Research on automotive ECU remote update and it’s security

Ji Zhang, Yu Lv\textsuperscript{1} and Zhi Liao

Tongji University

\textsuperscript{1} E-mail: 1733443@tongji.edu.cn

Abstract. With the rapid development of automotive electronic technology, more ECUs with much complicated control programs are used in automobile, which could cause serious problems in security and sensitivity. To solve this problem, on-board ECU software program needs to be upgraded. Traditional upgrading technology is so complicated that only professionals can make it, which consumes long period and high cost. Thus, the remote wireless upgrade technology is of great importance in simplifying update process and improving efficiency. This paper aims to develop a completed remote upgrading system for an on-board ECU, involving the overall scheme of remote update, security technology and functional design. Firstly, a general plan is come up including the functional framework and telecommunication mechanism between remote server, on-board platform and ECU. Secondly, aiming to avoid potential safety hazards in remote update, optimal encryption algorithms are added to increase security mechanisms. Thirdly, the on-board hardware platform with Android operating system, remote server and client application are built to realize update-package upload and download as well as other functions through 3G/WiFi. ECU is connected to the platform through distributed CAN bus. Moreover, a bootloader program is developed to reprogram the ECU, which is verified in a test combined with the on-board platform.

1. Introduction

On March 20, 2015, Shanghai General Motors co., LTD. announced the recall of some imported Chevrolet Wo blue car due to body control module software problem. At that time, the processing mode is free to update the module software.

Traditional vehicle ECU update process is very complex, which relies largely on update-tools and professional engineers. Besides, vehicles need to be recalled to realize direct refresh. It caused a huge cost burden to OEMs, required the user time and affected the consumer's experience as well. Nowadays, as shown in figure 1, the internal network of vehicles can be connected with external internet, which means that cars can communicate with remote server and even any other networked devices. The ECU update technology is accomplished through information interaction between remote server and vehicle terminal equipment. Therefore, the development of networking technology makes the ECU remote update technology have the application environment and technology foundation. Remote update of on-board ECU can greatly simplify the ECU firmware upgrade process, avoid vehicle recalls caused by ECU complicated control software, reduce the cost of OEMs, and eliminate many troubles in maintenance.

According to the functional requirements analysis of the vehicle remote update system, the solution as shown in figure 2 is put forward, including vehicle hardware platform establishment, remote server development, client function development and vehicle ECU bootloader design.
Literature [1] put forward that telecommunicate protocol with Hash chain table and symmetric key encryption technology costs less memory, which is suitable to vehicle telecommunication environment. In literature [2], firmware would conduct self-verification after update to avoid potential attack during update code writing to ECU. To intercept software updates that may be in the wrong behaviour, codes in update package would be checked and revised according to security strategy in literature [3]. Based on these studies, aiming at potential safety problems in the system, the security mechanism in this paper integrates encryption, authentication, identification and access to guarantee safety.

In this paper, Chapter 2 introduces overall scheme of vehicle ECU remote update including system functions. Chapter 3 comes up with new security mechanism aiming at potential problems in remote update. The security is strengthened from three components (developer, server and vehicle) and four aspects (encryption, authentication, identification and access). Chapter 4 builds remote update system covering on-board platform and server. Functions are developed in this system. Chapter 5 verifies the feasibility of total remote update system.

2. Overall scheme of vehicle ECU remote update system

This paper intends to realize remote update of automotive firmware. Updating package is sent by remote server and received by on-board software and hardware. In this process, six functions as ECU software updating-package upload, software version information management, updating-package release and download, wireless network connection, safety management and ECU software refresh should be accomplished.

Specifically, telecommunicate mechanism is shown in figure 3, which can be summarized in 7 simple steps. (1) OEMs or suppliers upload ECU firmware updates to the server, the vehicle platform client checks if ECU software update package is available when the vehicle starts; (2) If the available ECU software is a new package, vehicle platform notifies the user through a pop-up message that system displays; (3) According to the pop-up message box, users select update operation, such as immediate or delayed updating; (4) If the user confirms update operation, the vehicle platform sends request to server to download update package and save to the local storage device, or wait for the next round of notice; (5) By receiving download request, the server sends software update packets to the car by 4G/Wifi wireless technology. After finishing downloading, the server sends successfully downloaded information to vehicle platform; (6) Vehicle platform client installs ECU software update package on ECU through vehicle CAN network; (7) After the automatic update, vehicle platform sends vehicle code and updating success information to server.
3. Research on the security of vehicle ECU remote update

During the 2015 DEFCON hacker conference, Jeep, Tesla Model S and other vehicles was attacked, which is a signal that the car network security issues is becoming increasingly important.

