Whitepaper
09-2020

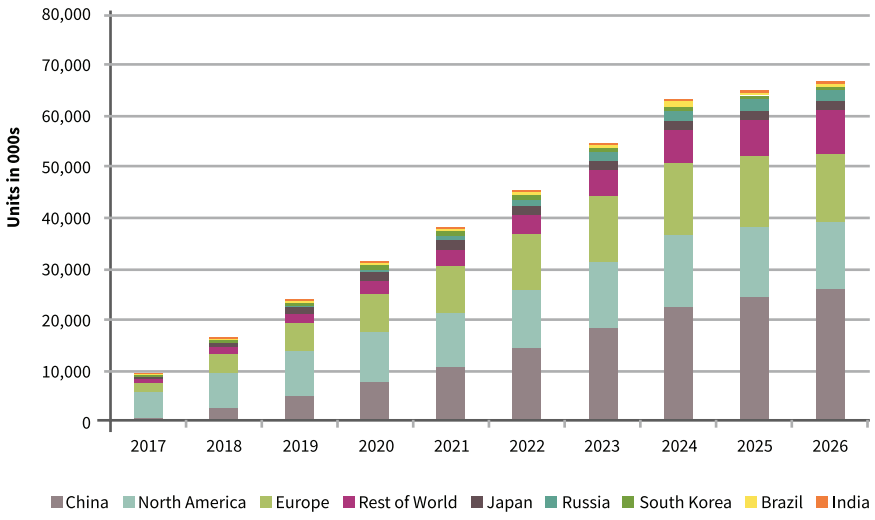
Over-the-air (OTA) Updates for Microcontroller-based Automotive Systems

Masayoshi Kusumoto, Director, Strategic Marketing, Automotive Division

Today's advanced automobiles may contain upwards of 100 million lines of code, more than a typical airplane. Automotive systems monitor the car's operation, automate certain functions and implement entertainment systems. Deploying new features, delivering maintenance releases and addressing security concerns is a daunting challenge, especially throughout the relatively long lifespan of a standard car. OTA (Over-the-Air) updates now play an important role, and manufacturers are developing cost-effective technology infrastructures to deliver updates securely and reliably. These solutions ultimately contribute to the comfort, safety and reliability of the cars we drive.

According to Strategy Analytics, most vehicles produced in the near future will be equipped with LTE network connectivity.

Telematics Node Shipments, LTE Network Type by Region, 2017-2026



Source: Strategy Analytics

OTA updates are common with infotainment systems, but less so with powertrain, chassis, body and safety applications. These applications call for higher reliability and safety requirements, as well as the additional processing and memory demands of an OTA update. However, as the amount of software in vehicles increases in the next few years, the number of software-related recalls will increase as well. Costing automakers tens of millions of dollars already, according to Strategy Analytics, large-scale software recalls can be avoided with OTA update systems in place. Therefore, future wide-scale adoption of automotive OTA updates can provide automakers with immense return-on-investment.

Infotainment systems generally run on high-performance processors with rich external memories, while several other applications utilize microcontrollers. The demands of an OTA update extend beyond simply flashing memory with new code. They also require that the system has the capability to:

- Present its identity
- Verify its own integrity
- Establish a secure connection with the server
- Verify the integrity and authenticity of the update received
- Protect stored data from cyber-attacks
- Download and program an update, regardless of its operating state
- Roll back the update in case of a failure

In short, the client system must ensure that the right update is securely received and stored, functioning without disrupting the operation of the vehicle.

