. On the other hand, MagicOS identifies risk behaviors on the local device, as well as identifies and notifies of fraud risk behaviors so as to mitigate fraud risks for users.

Malicious URL detection*

MagicOS can detect phishing or malicious URLs in scenarios such as SMS and QR code scanning. It can automatically identify such URL when they are received in SMS messages or scanned by the camera.

SMS Verification code*

Verification messages have become one of the important authentication factors for mobile applications nowadays. However, hijacked verification messages can put user privacy or even property at risk. MagicOS is built with verification SMS protection to prevent malicious apps from intercepting user SMS and stealing verification codes to mitigate such risks.

MagicOS features an intelligent system layer SMS verification code recognition engine. Recognized verification code messages will only be distributed to the default SMS client set in MagicOS. If the default SMS client is the system SMS client, the system SMS client will encrypt and save verification code messages and screen access to prevent third-party SMS clients or apps from reading these verification code messages. Even if the SMS database is read directly, the content of verification messages is still encrypted and cannot be decrypted by other apps.

*Note: This feature is only available on some models in China.

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 MagicOS's security mechanisms for network connection and transmission, and security protection that MagicOS 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. MagicOS 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

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

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

Wi-Fi Security*

MagicOS 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 MagicOS 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.

Devices use 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. MagicOS provides a Wi-Fi threat detection engine for access points. It detects Wi-Fi hotspots before connection. If any security risks are detected, it will prompt users so that they can take measures to ensure the connection is secure.

*Note: This function is only available in China.

Protection Against Fake Base Station*

Unauthorized users can obtain user location and identity information by deploying fake base stations, 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. MagicOS provides chip-level protection against fake stations. It compares and analyzes network parameter characteristics for access and reselection of fake GSM/LTE stations and network parameter characteristics of normal stations, and rejects the residence and access of identified fake stations. (Fake LTE stations can only be identified by some chip platforms.) In addition to decoding system messages, the device can identify fake stations through combined process characteristics such as fake station attack without authentication redirection. This prevents a device from camping on or accessing cells with such characteristics.

*Note: This function is only available in China.

Device Interconnection Security

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

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

Interconnection Security for MagicOS Devices Under the Same HONOR ID

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

Networking Service for MagicOS Devices Under the Same HONOR ID

The device authentication service supports trusted networking of MagicOS devices that are logged in with the same HONOR ID, including mobile phones, tablets, and PCs. When the trusted networking service is enabled on an MagicOS device, the device authentication service performs identity authentication on each nearby device that is logged in with the same HONOR ID, and negotiates the session key between devices.

When a user enables family album for continuing playing a video on another MagicOS device under the same HONOR ID, device authentication and session keys negotiation are implemented for the device based on the trusted device networking service, and the session key is used to encrypt data transmitted between the devices.

Interconnection Security for AI Space and Magic-link

AI Space provides services for local data storage of home IoT devices. Magic-link provides encryption and authentication services for IoT device data uploaded to the cloud. AI Space generates a unique device key based on the trusted environment, uses the AES-GCM-256 algorithm to encrypt and cache user device information, and allows users to upload account identifiers of shared family members to Magic-link. The communication between the AI Space and the Magic-link, based on the TLS 1.2 and above security protocol, uses the session key obtained through authentication negotiation to encrypt

the transmitted payload. Magic-link assigns a unique key to each IoT device, uses the AES-GCM-256 algorithm to encrypt and saves uploaded device identity information and account identifiers of family members.

IoT Device Interconnection Security

MagicOS supports P2P trust relationships between devices that do not have an HONOR ID login UI on themselves (such as wearable devices) and MagicOS devices (such as mobile phones and tablets), and allow devices that have established a trust relationship to establish secure connections for E2E encryption and transmission of user data.

IoT Service Identifiers of MagicOS Devices

A MagicOS device generates different identifiers for different IoT device management services to isolate these services. The identifier can be used for authentication and communication between an MagicOS device and an IoT device. Similar to the account-level device identifier generated when an HONOR ID is used to log in to the device, the IoT service identifier is also an Ed25519 public-private key pair. The key pair is generated using elliptic curve cryptography in TEE of the MagicOS device, and the plaintext private keys are not transmitted out of TEE.

