

Safe and Secure Software Updates Over The Air for Electronic Brake Control Systems

Axel Freiwald

Infineon Technologies AG Germany

Gunn Hwang

Infineon Technologies Korea Co., Ltd.

ABSTRACT

Vehicle manufacturers are suffering from increasing expenses for fixing software issues. This fact is mainly driving their desire to use mobile communication channels for doing Software Updates Over The Air (SOTA).

Software updates today are typically done at vehicle service stations by connecting the vehicles' electronic network via the On Board Diagnostic (OBD) interface to a service computer. These operations are done under the control of trained technicians.

SOTA means that the update process must get handled by the driver.

Two critical aspects need to get considered when doing SOTA at Electronic Brake Control (EBC) systems. Both will determine the acceptance of SOTA by legal authorities and by the passengers:

- The safety and security of the vehicle
- The availability of the vehicle for the passengers

The security aspect includes the necessity to protect the vehicle and the manufacturers IP from unwanted attacks.

Existing safety measures ensure safe operation of a vehicle at all times. In general and especially for EBC systems the existing safety measures need extensions for updating software in remote locations. It must be absolutely ensured, that the vehicle will not move while changing the software. In difference to software updates at service stations the SOTA concept must consider any possible level of technical knowhow of any potential driver, who might start an update. This requirement is not just critical for updating the EBC itself but also for updating other ECUs.

Vehicles cannot be driven while updating the software and thus they are not available for a certain period of time. In this paper three different approaches are discussed how to execute SOTA and what this means with respect to the duration of the update process.

The study starts with a detailed analysis of the operations and activities for updating software within a vehicle and EBC unit. It includes descriptions of data flows and examples of the influences by the various bus options of the vehicles' networks.

Next are explained options for scalable system solutions and implementing security functions within the vehicles network with examples of its implementation by using existing electronic components.

Finally we show options for the implementation and how to reduce the downtime of the EBC system.

Several aspects of cyber security in vehicles have been studied in the past. This paper concentrates onto the specifics for EBC systems in a holistic system approach.

CITATION: Freiwald, A. and Hwang, G., "Safe and Secure Software Updates Over The Air for Electronic Brake Control Systems," *SAE Int. J. Passeng. Cars – Electron. Electr. Syst.* 10(1):2017, doi:10.4271/2016-01-1948.

INTRODUCTION: SOFTWARE UPDATES OF ELECTRONIC BRAKE CONTROL UNITS

In recent years OEMs increasingly suffer from software issues and their resulting costs for recalls. With very little additional effort the upcoming emergency call feature and its wireless phone connection can be used to install an internet connection into the vehicles' networks. Thus it is obvious that all OEMs are planning to use this internet connection for updating vehicles' software over the air like it is common use for computers and mobile devices.

In the past the software of electronic brake controllers (EBC) was only updated at the manufactures' sites. For changing a vehicle's software typically the whole ECUs got replaced.

Updating Software on Board

In more modern vehicle generations EBCs can get software updates via the vehicles network using the onboard diagnostic (OBD) interface and a diagnostic system in a service station.

With ISO 14229-1:2013 the service and maintenance operations of vehicles got standardized with the so called Unified Diagnostic Services (UDS). It defines a diagnostics mode and protocol for service and maintenance. Within the diagnostic mode software updates are handled within so called update sessions.

In an UDS update session the Flash memories inside the ECUs can get erased, reprogrammed and verified. Finally when exiting these sessions and modes all ECUs of the whole vehicle are getting a reset and resume normal operation now with the new software versions.

SOTA Updates

The typical approach of OEMs system architects for SOTA is to implement a so called telematics control unit TCU which interfaces to the modem for the mobile communication. The TCU is connected to the gateway of the classical vehicle's onboard network. On vehicles side all software update operations are controlled and executed by the vehicle itself using onboard services and without additional external tools like e.g. a diagnostic system.