In general, vehicle network should have several security functions: identity authentication, abnormal behavior detection, data integrity, data confidentiality. However, in remote update system, there are potential problems such as reverse development of S19 files, tamper with S19 files, upload untrusted software and access ECU illegally, which could not be effectively solved by general network functions. Thus, this paper would improve the overall scheme of remote update and add corresponding security mechanism to weak links.

When it comes to the reverse development problem of S19 file, the file can be encrypted, namely OEMs or suppliers firstly encrypt S19 file and then upload it to the remote server. On receiving encrypted update, the vehicle platform decrypt to get the S19 files. So even the file is acquired by non-authorized agencies, they cannot obtain the content; For the problem that the file is tampered with, data integrity verification is added, so that the vehicle platform can accurately determine whether the file is modified, and then determine whether the next step should be in operation or not; By adding the
identity authentication mechanism, based on the software package certificate information, the vehicle platform can verify the source of the package and effectively prevent the non-trusted software update package from entering the vehicle platform. After adding the ECU security access mechanism, the terminal matched with the ECU encryption mechanism can enter the ECU to refresh it. The concrete process is shown in figure 4.

3.1. Remote update packet encryption processing

Update packet encryption link uses AES encryption algorithm, which is a symmetric encryption algorithm [4]. AES can be used for encryption and decryption of a 128 bits data grouping. The advantage of symmetric encryption method is that it can lead to less computation and high calculation efficiency by using simple logic operations such as and, or, XOR for encryption and decryption. For remote update packet encryption with large data amount and the long data stream, it is more suitable to use the symmetric encryption algorithm for efficiency. The process of remote update packet encryption is as follows. Firstly, S19 file needs to be divided into N groups. Each group is 128-bits, namely 16-bytes. However, sometimes, the S19 file cannot just be divided into each group of 128-bits. In such cases, before grouping, it needs to be filled after the data that S19 file has been parsed out, so that it can meet the needs of the grouping of 128 bits. As shown in figure 7, on the base of N data packet plaintext, the AES encryption algorithm and 128-bits key are used to obtain the n groups of cipher text, namely encrypted file of S19. In this paper, 128-bits key encryption algorithm is employed for S19 file encryption and decryption in 10 rounds. The decryption process of the symmetric encryption algorithm is actually the inversion of encryption process.

3.2. Remote update package authentication and data integrity verification

The AES algorithm is a grouping encryption algorithm, which means that the plaintext for each data group is separately encrypted. In the transmission process, if a data set has been destroyed or tampered with, it is difficult to detect. Therefore, the vehicle platform also needs to update the package to verify data integrity.

**Figure 5. Authentication and integrity verification.**

The authentication to update package is mainly to ensure the reliability of its sources and to prevent non-trusted institutions sending update package to the vehicle platform. Data integrity verification can verify whether the data has been tampered with or not and to ensure that the file downloaded to the vehicle platform is an effective ECU update package. Figure 5 shows the authentication and integrity verification process of the update package based on RSA digital signature algorithm [5].

Server side: Firstly, update package is hashed to generate updated file's information abstract. Then the abstract and server's secret key are used to generate signature by the RSA signature generation algorithm. At last, the signature and update package are sent to vehicle platform.

Vehicle platform side: After receiving the update packet and signature from the server, the server's public key is applied to verify through RSA algorithm. The valid signature means that update packet is
the private key of server, which is a reliable source. Then its integrity is further tested. If it is undamaged, the file would be saved.

3.3. ECU security access mechanism
According to the provisions of ECU refresh section in road traffic control LAN diagnostic protocol (ISO 15765), secure access is listed as vehicle ECU refresh recommended step [6]. The provisions of vehicle general fault diagnosis protocol ISO 14299 (ISO 14299) about the security access service gives the basic process of ECU refresh out. The security access mechanism in this paper is based on ISO 14299 as shown in figure 6.