IoT Device Identifiers

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

The identifier can be used for secure communication between an MagicOS device and an IoT device. After both devices authenticate the service identifier or device identifier, they can perform key negotiation and establish a secure communication channel.

P2P Trusted Binding Between Devices

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

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

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

Communication Security Between MagicOS Devices and IoT Devices

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

When OneHop or Multi-Screen Collaboration is used to share data between a mobile phone and an HONOR PC, large-screen device, or tablet, a P2P trust relationship can be established through the secure binding process. For encryption of the transmitted data, the session key obtained during authentication is used.

Remote password authentication

After a trusted relationship is established among MagicOS devices logged in to the same HONOR ID and before the cross-screen-application transmits data across devices, the service requester needs to verify whether the current user is also owner of the service-receiving device through the remote password authentication mechanism. The data transmission will continue only after the authentication is passed.

For example, before the applications on the tablet request to mirror the screen of the phone, it needs to prove that the user of the current tablet is the owner of the phone. The remote password authentication mechanism provides a secure method to verify the identity of the phone owner on the tablet. When the user enters the lock screen password of the phone on the tablet, with the password remaining in the APP, the correctness of the password will be verified by interacting with the phone through the remote authentication protocol to complete the identification of the owner.

Security for remote authentication

On the server of remote authentication, i.e., the service receiving device (hereinafter referred to as "Server"), the user will use PBKDF2 to derive a key based on the lock screen password with salt while entering the lock screen password or updating the lock screen password and store it in the Trusted Execution Environment (TEE). When the APP, i.e., the service requesting device (hereinafter referred to as the "APP"), initiates remote authentication, the Server generates the server public-private key pair based on the above key and sends the public key, signature public key, challenge value, and salt of the Server to the APP. After the user enters the lock screen password of the service receiving device in the APP, another key will be derived using the PBKDF2 algorithm with a salt value, and APP public-private key pair will be generated based on the above key. A session key will be negotiated based on the APP private key and the Server public key and will be used as the HMAC key for forming message authentication code based on the challenge value and sending the APP public key and message authentication code to the Server. The Server negotiates the session key based on the APP public key and the Server private key and uses this session key to verify the message authentication code. If the verification passes, it means that the lock screen password entered in the APP is the correct one for the Server. The Server also needs to conduct two-way authentication to the APP at the same time, use the negotiated session key as the HMAC key for constructing a message authentication code that is derived from the challenge value, sign the authentication result information by ESDSA using the signing key, use the session key to encrypt the authentication result information and signature in CCM mode with AES, and then send the message authentication code and the

cipher text of the authentication result to the APP. The APP uses the negotiated session key to verify the message authentication code, uses it to decrypt the authentication result, and uses the signed public key to verify the authentication result. If the above operation is completed correctly, the authentication result is trusted, and thus the two-way authentication is completed. The whole processing involved in the above is done in the Trusted Execution Environment (TEE). The Server has the capability to prevent brute force attacks and the key exchange protocol relied on is oriented to the non-secure network environment.

Owner identification

When using HONOR Connect, it provides you with the ability to identify the owner and protect the privacy and security of your cross-device information. After the Owner Identification feature is turned on, the system will recognize your identity and protect your private information when you use HONOR Connect services on your device.

When you use Connected Call, Connected Notification, and other features on the tablet, the Owner Identification -feature will be used to identify you. If it is positive, information such as incoming call contact or message details will automatically appear on your tablet. If it is negative, information such as contact names and notification details will be hidden.

When you log in to your HONOR ID on devices (phones, tablets, etc.) using HONOR Connect and enable the Owner Identification feature on your phone, the biometrics for facial recognition enrolled in your phone will be securely mirrored to other connected devices in form of encryption, and will not be uploaded to the cloud. When you log out of your HONOR ID, turn off the Owner Identification feature, or delete biometrics for facial recognition on your phone, we shall simultaneously delete the biometrics stored on the device under the same account.

The biometrics for facial recognition used for the Owner Identification feature is only transferred between and stored on devices under the same account with near-field encryption, and HONOR will not collect your data or upload it to the cloud.

Service Security

This chapter describes the security protection of services supported by HONOR MagicOS products. For third-party payment apps, MagicOS can identify malicious apps and isolate the payment environment to ensure payment security.

