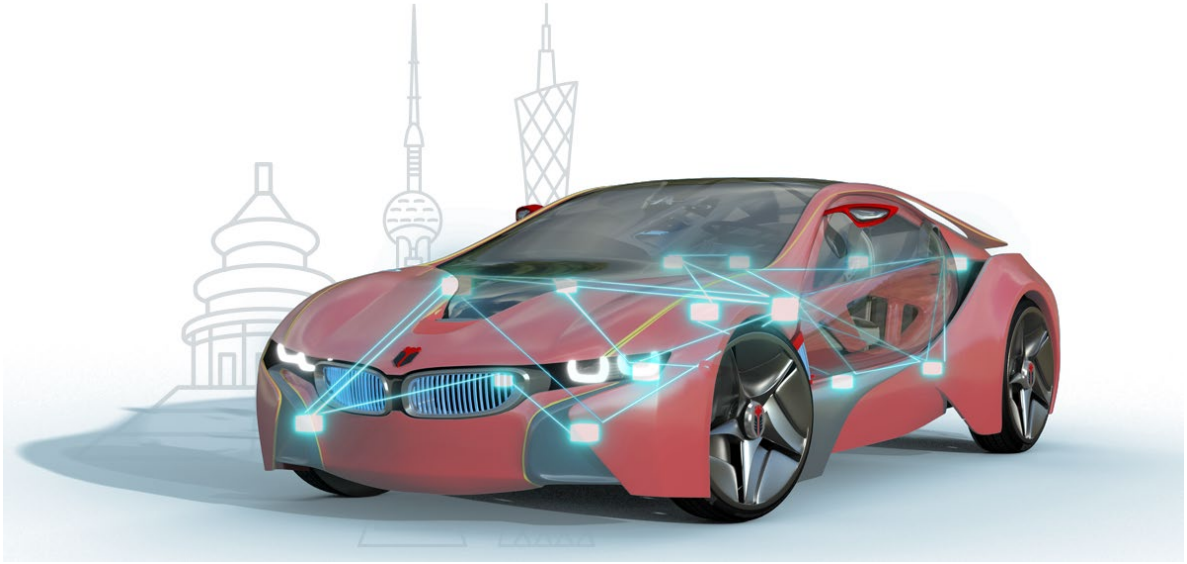
知从木牛英飞凌 TRAVEO CYT4BB 信息安全应用介绍

Application of ZC.MuNiu CyberSecurity on Infineon Traveo CYT4BB

知从木牛基础软件平台信息安全库

ZC.MuNiu Basic Software Platform CryptoLibrary



知从木牛英飞凌 TRAVEO CYT4BB 信息安全应用介绍

CYBERSECURITY APPLICATION OF ZC.MUNIU ON INFINEON TRAVEO CYT4BB

知从木牛基础软件平台信息安全库

ZC.MuNiu Basic Software Platform CryptoLibrary

1 CYT4BB 介绍及信息安全应用 INTRODUCTION OF CYT4BB AND CYBERSECURITY APPLICATIONS

CYT4BB TRAVEO™ T2G 32 位汽车 MCU 面向高端车身控制单元等汽车系统。CYT4BB 具有两个 Arm® Cortex®-M7 CPU（用于主处理）和一个 Arm Cortex-M0+ CPU（用于外设和安全处理）。其中 CM0 核集成了多种硬件算法功能，包括：AES、CHACHA、CMAC、CRC、DES/TDES、SHA1/SHA2/SHA3、HMAC、TRNG/PRNG、RSA 等多种算法，基于此可以实现多种信息安全应用的扩展，如：安全启动、安全升级、安全诊断等功能。

CYT4BB TRAVEO™ T2G 32-bit automotive MCU is targeted at automotive systems such as high-end body control units.The CYT4BB features two Arm® Cortex®-M7 CPUs (for main processing) and one Arm Cortex-M0+ CPU (for peripheral and security processing). Among them, the CM0 core integrates a variety of hardware algorithm functions, including: AES, CHACHA, CMAC, CRC, DES/TDES, SHA1/SHA2/SHA3, HMAC, TRNG/PRNG, RSA, and many other algorithms, based on which it can be implemented to extend a variety of information security applications, such as: SecureBoot, SecurUpdate, SecureDiagnostics other functions.