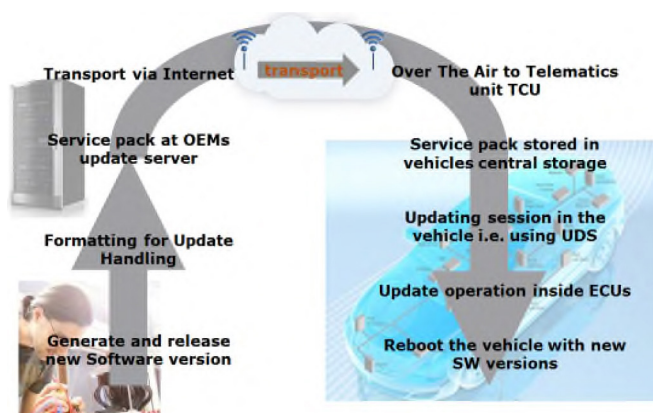


Figure 1. overview of SOTA operations

Figure 1 gives a rough overview of the operations required for updating a vehicle's software. These steps are described in details later in this paper.

Inside the ECUs e.g. EBC

Service and maintenance modes provide functions for updating software of ECUs like electronic braking units.

Within AUTOSAR the so called ECU state manager handles the transitions between normal operation and service modes. In order to enter a service mode all ECUs of a vehicle receive dedicated commands via the network.

In OEMs' first approach for ECUs including EBCs the functions of the service mode can be used regardless whether the update is executed via OBD or over the air (OTA). This approach has the advantage, that the UDS protocol and already existing update functions in the EBCs can get reused for SOTA without changing the EBCs.

New level of Protection Requirements for SOTA

Regarding security the method of doing updates via OBD has a big advantage: The OBD connector can get physically protected by locking a vehicle. This reduces the risk of criminal activities to attack the security and safety of vehicles via OBD without detection. In addition only single vehicles can get attacked at a time.

With a SOTA feature the TCUs wireless data connection does not provide such mechanical protection. The wireless internet connection and the update services of the vehicle are active all the time and thus vulnerable for getting attacked. Another fact makes the situation even worse: Via SOTA it is possible to attack a whole fleet at the same time.

The possibility to attack vehicles was already demonstrated in details in 2015. Not just since then most OEMs and Tier1s were aware of these threats and organized the development of features to make SOTA secure. Secure Hardware Extension (SHE) [11] and EVITA [12] standards are examples of these activities.

ELEMENTS FOR UPDATING A VEHICLE'S SOFTWARE

In figure 2 the building blocks for SOTA inside of vehicles are shown.

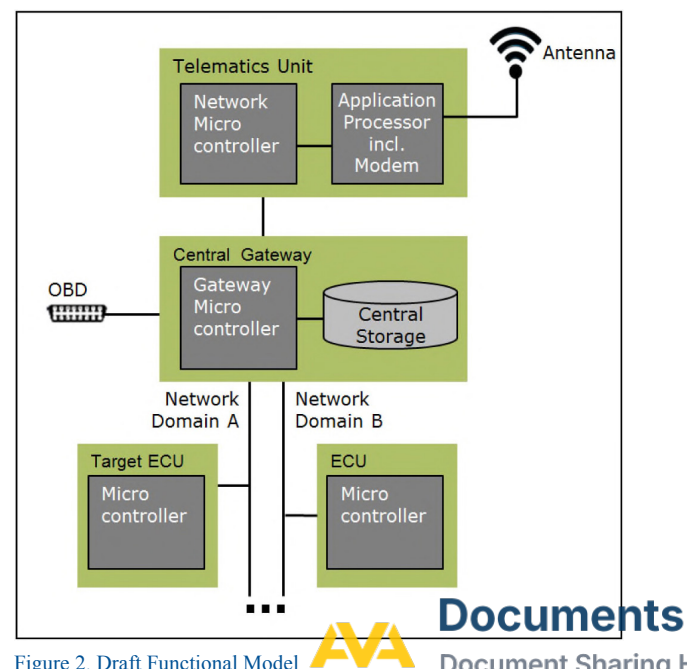


Figure 2. Draft Functional Model



Documents

Document Sharing Hub

Telematics Unit

The telematics unit (TCU) provides the mobile communication functions using mobile phone networks. They provide the emergency call feature as well as an internet connection e.g. using the UMTS protocol.