HONOR ID

An HONOR ID can be used to access all HONOR services. Ensuring the security of HONOR ID and preventing unauthorized access are important concerns for users. To achieve this goal, HONOR requires users to use a strong, eight characters or more complex password that is not commonly used and that must contain 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.

When the user changes the password or uses the HONOR ID on a new device, HONOR system will send a text message, email, or notification to the user. If any exception occurs, HONOR will prompt users to immediately change their passwords. HONOR has also adopted various policies and procedures to protect users' HONOR 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.

Two-Factor Authentication

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

Account protection allows users to log in to their HONOR IDs using only their trusted devices. When attempting to log in from a new device, the user must enter the HONOR ID password and security verification code, which is automatically sent to the user's trusted phone number or displayed on the user's trusted device. If the new device passes verification, it will become the user's trusted device. This approach helps to enhance the security of HONOR IDs and associated HONOR ID services (such as HONOR Store and HONOR Club).

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 HONOR ID password, want to reset the password, or the phone number or email address bound to the HONOR ID is no longer available.

Account Risk Control

HONOR possesses an end-to-end risk identification mechanism and confrontation capabilities throughout the lifecycle of account management. Risk prevention is provided across the entire process of account registration, login, password reset, and account update. The system protects account security based on expert system, machine learning, and such technology as a variety of judgment factors integrated with multiple verifications of the login environment, to prevent malicious attacks on accounts and ensure the security of user assets and profile

Login Collaboration

During the OOBE period on a new device, a user can use the old device that has logged in to HONOR ID for collaborative login. The new and old devices will establish a secure near-field communication channel. After these devices carry out a series of verification of security parameters and trustworthy parameters, the account cloud sends a temporary credential to the new device for account login based on an HTTPS channel, achieving account collaborative login for the new and old devices.

Clone Login

The account-related information on an old device can be synchronized and cloned to a new device through a trusted secure transmission channel established using the clone function. The account cloud sends a temporary credential to the new device for account login based on an HTTPS channel, achieving account clone login for the new and old devices.

Fingerprint Login

On a new device, a user can enable the fingerprint login function after login with HONOR ID. The fingerprint login function is implemented based on the FIDO protocol. With a user’s fingerprint recorded on a device, the FIDO authenticator on the device will calculate the user’s fingerprint identifier

through obfuscation based on the fingerprint data and the keyID randomly generated during user registration. The follow-up authentication procedure will be performed based on the fingerprint identifier. HONOR devices will not store any fingerprint data or identifier of a user, and will only encrypt and store a user's keyID.

Scan for Login

On an old device that has logged in to HORNO ID, the user can scan the QR code of a new device for login. After that, the user needs to confirm the scan operation on the old device, and then HONOR ID will complete verification on the scan request through an HTTPS channel and on the cloud. In this case, the new device will receive a verification on scan for login, and the entire scan for login process is completed upon the user confirmation.

HONOR Cards

Transport Card

With this function, transport card companies can load their transport card apps to the secure element(SE) chip of a mobile phone through over-the-air download. After an association with a specified supplementary security domain (SSD), the personal data of a transport card will be downloaded and stored in the card app in the SE, and the associated SSD provides security guarantee. After a user adds a transport card, the user can perform operations, such as top-ups, viewing card information including the card number and balance, removing the transport card from the mobile phone and then storing on the cloud, migrating the transport card stored on the cloud back to the mobile phone, returning cards with card balance refunded when they are not used any more.

Card addition: After paying the fee for adding a transport card on the HONOR Cards app, a user can initiate a card addition request. HONOR's SEI TSM will create an independent SSD for the requesting transport card based on the Secure Channel Protocol (SCP) protection in the Issuer Security Domain (ISP), download and install the corresponding transport card application on the SSD in accordance with the GlobalPlatform Card (GP Card) regulations, and then transfer card instances to the SSD. The private key of the SSD will be managed by the SP TSM. The SP TSM downloads personal data, such as the card's private key, to the SE's transport card application through the SCP encrypted protection established using the SSD's private key. In this case, the card is successfully added to the mobile phone.