Secure Updates

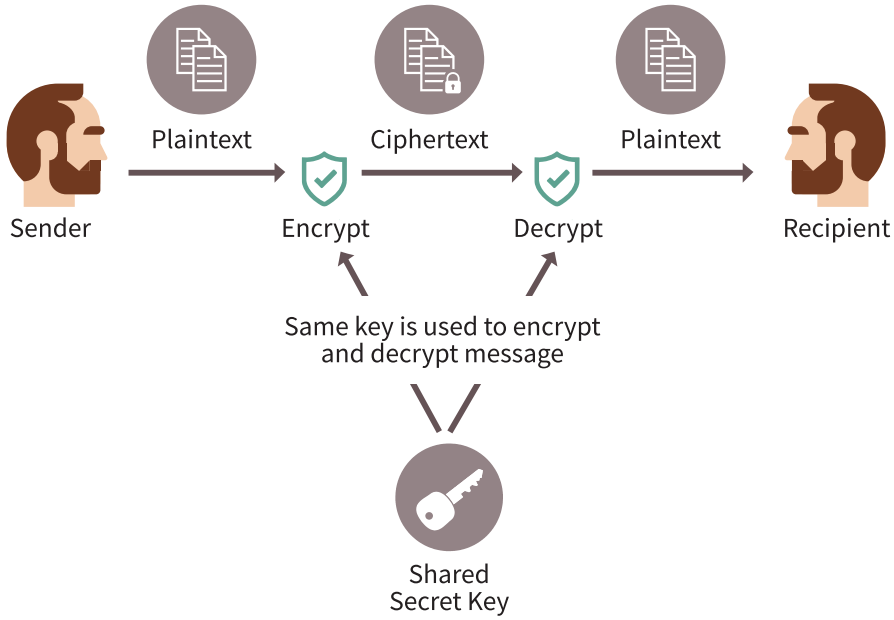
Automotive systems manage vehicle operation and safety, especially with the increasing role of advanced driver assistance systems (ADAS). Thus, it's important that any changes to those systems be handled in a reliable and secure manner. An OTA update is the passing of an OTA object (the new code) from the manufacturer (a server in the cloud) to the vehicle (the client). The client must:

1. Verify the server's identity
2. Check that the OTA object is received intact and unaltered
3. Protect and store it

These steps are achieved via cryptography and two types of algorithms: symmetric and asymmetric.

Symmetric cryptography relies on a secret key that is shared between the server and the client. In order to protect the OTA objects, the key must be kept confidential.

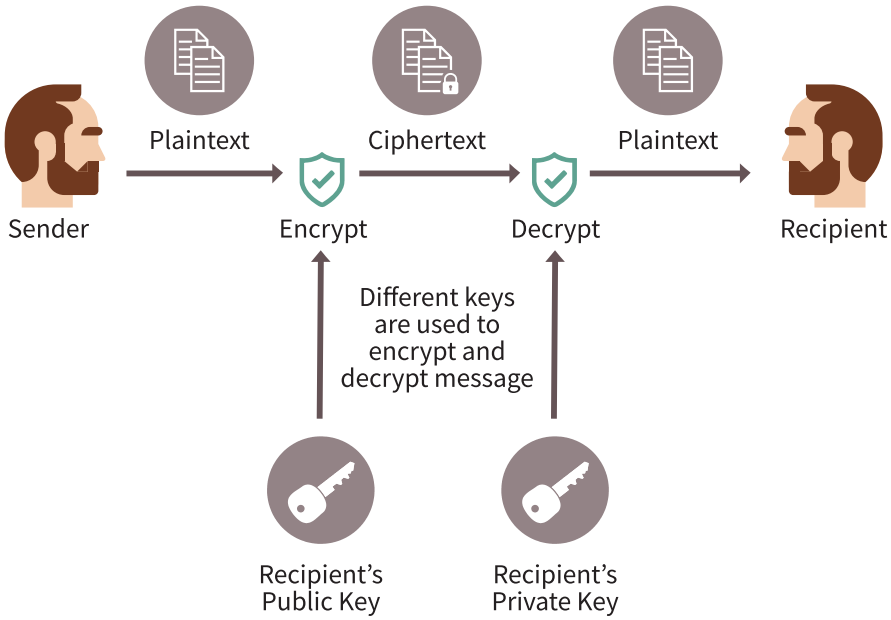
OTA Update with Symmetric Cryptography



This poses a supply chain challenge, as the key must be selected and stored during manufacturing and delivery. To contain the associated risk, different systems may utilize different keys. A master key system manages which keys are used for which systems but can become increasingly complicated when each car has a different set of keys. Because shared secret keys are naturally more prone to security breaches, OTA updates cannot rely solely on symmetric algorithms.