![Figure 6. Security verification process.](image)

The following steps are included: (1) The vehicle platform computer requests access to ECU; (2) Vehicle ECU terminal generates a random seed, and send it to vehicle platform. At the same time, the ECU uses key and seed as input to calculate the cipher text by the agreed encryption algorithm; (3) The vehicle platform computer receives the seeds. Then the seeds are encrypted through the same algorithm and key to get cipher text. Next the cipher text would be sent to vehicle ECU; (4) ECU compares the cipher text from the vehicle platform and the step-2-generated cipher text. The consistent two means that the requesting device has the agreed key and knows key encryption algorithm, ECU terminal is legal, so ECU will be unlocked. Otherwise ECU will deny access; (5) The result is sent to the vehicle platform.

4. Vehicle platform function building and application development

4.1. Vehicle platform function building
Android operating system is chosen to use in vehicle platform considering its open source, abundant application and low development costs. Under the condition, main processor chip for development board is selected as i.MX 6Quad, which has four cores. Its operating frequency is up to 1.2 GHz. Devices in this series integrate FlexCAN, MLB bus, PCI Express, MIPI display port, MIPI camera port interface and so on. In other words, it is an application platform for consumer electronics, automotive and industrial advanced multimedia.

The concrete process of android porting [7] is as shown in figure 7. Firstly, development environment should be set up. Android source code should be obtained. Considering that the amount of Android source code is very large, so it needs to be divided into two parts to obtain respectively, Android and Android kernel source code (Android Linux Kernel). Then the Linux kernel and Android file system should be compiled separately. Finally the image file generated by the compiler is downloaded to the hardware platform. The function of the application on Android platform is to download the ECU update package of the remote server to the vehicle platform, and then vehicle platform downloads the ECU update file to the vehicle ECU through the CAN bus.
4.2. Application development

4.2.1. Application development in server side. The application framework of server is shown in figure 8.

![Server side application framework](image)

Figure 8. Server side application framework.

Functions of remote update server include user login, remote update package download, ECU update check, security check and ECU refreshing. Remote update system’s ECU update package upload function is mainly for users of OEMs or suppliers rather than owners of cars. When the car has ECU failure or needs to upgrade, ECU application developers should upload update package to the remote server, so that the vehicle platform can download the update package to upgrade ECU software program. ECU update check function is for the car platform to check whether there is a new package on the server to update or not. Update check is achieved mainly through the relevant information comparison between the remote update service and ECU update packet.

4.2.2. Application development on-board. Applications are developed aiming at the five functions of remote update.

1. Vehicle platform user login
   After obtaining the password information of the input account, the vehicle platform client connects to the server login to check the port, and sends the account password information to the server. The core code is as follows:
   ```java
   httpsRequest.execute(Common.ServerIp+"/session/logincheck.php","username",username.getText().toString(),"password",password.getText().toString());
   ```

2. ECU update verification
Under the condition of update verification, client side is linked to “update_check” of remote server client API interface module to update verification interface through codes:

   mHttpsAsyncRequest.execute(Common.ServerIp + "/client_api/update_check.php")

(3) Update package download

Server and automotive platform client communication stage begins in the car platform client applies to the server to download the application, end in car ECU package download firmware to the vehicle platform SD card within the specified folder

(4) ECU refresh

The refresh process includes four basic steps: update package download; update package parsing; Flash block erasure; writing Flash block. Update package download means receiving update S19 files data from on-board platform through CAN bus. S19 files are needed to be parsed to write on Flash. However, Flash with original code data should be erased first and then be written.

4.2.3. Vehicle ECU terminal Bootloader development. In addition to boot program in the boot phase, bootloader has the function of loading application. In this remote update system, bootloader program on the ECU is mainly responsible for downloading update package from vehicle platform to the corresponding ECU memory, in order to achieve ECU software online upgrade.