Top-up: A user can initiate a top-up request in the HONOR Cards app. Upon confirmation of a notification indicating the completion of item payment, the SP TSM sends the card on the SE a random number challenge value through the top-up initialization instruction. After receiving such a challenge value, the card performs calculation using its private key and returns the calculation result. The SP TSM then uses the private key of the card to verify the returned calculation result. If the calculation result is correct, the SP TSM verifies that the card is valid. Then, the SP TSM again uses the private key of the card to perform another round of calculation and encapsulates the calculation result into the card app of the SE downloaded in the top-up instruction. The card also needs to verify the calculation result. If the calculation result is correct, the card's validity verification on the SP TSM IS successful. In this case, the card will add the top-up amount to the balance storage area of the card. As the storages on two ends of the private key both deliver hardware-level security and no third party is aware of the private key, only the SP TSM of the transport card company can complete the top-up operation.

Over-the-air removal (migration): When a user temporarily does not need an added transport card, the user can remove it from the device, and the data about the removed card is still saved on the SP TSM. In the course of data backup from the transport card to the cloud, the SP TSM delivers a migration instruction to the card of the SE, and the card requests to obtain the corresponding data based on the instruction, encapsulates the data, and returns data after MAC encapsulation. Upon receipt of the result, the SP TSM checks the MAC address, decrypts data to obtain card data, and saves the data. The confidentiality and integrity of card data is ensured through card encapsulation and MAC encapsulation.

Card return: When a user's transport card is no longer needed, the user can initiate a card return request in the HONOR Cards app. During the card return procedure, the SP TSM obtains the card balance, and then the SEI TSM deletes the card from the SE chip permanently. The SP TSM will also return the obtained history records of card balance to the user's bank card for payment through the original payment path.

Car Key

A user can create a car key through a car manufacturer's app. The car manufacturer app calls the interface of the HONOR Cards app and notifies the car device of the Bluetooth device's MAC address. The HONOR Cards app calls the Bluetooth service interface to register for the car device's MAC

address. After scanning for Bluetooth broadcast of the car device, the Bluetooth service determines that the car device is within a valid instance and connects to the HONOR Cards app which then connects to the car manufacturer's app. The car manufacturer app service checks the key validity to complete car unlocking. If the Bluetooth is out of a valid instance, the car is automatically locked. Bluetooth connections comply with Bluetooth protocols, and car unlocking complies with security regulations of the car manufacturer. A user can choose to delete a car key in the car manufacturer's app or the HONOR Cards app. During deletion of a car key, the car device's MAC address which is registered to the Bluetooth service and the local data of the HONOR Cards app are both deleted.

HONOR Cloud

Data synchronization

HONOR Cloud provides data sync service and supports Multi-Device Collaboration. Users can sync data across multiple mobile devices to avoid data loss.

Features such as Calendar, Notepad, Contacts, Password Vault, and WLAN, currently adopt HONOR Cloud SDK to sync data. Password Vault employs end-to-end encryption to sync data. The data will be encrypted again on the Cloud side with an HMACSHA256 signature to ensure its security.

Before syncing the structured data, HONOR Cloud will establish an encrypted channel with mobile terminals based on TLS1.2 protocol, adopt the ECDH algorithm to exchange the session key, and use the AES-GCM-256 algorithm for encryption. HONOR Cloud will assign a unique key to each user and use it to encrypt the data before storage. The key will be protected by the HONOR Cloud key, which is kept in the KMS system through hardware mechanisms.

For unstructured data (images, text, etc.), a unique master key will be created on the device. When processing a large file, each data chunk will be encrypted by a corresponding subkey derived from the master key. When syncing data to the Cloud, the master key will be encrypted and uploaded to HONOR Cloud using the work key. The work key can be encrypted and distributed with the temporary session key generated by the ECC algorithm. The terminals use an asymmetric key provided by HUKS to encrypt the work key.

Data backup

HONOR Cloud backs up application data (such as the contact, calendar, recording, information, and call records), system settings (such as the input method, alarm, clock, camera, and optimizer settings), and some third-party apps data. The encryption mode of backup files is consistent with that of the synchronized unstructured data.

App Market

Developer Real-Name Verification

In accordance with related laws and regulations, HONOR App Market will perform real-name verification on the developers to which on-the-shelf apps belong to. Developers can be categorized into individual developers and enterprise developers, both of which enjoy various types of capabilities and services opened by HONOR only after they complete real-name verification. This is to ensure that the on-the-shelf apps are fully compliant with laws and regulations and are traceable, aiming to promote healthy app ecosystem development.