Asymmetric cryptography is a good complement. Asymmetric encryption uses a private/public key pairing. The public key is unconcealed, and therefore doesn't need to be protected. Here it is assumed that the public key is obtained from a trusted source. The key management system maintains a list of private keys, and this list never leaves the manufacturer's secure server. The client and server can establish a secure communication channel by starting with a message that can only be decrypted with the appropriate private key.

OTA Update with Asymmetric Cryptography



When the server successfully decrypts the message, the client is confident it is the right server. This method occurs through internet communications like HTTPS and has proven success in validating the identities of nodes (servers and clients) connected to the network.

The drawback of asymmetric cryptography is that it is far more computing intensive than symmetric. Because of this, the OTA system combines both asymmetric cryptography for higher security and symmetric encryption for faster processing.

True FOTA

Another challenge for automotive OTA updates is deciding when and where the update should be performed. Requiring a visit to the dealer is costly and inconvenient. On the other hand, enough network bandwidth must be available for the size of the update.

True FOTA is a feature with an embedded system that enables updates to be performed regardless of time and location. The update is downloaded quietly in the background, whether or not the vehicle is in operation and network bandwidth is available. The driver is prompted to accept the update when they turn on the ignition, but there is no need to visit a dealer or to park the vehicle.

Requirements for an OTA-enabled Microcontroller

These system-level requirements call for microcontrollers in ECUs to be able to:

- Indicate its identity
- Authenticate its code
- Perform asymmetric encryption, e.g. RSA or ECC, for secure connections and for authenticating updates received
- Store secrets in secure storage
- Store updated code during normal operation
- Transparently switch from old to new code
- Roll back to old code

In addition, the microcontroller should be capable of accommodating different OTA requirements/ applications by OEM/Tier 1 since OTA is primarily driven by automakers whose requirements may vary. Furthermore, the ability to support both symmetric and asymmetric encryption is required.

Though not directly related to the update process, the microcontroller must also support secure boot to ensure that the correct code is loaded during start up. Without a secure boot process, the microcontroller may be operating with unauthorized or corrupt code. To enable field failure analysis, there should be a mechanism to unprotect the device. This could be a common password shared between devices or another method more secure than a simple backdoor.

Traveo™ II Hardware Security Module (HSM)

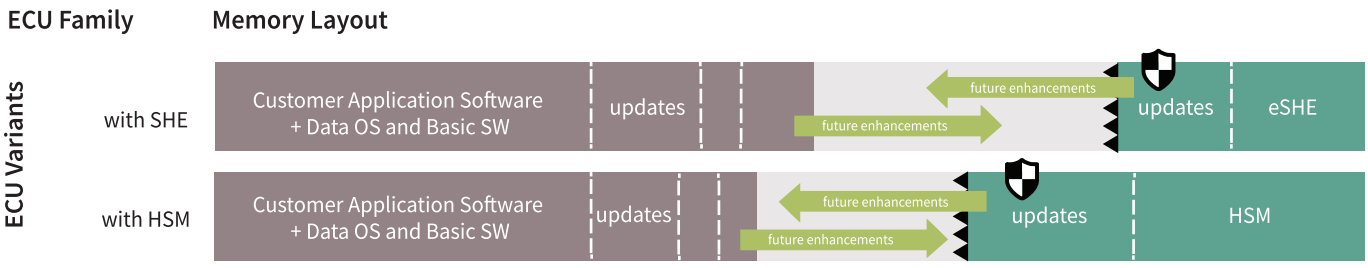
Traveo II meets all these requirements and integrates a configurable HSM with the following features:

- Secure eFuses
- Secure memories
- Secure boot
- Crypto acceleration including asymmetric encryption
- Read-while-write dual bank Flash with bank switching mechanism for True FOTA

Upon reset, Traveo II provides a root-of-trust secure boot to establish a secure HSM domain utilizing advanced hardware protection mechanisms. Code authentication employs a digital signature verification based on asymmetric encryption. Once secure boot is completed, access to secured assets is protected by the hardware and allowed only by the dedicated Arm® Cortex®-M0+ CPU.