此外，知从木牛信息安全库在原有的硬件算法功能基础上，集成了多种软件算法及扩展应用，如：SM2/SM3/SM4、ECDSA 256R1、RSASSA-PKCS-v1_5(4096bits Key)、RSASSA-PSS-(4096bits Key)、SHE/USERKEY Load、GetUid、KDF、DebugHandling 等，实现了安全存储的功能。

In addition, ZC Muniu CryptoLibrary integrates a variety of software algorithms and extended applications based on the original hardware algorithm function, such as: SM2/SM3/SM4, ECDSA 256R1, RSASSA-PKCS-v1_5(4096bits Key), RSASSA-PSS-(4096bits Key), SHE/ USERKEY Load, GetUid, KDF, DebugHandling, etc., to achieve secure storage.

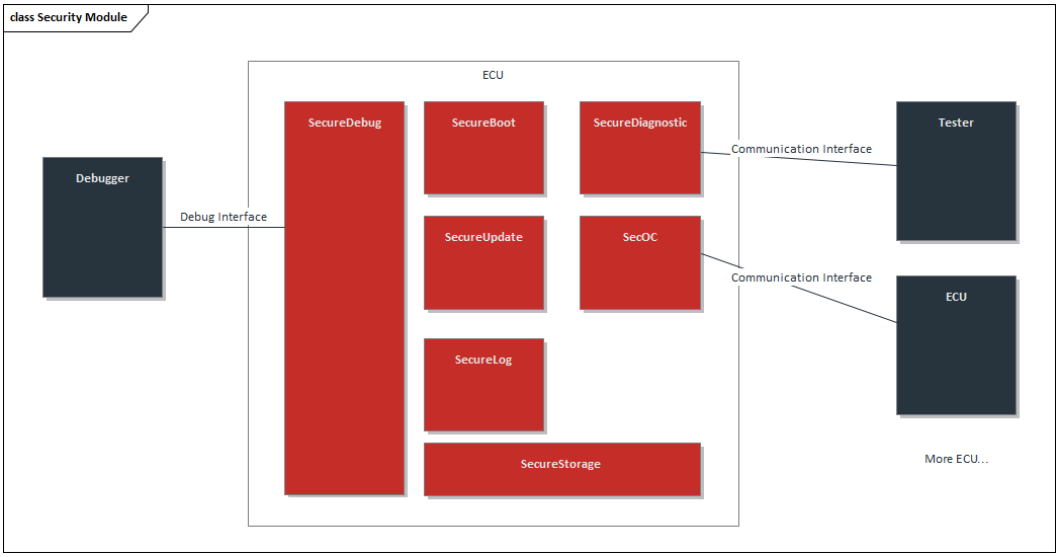


FIGURE 1 CRYPTOLIBRARY ARCHITECTURE

1.1 通信机制 Communication Feature

通信机制：IPC 硬件包含 IPC 通道和 IPC 中断的寄存器结构，其通道寄存器实现 Lock/Release 机制和消息传递。IPC 的中断结构寄存器为消息传递事件和 Lock/Release 事件生成对每个 CPU 的中断。IPC 通道结构中的 ACQUIRE 寄存器提供 Lock 特性，IPC_STRUCTx_LOCK_STATUS 表示 Lock 状态。IPC_STRUCTx_NOTIFY 寄存器生成通知事件，IPC_STRUCTx_RELEASE 寄存器释放 IPC 通道结构并生成释放事件。

The IPC hardware contains register structures for IPC channel and IPC interrupt. IPC channel registers implement lock/release mechanisms, and messaging. IPC interrupt structure registers generate interrupts to each CPU for messaging events and lock/release events.The IPC channel structure ACQUIRE register provides lock feature and IPC_STRUCTx_LOCK_STATUS indicates lock status. The IPC_STRUCTx_NOTIFY register generates notification event, the IPC_STRUCTx_RELEASE register releases IPC channel structure and generates release event.