App security control

HONOR App Market is committed to providing users with secure and reliable applications which are compliant with privacy requirements. Upon an application of an on-the-shelf app, HONOR App Market performs all-the-around security scanning and manual audits on items such as the app function, permissions, content, and payment. Such scanning covers diversified auditing requirements, including virus detection, principle of least privilege, ad blocking, rogue behavior, and personal data collection. After an app goes to the shelf, HONOR App Market will periodically develop inspection on the app and promotes developers to modify or delete the app if it does not meet requirements or is against public sentiment, or a corresponding user feedback or complaint is received. Through audits against apps to be put on the shelves, routine inspection on the on-the-shelf apps, and removal of unsatisfied apps from the shelves, apps are strictly controlled to protect users' lawful interests.

Protection of Minors

HONOR App Market provides well-defined mechanism for protecting minors, with all applications being classified into diversified levels, aiming to offer users in different ages with healthy and appropriate services. After identity verification on minors through accounts, parents can manage the viewing

content and duration and restrict functions such as app download, in-app consumption, and personalized recommendation. This aims to ensure content security and protects the physical and mental health of minors.

Developer Kit Security

HONOR Developers provides a number of kit frameworks to enable third-party developers to extend and enrich app capabilities on HONOR terminal devices.

Apps that want to access HONOR Accessible Services need to register on the HONOR Developers platform and fill in the unique app identifier and the signing certificate fingerprint. When the registration is approved, developers can apply for the permission scopes of Accessible Services.

To avoid the abuse of Accessible Services, the background service of accessible service frameworks on the terminal side will verify the unique app identifier, signing certificate fingerprint, and whether the accessible services requested by the third-party app are within the applied permission scopes.

Find Device & Activation Lock*

MagicOS provides the Find Device function. If your HONOR phone or tablet is lost or stolen, you can log in to the official website of HONOR Find Device (https://cloud.hihonor.com/findmydevice/wapFindPhone) or the Find Device app to find your lost device. The following functions can help you find your lost device, protect the data in your device, and protect your privacy.

Locate device: You can display the location of your device on the map. Including active location and automatic location reporting at a low battery level.

Play ringtone: The device will play the alert ringtone at maximum volume regardless of whether the device is in silent or vibration mode.

Turn-off verification password: When the function is turned on, the lock screen password will be required to turn off the device from the lock screen to avoid the device from getting turned off by the finder.

Remote internet connection: If the device is offline when you use the Find Device feature, Find Device will help you remotely turn on the mobile data of the lost device so that you can locate it.

Lost mode: The device's screen will be locked and enters a super power-saving mode, displays message and contact number on the screen, automatically reports the location, and displays the message when an internet connection is established. At the same time, the device will hide the contact information of the incoming call and the content of new text messages.

SIM card locking: You can lock the SIM card on the device after entering the lost mode. After locking, when the SIM card is inserted into other devices or the device is restarted, a password will be required before use.

Erase data: Restore the device to factory settings and permanently erase all data (including the storage card). You can still locate the device after erasing data, and your HONOR ID password will still be required when using the device.

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

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

*Note: This feature is not supported in overseas regions at this time.

HONOR Health

Based on phones and wearables, HONOR Health provides services to record, sync, securely store, and authorize exercise and health data. Protecting user data security is the cornerstone of HONOR Health.

During mobile terminal services and data exchange, we protect users' exercise and health data security in a comprehensive way based on the data security capability provided by MagicOS and the encryption capability provided by hardware security and trusted execution environment.

All exercise and health data are transferred through a secure channel (TLS1.2 and above). On this basis, we use the ECC algorithm to encrypt personal data again and attest the signature during transfer to protect the data from theft and tampering.

All personal data users voluntarily upload to Health Cloud is encrypted and stored with the AES-256-GCM algorithm and the key is kept in the KMS system through hardware mechanisms so as to ensure the security of personal data storage, usage, and destruction.

In addition, based on the security capability of HONOR ID and device interconnection, users can sync exercise and health data safely and conveniently across devices logging into the same HONOR ID via Health Cloud.