Central Onboard Update Server Within Gateway

At service stations there is a tool like a diagnostic system which is typically outside of the vehicle and is connected via the OBD interface. The diagnostic tool is connected to the OEMs server for software updates via the internet. For software updates over the air these functions need to get integrated into an onboard ECU within the vehicle's network. In this paper such function is called onboard update server and is for example integrated in the gateway.

The operations and tasks of a SOTA process and their execution in the update server are done in two phases.

1. Download of the service pack from the OEM server OTA and store it in a central storage
2. Update of the target ECUs

In a first phase the onboard update server autonomously handles all communication tasks with the OEMs server. It uses the telematics unit for setting up an internet connection downloads the service pack and stores it in a central storage.

In a second phase of the update process the update server executes and controls the update process itself. It initiates the update process and distributes the software to the ECUs e.g. the EBC.

The onboard update server needs the following hardware:

- Fast data connection to the TCU which can handle the download data rate
- Significant amount of local nonvolatile memory e.g. Flash memory in order to store the downloads: 128MB to 1GB
- Fast connection to the vehicle's network

Typically the update server is located inside the central gateway module though there are other options. There are basically two reasons why to place the update service into the central gateway module:

1. The gateway has a parallel access to most busses of the vehicle's network and thereby it can operate several tasks simultaneously and speed up the process.
2. The gateway is already today a security critical ECU because it performs other security critical tasks like bus separation and firewalling and thus should already provide a high level of security protection hardware e.g. TPM or HSM.

Service Packs and Individual Service Packs

Software updates are typically handled in so called service packs. In the front end of the update process the new software version is being released. In a second step the files of the software are formatted into a

service pack such that they can get handled by the update services. These service packs are stored at the OEMs' update server for distribution to the vehicles.

The update flow shall prepare for the option to update several ECUs in one service operation. This is typically required in cases when only the combination of updates for multiple ECUs were released to drive on the road. Thus a service pack may contain several individual service packs.

For the OEMs' update server there is no difference whether the update is executed at a service station or over the air. For updates in a service station the diagnostic system handles the update operations. In an OTA process the vehicle itself has to act and handle the process. The transport media for both options is the internet.

HARDWARE OPTIONS AND METHODS FOR SECURE CONNECTED VEHICLES

Connected vehicles need secure communication, secure update processes and firewalls in order to protect them from malicious attacks.

The most critical part of protection is to be applied for securing the keys. In vehicles network communication there are typically several keys which are used for different purposes, e.g. for unlocking Flash memories or debug ports, for cyphering messages, for signatures and certificates, for exchanging checksums in form of a secure HASH, or keys for exchanging keys.

The storage for keys is the most critical location from security point of view. Special hardware units with high level protection measures are required in order to protect the keys from external access. These hardware units are typically also responsible for operating all security processes where those keys are being used.

Infineon offers two types of hardware units for protected key storage and security operations, TPM and HSM. These are described later in this paper.

Depending on the key length, keys are more or less secure and the security processes take more or less time.

Asymmetric encryption methods are regarded as more secure than symmetric ones. Symmetric encryption methods are using the same key at both ends of a communication channel. Asymmetric methods are using two keys, a private and a public key. The public key is used to encrypt messages. After that these messages can only get decrypted by the owner of the corresponding private key. For example the OEM's server has the public key of a vehicle and thus can send to this vehicle messages which only can be decrypted by that vehicle. Symmetric methods are typically significantly faster than asymmetric.

Both methods are typically combined to improve the communication throughput without losing the security level. In order to set up a secure communication channel, asymmetric methods are used to exchange symmetric keys which then will be valid for a short period of time and might be changed several times during the conversation.

Inside the vehicles typically symmetric AES128 keys are being used for cyphering. The same method but with frequently changing keys is used to transport service packs from the OEM's server to the vehicle.

Malicious Attacks and Their Prevention

There are mainly two types of attacks to the networks of vehicles, attacking the protection of the vehicle itself and attacking the systems of the OEMs or Tier1s. In this paper the first type is elaborated.