CM0 核通过安全启动释放 CM7 核后，CM7 调用 IPC 通道及中断初始化，并调用 Cy_Crypto_Enable 通知 CM7 核进行 IPC 通道及相关中断初始化。当两边初始化均完成后，方可调用其他算法任务进行处理。

After the CM0 core releases the CM7 core through secure boot, the CM7 calls IPC channel and interrupt initialisation, and calls Cy_Crypto_Enable to notify the CM7 core to initialise the IPC channel and related interrupts. When both initialisations are complete, other algorithmic tasks can be invoked for processing.

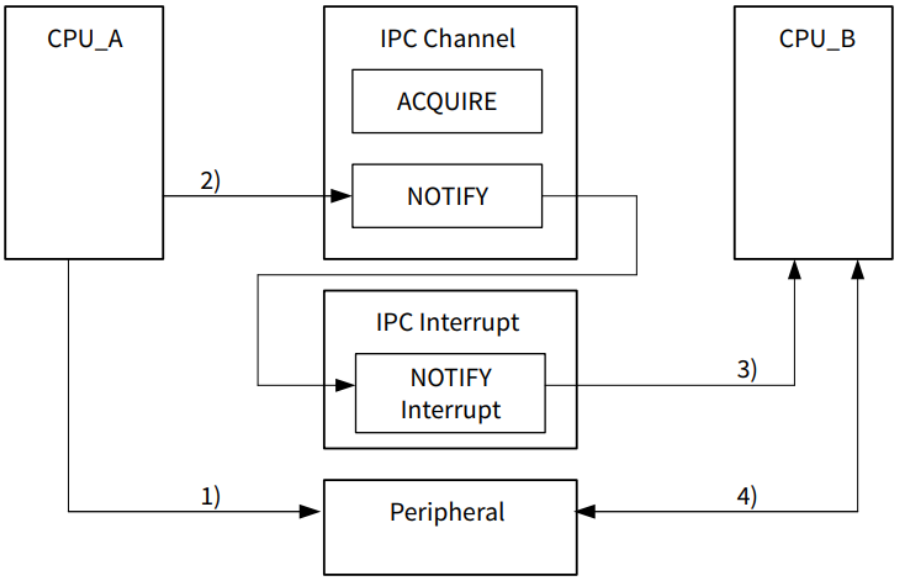


FIGURE 2 IPC COMMUNICATION

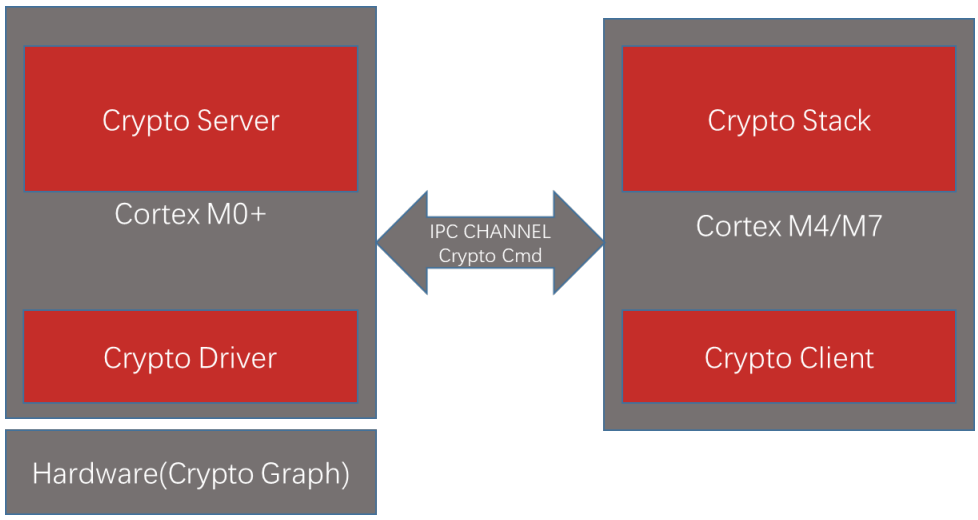


FIGURE 3 CM0/CM7 COMMUNICATION

2 安全启动 SECURE BOOT

安全启动（SecureBoot）是 MCU 的基本功能，通过硬件加密模块来实现，该机制必须独立于用户程序运行，不能被破坏。作为整个安全启动信任链的基础，安全启动主要用于在 MCU 启动之后，用户程序执行之前，对用户定义的 Flash 中关键程序的数据完整性和真实性进行验证，确定是否被篡改。如果验证失败，说明 MCU 处于不可信的状态，部分功能甚至整个程序不能运行。

Secure Boot is a fundamental function of the MCU, implemented through hardware encryption modules. This mechanism must operate independently of user programs and cannot be compromised. As the foundation of the entire secure boot trust chain, Secure Boot is mainly used to verify the integrity and authenticity of key programs defined by users in Flash memory