Payment Protection Center

The payment protection center provides an independent secure environment for payment-related applications. The payment protection center ensures that payment apps are from official sources. It also strictly

controls the interaction between payment apps and outside apps to reduce the risk of malicious calls and attacks on apps within the protection center. Moreover, the system will test payment apps when they are running to ensure the security of the operating environment and protect user transactions and property.

MDM API*

MagicOS now provides a device management SDK for HONOR mobile devices. It enables policy configuration, access control, and device management functions for enterprise mobile offices or industry-specific mobile device management (MDM) applications.

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

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

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

*Note: Only available on China models.

Privacy Protection

This chapter describes MagicOS's user privacy protection. HONOR devices may contain user privacy data, such as contacts, short messages, and photos. To protect user privacy, MagicOS ensures that pre-installed apps fully meet privacy compliance requirements, and provides app permission management, notification management, location-based service (LBS), 7-day privacy access history and other privacy management functions. To further protect users' privacy, MagicOS provides the device identifier system, differential privacy, and other technical privacy protection means.

Permission management

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

Phone

SMS

Contacts

Call log

Camera

Location

Microphone

Calendar

Body sensors

Health

Photos and videos

Music and audios

Documents and files

MMS

Using call transfer (CT)

Floating window

Creating desktop shortcut

Notifications

Device app list

Nearby devices

Health data

MagicOS 5.0 provides more detailed controls when an app requests to use the Camera, Microphone, and Location permissions by offering the “allow only while in use” and “allow this time” choices. If the user chooses the “allow only while in use” option, granted permissions will be removed once the app is switched to the background. On the other hand, permissions will be removed once an app is switched to the background or stopped by selecting the “allow this time” option and must be granted again during the next use.

File Access Permissions

To further enhance the security of files stored in the public storage area, in MagicOS 7.2, three types of permissions are specifically granted for photos and videos, music and audios, and documents and files, respectively. MagicOS 8.0 supports selection of desired photos and videos for access permission granting. In this way, users can grant access permission to only desired photos and videos, instead of the whole media library.

Privacy Report

Mobile apps have become increasingly more powerful thanks to explosive developments. To make services richer, apps also require a myriad of permissions that frequently touches upon user privacy. However, due to the lack of effective control and supervision, some apps can obtain private user data without their knowledge. Even if information disclosure was to occur, there would be no records to resort to.

MagicOS 5.0 systematically records information such as app name, accessed permission, time, and outcome each time an app tries to access sensitive information. From the home screen, users can go to Settings > Privacy and view the top 5 apps that accessed the Location, Camera, Microphone, Contacts, and Storage permissions. Records are presented according to the time, app and permission for easy viewing and management.

MagicOS 6.0 implements detailed logging of apps' access to stored data. In the privacy access log, the stored data is subdivided into four types: image, audio, video, and other files, and four types of operations are recorded: read, create, modify, and delete. When images and video files are deleted by a three-party app, the system will move the deleted files to the recycle bin of the gallery and notify the user to protect the security of his/her data assets.

Audio/Video Recording Reminder

To prevent malicious apps from obtaining permission to access the microphone or camera through spoofing and recording audio or videos at the backend without users' knowledge, MagicOS 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 tap the close button to close the app that is recording audio or a video.

Location Service

MagicOS allows a user to enable or disable location access in Settings. After location access is disabled, MagicOS also disables the Global Positioning System (GPS), Wi-Fi, Bluetooth, and mobile base station positioning. In this way, users' location service is 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.

MagicOS 5.0 further provide users with the coarse location features in addition to precise and approximate locations. To prevent tracking, users can decide whether an app can only obtain coarse location data instead of fine location by reducing the positioning accuracy.

Clear clipboard automatically

During daily use of the phone, some important information will inevitably enter the clipboard, such as shipping information, phone numbers, email addresses, and even passwords. To protect the user's private information from leakage, the clipboard will be automatically cleared in 15 minutes every time it is updated. When an app reads the clipboard, the system reminds the user of this behavior through a message.

Device Identifier

During system processing, unique identification is required. MagicOS provides multiple unique identifiers with different behavior features.

The app selects the most appropriate identifier based on different scenarios. These features involve privacy.

Scope

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

Single App: The ID is only available to the app and cannot be accessed to any other apps.