Protection of networks and its security architecture must get planned. With a security assessment the attack scenarios and threads are listed and classified according to their criticality for the system. In a second step the security efforts and hardening measures can get defined.

Attacks to protected systems are often performed by operating those in a non-specified mode or certain functions are manipulated to perform a task which it is not supposed to do. A general concept to mitigate these attacks is to separate the system in various security domains which are isolated from each other.

Example of security domains are:

1. Hardware security domains which isolate the cryptographic processing and key storage from the rest of system
2. Different privilege level in operating systems
3. The isolation between domains within the vehicle by a gateway incorporating firewalling

To create security domains soft- and hardware support is required.

Vehicle Security Architecture for SOTA

Well-defined security architectures make vehicles' electronic networks secure without reducing its performance.

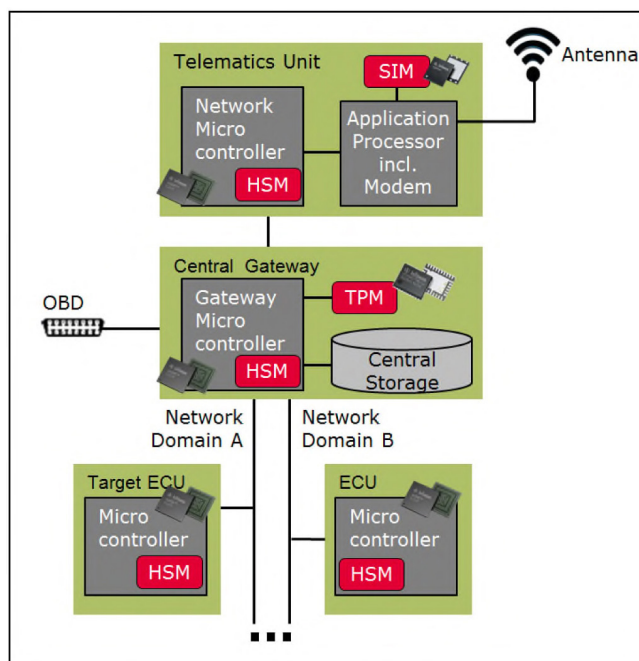


Figure 3. Vehicle security architecture

The following chapter describes the security aspects of the three building blocks which are recommended for a secure architecture for SOTA:

1. The telematics control unit establishes the data link and is executing the service authentication. An automotive microcontroller is typically used to connect the modem and its application processor to the vehicle's network.
2. The Central Update server provides all SOTA services like they used to be executed in the service station by the diagnostic system connected via OBD. In this paper and as an example the central update server is implemented within the central gateway. The service pack is stored in a central storage within the gateway and is verified for the first time. The central storage may contain several versions and act as storage for all software updates. This storage is likely to be implemented in a dedicated flash memory of decent size.
The OBD interface remains connected with the central gateway, and may provide alternate options and functions.
3. The Target ECU e.g. the EBC which will receive the update and will execute the secure update.

The control units are connected by different bus systems with different data rates and communication protocols. The type of bus can have a significant impact on the update duration depending on the implementation model as it will be described later in this paper.

Trusted Platform Module

The Trusted Platform Module (TPM) is a standardized embedded hardware device intended for security purposes. Its technical specification was defined by the Trusted Computing Group (TCG). TPMs were first introduced to enhance the security of portable personal computer like business laptops.

The most important capabilities of a TPM on computer platforms are integrity checks of critical parts of the software, storing private keys used in cryptographic operations, creating true random numbers, storing certificates used in the Public Key Infrastructure (PKI), creating symmetric and asymmetric keys and securely executing cryptographic algorithms where private keys are processed.

The following subsections show how some of these features can be used in the automotive domain.

Vehicle Identification

Identifying vehicles only by their VIN or by their IP address is not secure and can easily be attacked. This is a fact which is already well known from computers.

Secure identification methods are also called authentication and are using asymmetric cryptography with a public and private key. A TPM can securely store private keys and use them to securely sign random numbers used in authentication protocols.