after the MCU starts and before user programs execute, to determine if they have been tampered with. If the verification fails, it indicates that the MCU is in an untrusted state, and some functions or even the entire program cannot run.

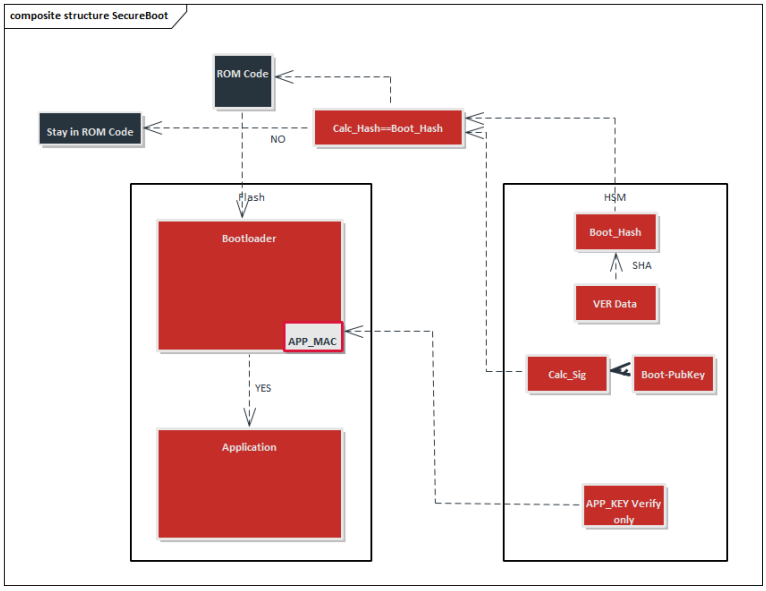


FIGURE 4 SECURE BOOT

CYT4BB 采用 RSASSA-PKCS-v1_5(2048bits Key)的方式，于 CM0 核上电启动时读取预先存储在 SFLASH 中的 Public Key 和 Code Flash 中的 Signature 值，通过集成在 SFLASH 中的预设 RSA Verify 接口，计算得出 Boot_Hash 和 Calc_Hash，比较后得出安全启动信任根的可靠性。之后启动 CM7 核进行安全启动信任链的校验。通过信任根代码的 root 对第一阶段引导程序进行验证，验证成功则可通过此验证有效的软件执行并继续验证后续引导阶段软件有效性。

CYT4BB uses RSASSA-PKCS-v1_5 (2048bits Key) to read the Public Key pre-stored in SFLASH and the Signature value in Code Flash when the CM0 core is powered on and started up, and then through the pre-programmed RSA Verify interface integrated in SFLASH, it computes the Boot_Hash and Calc_Hash, which are compared to determine the reliability of the secure boot root of trust. Afterwards, the CM7 core is started for the verification of the secure boot trust chain. The first stage bootloader is verified by the root of trust code, and successful verification allows this to verify valid software execution and continue to verify the validity of software in subsequent boot stages.