App group: The ID is available to a group of apps, such as a group of apps provided by the same app developer.

Device: All apps installed on the device access a same ID.

Resettability and Durability

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

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

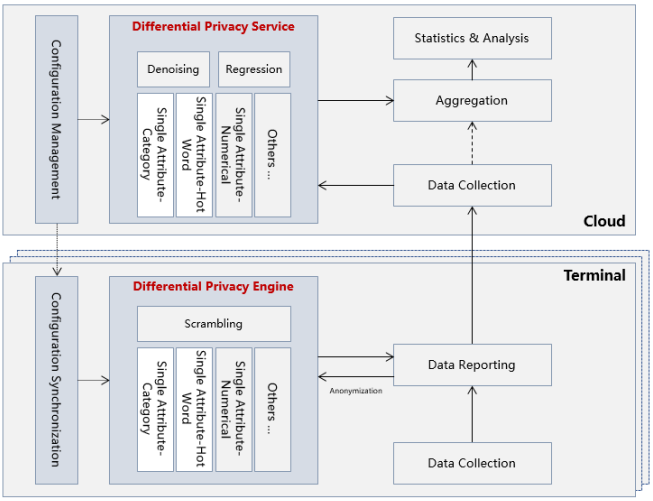
The MagicOS identifier system includes:

ID Type	ID Name	Application Scenario & Scope	Generation Time	Reset
Resettability	UUID	Used in scenarios where apps are associated with random identifiers.	A random number is generated each time an identifier is invoked.	The UUID is regenerated each time an identifier is invoked.
User ID	HONOR ID	Used for HONOR Cloud service features.	Generated upon creation of an HONOR ID.	Deleted upon deregistration of an HONOR ID.

Differential Privacy

With the rapid development of information technology and the continuous improvement of personalized services, MagicOS provides users with intelligent recommendation services. After obtaining users' consent, we will collect some user data and usage information to improve the experience of intelligent recommendation services. We adopt differential privacy

technology to secure user privacy. Add random noise to the data shared and uploaded to the Cloud to avoid the Cloud from obtaining personal data. The noise-added data in the Cloud can produce relatively accurate statistical information after data denoising and aggregation so as to support improving the experience of intelligent services. HONOR cannot restore the user's original data from the noise-added data.



Differential Privacy Platform

On MagicOS 7.0, we provide differential privacy protection based on the Copy & Go feature of YOYO Suggestions. The main processes are as follows:

1. Add noise to personal data: Copy & Go provides different features (e.g., make a phone call, send SMS messages, copy, share, etc.) according to text types identified by the system (phone number, tracking number, email address, etc.). To improve user experience, the Cloud needs to collect information on what features users select. The phone will add noise to users' feature usage record when reporting it to ensure that personal data cannot be distinguished when shared to the Cloud and the potential user privacy (usage and habits) is protected.

2. Denoising and regression: The Cloud side needs to process the feature selection record of different users through data denoising, aggregation, and regression to get to know what features the user group tends to select according to specific text types in a relatively accurate manner. The selection tendency of a certain user cannot be inferred and distinguished. The statistical information will be used to improve the user experience of Copy & Go AI algorithm of YOYO Suggestions.

3. Privacy budget: The shared personal information will be scrambled with different intensities according to its sensitivity level so as to remove the concern of data sharing. The data scrambling intensity refers to the privacy budget. The more sensitive the privacy, the less the privacy budget allocated and the higher the data scrambling intensity. Besides, the privacy budget is also an important factor in terms of data sharing frequency (e.g., daily, monthly). The budget accrues linearly with frequency. Copy & Go of YOYO Suggestions sets privacy budget epsilon to 4 referring to the industry level.

4. User perception: When personal data is shared to the Cloud, the biggest concern is if the shared data itself or any third-party information can be inferred to a certain user, resulting in the disclosure of personal (private) data. However, differential privacy uses technical means to ensure that the data shared to the Cloud is non-personal data from the very beginning on the phone side.

In addition to differential privacy, the privacy computing platform also adopts federated learning, Secure Multi-Party Computation, TEE-based confidential computing, and other technologies to protect user privacy. With further development of these technologies in the future, the user experience will continue to improve according to various application scenarios and areas under the premise of secured user privacy.

Privacy Statement

MagicOS 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 MagicOS released in the local countries.