S19 documents have mainly three record types of S0, S1 and S9. S0 represents the beginning of the program while S9 represents the end of the program. The two types do not need to be burned to Flash. Address field in S1 record is represented with 2 bytes, data field consists of data that can be loaded into memory, which is the main content of the S19 file. All the codes of update program are recorded by S1 record. Thus, the essence of ECU update is to burn the data field of the S1 record to the Flash storage unit, which is specified by the address field. Take the P-Flash write and erase function as an example: a structure array is defined to save contents of each record in file.

typedef struct SRecord {
    uint8 Type; // number of the record
    uint8 DataLength; // data length
    uint32 LoadAddr; // initial address value
    uint8 Data[64]; // data field array
    uint8 CheckSum // data checksum
} *SRecData;

According to the above steps, the P-Flash sector erase function, namely PFlash_EraseSector (address UINT32) and the P-Flash write function, namely PFlash_Program (address UINT32, *ptr UINT16) are designed to implement the erase and write operation of Flash. Using these two functions with the data parsed from the S19 file, through the process shown in figure 9, traversing the data structure through the update package, and extracting the corresponding addresses code and data code. Firstly, use the PFlash_EraseSector function to erase the P-Flash sector corresponded with the address information, and then write the corresponding code data to the sector through the PFlash_Program, read the next record check after inspection and correct until the end of S9 record, then finish the write operation of update files.
5. Experimental verification
The basic working process of ECU vehicle remote update system developed in this paper is: firstly, developers need to upgrade the ECU update encrypted files uploaded to the remote server; vehicle platform downloads the update package from the remote server, and decrypt it to get the S19 file; the vehicle platform client through the ECU refresh function will refresh the S19 file to ECU; the function also need to use the ECU Bootloader to receive the update file from vehicle platform, and write it into the ECU memory, complete remote update operation.

As the design part of this paper, the three parts included in the remote update system, have been separately marked in the figure 10. The remote service is built on the Linux LAMP server architecture; vehicle platform is an Android system development platform based on imx6, connects to the remote server through WIFI/3G, using the CAN bus communicates with automotive ECU; automotive ECU uses a Freescale microcontroller development board MC9S12XS128 to carry on the simulation.

Three steps are taken to test the function of vehicle terminal, the first is to download the firmware, then check the local firmware package, and finally to implement the update firmware operation and observe whether the automotive ECU is successfully refreshed.

After the implementation of the corresponding operation, observing the sequence of LED lights on vehicle ECU. It shows that lights program has been started running, which means the entire remote update operation has successfully completed.
6. Conclusions
With the rapid development of automobile electronic technology, the complexity and the number of vehicle control program in ECU is increased, the car's warranty and recall issues caused by the software code are correspondingly becoming more and more frequent, the traditional way to update the vehicle ECU is difficult to meet the development requirements of automobile electronic technology. Therefore, this paper proposes an implementation of automotive ECU remote wireless update technology program, which would download update files from the vehicle information platform to the remote server, and brush it to the car ECU. In this paper, we have studied the security problems that may exist in the remote update system, and put forward the corresponding solutions. Based on this scheme, this paper has carried on the remote server configuration, the vehicle platform construction and the bootloader design of the vehicle ECU, and completed the application design of the whole ECU remote update process. At last, the feasibility of the system is tested. In general, in the future when vehicles need to be recalled due to software code problems, the remote update system designed in this paper can be referred, the corresponding renewal mechanism can be established, in order to simplify the ECU update process, ensure the safety of the ECU update, and save the time cost of the owner and host manufacturers.

References
[1] Sun C X 2012 Design and analysis of digital signature algorithm based on attribute[D] Xi’an: Xi’an university of electronic science and technology
[2] Idrees M S, Schwegge H, Roudier Y, et al. 2011 Secure automotive on-board protocols: A case of over-the-air firmware updates[M] Communication Technologies for Vehicles 224-238
[3] Phung P H and Nilsson D K 2010 A model for safe and secure execution of downloaded vehicle applications[C] Road Transport Information and Control Conference and the ITS United Kingdom Members’ Conference (RTIC 2010) - Better transport through technology 1-6
[4] Wu W F 2015 Research and realization of the security technology of electricity bill based on U shield technology[D] Beijing: North China electric power university
[5] Liu H 2013 Research on the self-organizing network information authentication and privacy protection mechanism[D] Xi’an: Xi’an university of electronic science and technology
[6] ISO15765-3 2004 Road vehicles - diagnostics on controller area networks (CAN) - Part3: Implementation of unified diagnostic services (UDS on CAN). International Organization for Standardization
[7] Ye B F 2010 Research on Android operating system transplantation and key technologies[D] Guangzhou: Ji’nan University