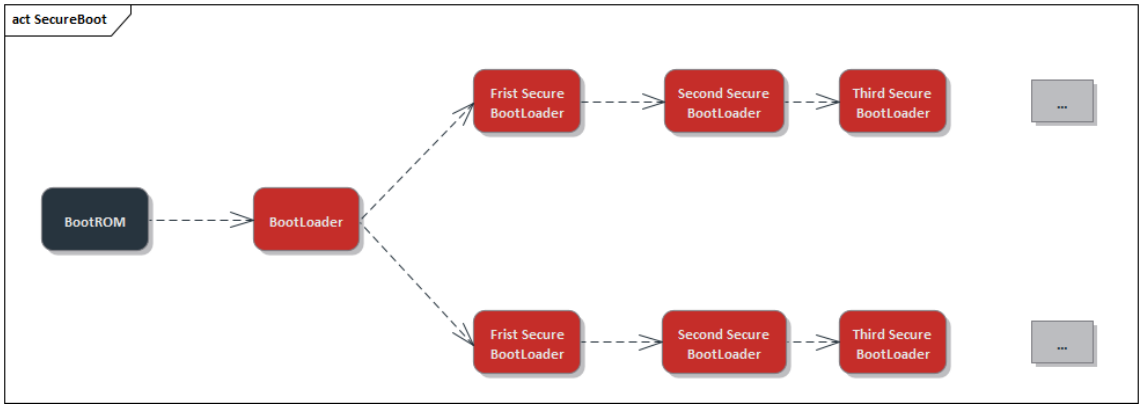


FIGURE 5 SECURE BOOT ROUTINE

3 安全诊断 SECURE DIAGNOSTICS

安全诊断（Secure Diagnostic）是保护 ECU 内部数据安全的重要手段，主要用于将程序或数据下载 / 上传到服务器以及从服务器读取特定内存位置的诊断服务需要进行身份验证。异常的程序上传或下载到服务器的数据可能会潜在地破坏电子设备或其他车辆部件，或可能违背车辆的排放或安全等标准。另一方面，当从服务器检索数据时，可能会违反数据安全性。因此需在这些服务执行前，要求上位机证明其身份，在合法身份确认之后，才允许其访问数据和诊断服务。

Secure Diagnostic is an important means of protecting the internal data security of ECUs (Electronic Control Units). It is primarily used for diagnostic services that require identity verification when programs or data are downloaded/upload to a server and when specific memory locations are read from the server. Unusual program uploads or downloads to the server could potentially damage electronic devices or other vehicle components, or may violate vehicle emission or safety standards. On the other hand, when retrieving data from the server, data security could be compromised. Therefore, it is necessary to require the upper computer to prove its identity before executing these services, and only after legal identity confirmation is allowed to access data and diagnostic services.

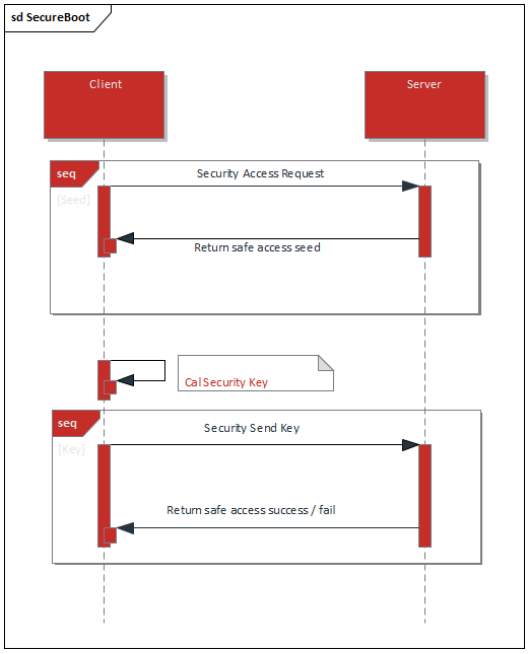


FIGURE 6 SECURE DIAGNOSTIC

CYT4BB 采用可使用 TRNG\PRNG、AES128\256、AES-CMAC、HMAC 等多种硬件加密算法机制，来确认客户端的身份，并决定客户端是否被允许访问。其中，对 AES、CMAC、HMAC 等算法所使用的密钥，皆可使用 LoadKey 接口进行预先存储，通过 KeyID 调用的方式在 CM0 侧进行处理，保证了身份验证信息的可靠性。