Refer to the privacy statement as follows:

https://www.hihonor.com/privacy-policy/worldwide/

Conclusion

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

While providing security solutions, HONOR also attaches great importance to establishing security process and capabilities, which are vital for implementing security management of products throughout the lifecycle. HONOR has set up a Security Response Center (SRC) dedicated to improving product security. Any organization or individual that finds security vulnerabilities in HONOR products can contact HONOR at security@hihonor.com. HONOR SRC will reply promptly while organizing internal vulnerability fixing, releasing vulnerability warning, and pushing patches for update. HONOR is sincere in its willingness to jointly build HONOR device security with all stakeholders.

Acronyms and Abbreviations

List of abbreviations

English Abbreviations	English
2D	Two Dimension
3D	Three Dimension
3DES	Triple Data Encryption Standard
AES	Advanced Encryption Standard
AI	Artificial Intelligence
API	Application Programming Interface
APK	Android application Package
ARM	Advanced RISC Machines
ASLR	Address Space Layout Randomization
BLE	Bluetooth Low Energy
BTI	Branch Target Identification
BYOD	Bring Your Own Device
CA	Certificate Authority
CC	Common Criteria
CE	Credential Encryption
CFI	Control Flow Integrity
CNN	Convolutional Neural Network
DE	Device Encryption
DEP	Data Execution Prevention
CMP	Certificate Management Protocol
DAC	Discretionary Access Control

DRM	Digital Rights Management
ECB	Electronic Code Book
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic Curve Digital Signature Algorithm
EAP	Extensible Authentication Protocol
eID	electronic IDentity
EMM	Enterprise Mobility Management
eMMC	Embedded Multimedia Card
MagicOS	MagicOS
GP	GlobalPlatform
GSM	Global System for Mobile Communications
HKIP	HONOR Kernel Integrity Protection
HMAC	Hash-based message Authentication Code
HOTA	HONOR Over The Air
HTEE	HONOR Trusted Execution Environment
HUK	Hardware Unique Key
HUKS	HONOR Universal Keystore
ID	Identifier
IMEI	International Mobile Equipment Identity
IOT	Internet of Things
IPSec	Internet Protocol Security
IT	Information Technology
JOP	Jump Oriented Programming
L2TP	Layer Two Tunneling Protocol
LKM	Loadable Kernel Module
LSM	Linux Security Module
LTE	Long-Term Evolution

LTO	Link Time Optimization
MAC	Mandatory Access Control
MAC	Media Access Control
MDM	Mobile Device Management
MPPE	Microsoft Point-to-Point Encryption
NFC	Near Field Communication
NIST	National Institute of Standards and Technology
NPU	Neural Processing Unit
OS	Operating System
OTA	Over The Air
P2P	Peer to Peer
PA	Pointer Authentication
PAN	Privileged Access Never
PIN	Personal Identification Number
PKI	Public Key Infrastructure
POS	Point of Sales
PPTP	Point-to-Point Tunneling Protocol
PRNG	Pseudo-Random Number Generator
PSK	Pre-Shared Key
PXN	Privileged eXecute Never
REE	Rich Execution Environment
ROM	Read-Only Memory
ROP	Return Oriented Programming
RSA	Rivest Shamir Adleman
RPMB	Replay Protected Memory Block
SCEP	Simple Certificate Enrollment Protocol
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
SN	Serial Number
SOTER	Standard Of authentication with fingerprint
SRC	Security Response Center
SSL	Security Sockets Layer
TA	Trusted Application
TEE	Trusted Execution Environment
TLS	Transport Layer Security
TSM	Trusted Service Manager
TUI	Trusted User Interface
UDID	Unique Device Identifier
UID	User Identifier
UUID	Universally Unique Identifier
VM	Virtual Machine
VPN	Virtual Private Network
WAPI	WLAN Authentication and Privacy Infrastructure
WEP	Wired Equivalent Privacy
WLAN	Wireless Local Area Network
WPA	Wi-Fi Protected Access
WPS	Wi-Fi Protected Setup

Modification Records

Date	Description
2021-08-12	First released
2022-03-08	MagicOS 6.0 version update
2022-11-18	MagicOS 7.0 version update
2024-01-10	MagicOS 8.0 version update