Since HSM is established during secure boot, it's possible to define secure domains in numerous ways. Isolating or partitioning memories and other hardware resources can be done in a user-defined portion of the secure boot process. User defined secure boot is authenticated by factory programmed ROM Boot before it starts. Because of this feature, Traveo II HSM can accommodate software to support various requirements.

Traveo II HSM Memory Layout



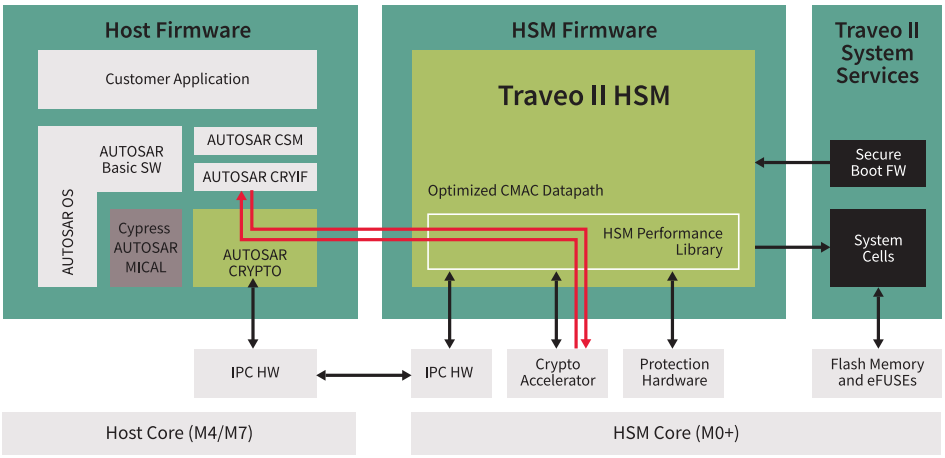
Traveo II hardware protection includes protection context switches. It is structured on a unique context-based access control mechanism, which allows efficient creation of partitions. It also enables fast switching between different active access permission sets without the need for time-consuming reconfiguration of individual protection units.

When failure analysis is required, Traveo II has an RMA (Return Merchandise Authorization) lifecycle stage. Transitioning to the RMA lifecycle stage and opening the device under investigation requires a digital certificate signed by the user who owns the private key corresponding to the programmed public key. The certificate is unique to each device, enabling a very high-level of security.

To fully utilize advanced security features and extract the hardware's best performance, Traveo II HSM is supported by the HSM Performance Library.

Traveo II HSM Cryptographic Performance

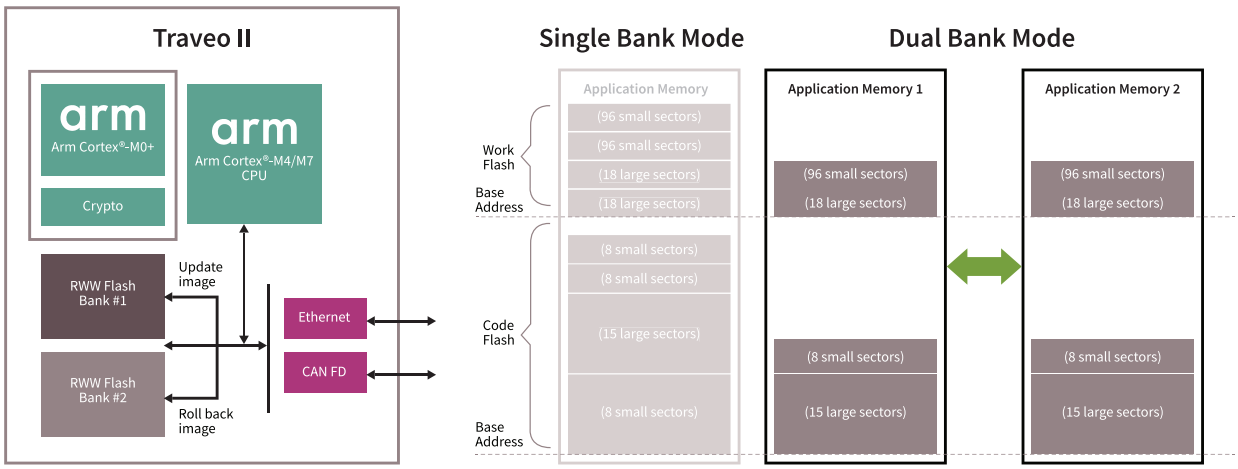
Host Firmware, HSM Firmware, Traveo II System Services