CYT4BB uses various hardware encryption algorithms such as TRNG/PRNG, AES128/256, AES-CMAC, HMAC, etc. to confirm the identity of the client and decide whether the client is allowed to access. The keys used for AES, CMAC, HMAC and other algorithms can be stored in advance using the LoadKey interface, and processed on the CM0 side by means of KeyID call, which ensures the reliability of the authentication information.

4 安全升级 SECURE UPDATE

随着越来越复杂的网络环境，在软件升级更新过程中，保证升级包的发布来源有效、不被篡改、数据不丢失以及升级内容不被恶意获取变得越来越重要。传统升级过程的升级包数据基本上是以明文传输，数据校验方式也是安全性较低的 CRC 算法。

As network environments become increasingly complex, ensuring that the release source of upgrade packages is valid, not tampered with, data is not lost, and upgrade content is not maliciously obtained during the software upgrade process is becoming more and more important.In traditional upgrade processes, the upgrade package data is essentially transmitted in plaintext, and the data verification method is also a less secure CRC algorithm.

而 CYT4BB 一方面可采用其 CM0 核内部自带的 RSASSA-PKCS-v1_5(2048/3072bits Key)算法，也可采用知从木牛所集成 CryptoLibrary 中的 Ed25519、ECDSA 256R1、RSASSA-PKCS-

v1_5(4096bits Key)、RSASSA-PSS-(4096bits Key)等，可满足用户的多种算法使用需求。并且，对于其中非对称算法所使用的 PublicKey、PrivateKey，同样可以使用 LoadUserKey 接口，将其预先存储于 SFLASH 指定位置，并通过 KeyID 进行调用，保证了安全升级过程的真实性和完整性。

On the one hand, CYT4BB can adopt the RSASSA-PKCS-v1_5(2048/3072bits Key) algorithm that comes with its CM0 core, and on the other hand, it can also adopt Ed25519, ECDSA 256R1, RSASSA-PKCS-v1_5(4096bits Key), RSASSA-PSS-(4096bits Key), and so on, which can satisfy the user's needs for multiple algorithms. Key), RSASSA-PSS-(4096bits Key), etc., which can meet the user's needs for multiple algorithms. Moreover, for the PublicKey and PrivateKey used by the asymmetric algorithms therein, the same LoadUserKey interface can be used to pre-store them in the specified location of SFLASH and invoke them through the KeyID, which ensures the authenticity and integrity of the security upgrade process.

此外，CYT4BB 也支持证书存储，可调用 CM0 核 Srom 相关接口，将所需证书存储于 Work Flash 相应位置，通过 X.509 证书解析调用接口并配置对于算法，实现基于证书的安全升级机制。

In addition, CYT4BB also supports certificate storage, which can call CM0 core Srom related interfaces to store the required certificates in the corresponding location of the Work Flash, and through the X.509 certificate parsing call interface and configure the algorithms for the implementation of certificate-based security upgrade mechanism.

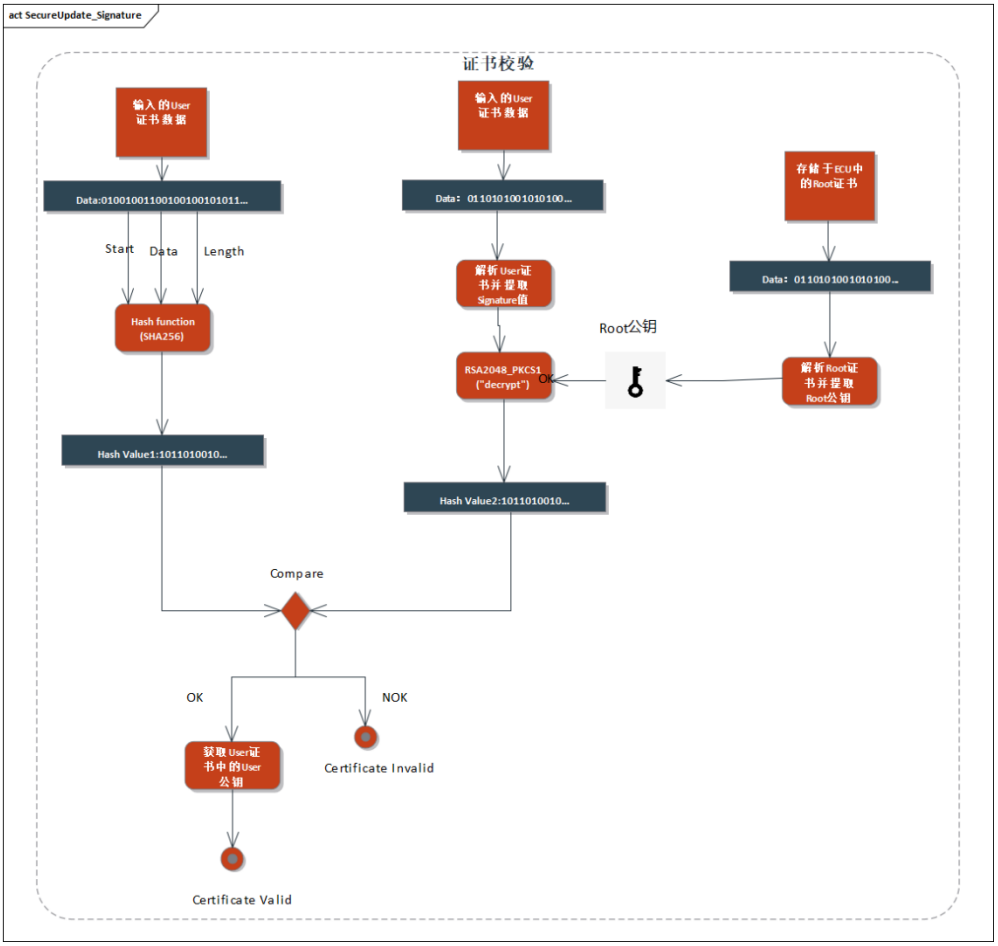


FIGURE 7 SECURE UPDATE WITH CERTIFICATE

5 安全存储 SECURE STORAGE

安全存储可保护数据区域内容不被异常窃取，避免因为控制器通过强行访问数据存储区，将存储的密钥，证书等内容进行复制。目前主流芯片都可通过设置 Flash，Nvm，RAM 等存储区进行数据保护，开启此功能可有效避免上述情况产生。

Secure Storage can protect the contents of data areas from being stolen abnormally, preventing the copying of stored keys, certificates, and other content due to forced access to the data storage area by controllers. Currently, mainstream chips can protect data by setting up Flash, Nvm, RAM, and other storage areas. Activating this feature can effectively prevent the aforementioned situations.

CYT4BB 则通过在内部专门划定一块用于密钥与用户信息存储的 SFLASH 区域，实现了安全存储的基础。在此基础上，知从木牛按照 AUTOSAR_TR_SecureHardwareExtensions 规范要求，实现 SHE Key 的安全存储，对存入 SFLASH 中的密钥进行加密保护，保证了安全启动，安全诊断、安全升级等功能的可靠性。

CYT4BB achieves the foundation of secure storage by designating an internal SFLASH area for key and user information storage. On this basis, CYT4BB implements the secure storage of SHE key in accordance with AUTOSAR_TR_SecureHardwareExtensions specification, and encrypts and protects the key deposited in SFLASH to ensure the reliability of secure startup, secure diagnosis, and secure upgrade functions.