True FOTA Dual-Bank Flash

The onboard Flash in Traveo II supports read-while-write operations and allows switching of memory banks while maintaining the same physical address. Two different codes based on the same physical address can coexist in the Flash memory, so switching between each code can be performed instantaneously.

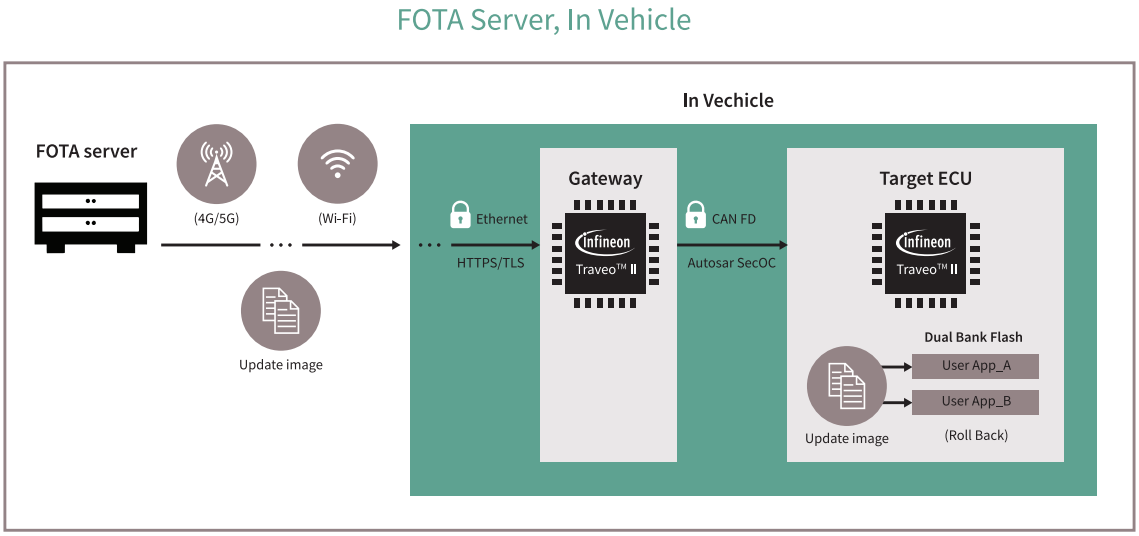
Traveo II Enables True FOTA



In short, the Traveo II MCU is capable of programming updated code during normal operation. After updating the code successfully, the boot firmware can switch from the old to the new code in the next power-on cycle. In the event of unexpected failure with the updated code, rolling back to the old code is also possible.

Conclusion

Traveo II is a True FOTA-capable microcontroller, ideal for connected car systems.



For more information, visit:
https://www.cypress.com/products/cypress-traveo-ii-32-bit-arm-automotive-microcontrollers-mcus

Published by
Infineon Technologies AG
81726 Munich, Germany

Document number:B000-I0000-V1-X-7600-EU-EC
Date: 09/2020