Computer Security Symposium 2016
11 - 13 October 2016

セキュアなリモートリプログラミング方式の実装

溝口 誠一郎[†1]　竹森 敬佑　川端 秀明　窪田 歩

概要：コネクティッドカー時代において，車載 ECU (Electric Control Unit) のセキュアなファームウェア更新は重要な課題である．本稿では，車載ネットワークのアーキテクチャについて整理を行った後，リモートリプログラミングの実装モデルについて提案を行った．また，UDS (Unified Diagnostic Services) に準拠したセキュアなリプログラミング手法について，その実装方法を検討した．

キーワード：自動車セキュリティ，組み込み，リプログラミング, UDS

Implementation of Secure Remote Re-programming Schemes

Seiichiro Mizoguchi[†1]　Keisuke Takemori　Hideaki Kawabata　Ayumu Kubota

Abstract: In the connected car era, a secure firmware update technology against in-vehicle ECUs is really needed. In this paper, we analyze the in-vehicle network architecture and we propose several secure re-programming schemes. We also consider implementation with UDS (unified diagnostic services) .

Keywords: In-vehicle security, embedded systems, re-programming, UDS

1. はじめに

コネクティッドカー時代に向かって，車載 ECU の機能はより多様化･複雑化することが予想される. それに伴い，これらの機能の追加や改修も，これまで以上の頻度で実施されると考えられる．現在は，車の所有者が自ら販売店に足を運び，作業員が専用のツールを接続することでメンテナンスを実施することが主流となっているが，この頻度が増加すると，所有者ならびに作業員の両者にとっても負担が大きい. そこで，車載 ECU のファームウェアをリモートでアップデートする技術の確立が求められている.

しかしながら，セキュリティが考慮されていない設計により，これらのアップデート機能を悪用される事態も起きている．チャーリー・ミラーとクリス・バラセクは，Jeep Cherokee に搭載されたインフォ端末の脆弱性を突き，インフォ端末から車載制御ネットワークに対して任意のパケットを送信できるように改造した上で，リモートから運転を制御するデモを成功させている[1].

一方，車載マイコンやそれらに搭載するソフトウェアのセキュリティについては，EVITA Project や AUTOSAR など，セキュリティ機能の標準搭載に向けた取り組みがなされている．セキュアなリモートアップデートでは，これらの機能と連携しつつ，シームレスな移行のために，既存のアーキテクチャやリプログラミングの方式との親和性も維持しなければならない．そこで本稿では，車載ネットワークのアーキテクチャについて整理を行い，リモートリプログラミングの実施モデルを検討した．検討したモデルのうち，セントラルゲートウェイからセキュリティ機能を切り出してゲートウェイの内側に配置したモデルにおいて，セキュアなリモートリプログラミングの実施方法を提案する. さらに，既存のリプログラミング手法である UDS ベースのリプロ方式に準拠した，セキュアなリモートリプログラミング方式を検討する.

2. 車載ネットワークのアーキテクチャ

車載 ECU のリモートアップデートを検討するにあたり，車載制御ネットワークのアーキテクチャについて整理する.

2.1 ゲートウェイによるドメイン分離

まず，これまでのバス型のネットワークを，セントラルゲートウェイによってインフォ端末系と制御系のネットワークを分割するアーキテクチャである ()．セントラルゲートウェイを配置し, CAN パケットのフィルタリングを実施することによって，インフォ端末系ネットワークや診断ポートから，意図しない CAN パケットが制御系ネットワークへ送信されることを防ぐことが期待されている.

柳川らは，車載ネットワークを，制御系ネットワーク・OEM 情報系ネットワーク・OPEN 情報系ネットワークの 3 つのドメインに分割し，情報系と OEM 情報系を Control Gateway で，OEM 情報系と OPEN 情報系を Information Gateway で相互接続するモデルを提唱している[2]．早川らも，車両外部から車外ゲートウェイを介して情報系ネットワークに接続し，さらに情報系ネットワークと制御系ネットワークを車内ゲートウェイによって分離する多層防御のセキュリティコンセプトを示している[3].

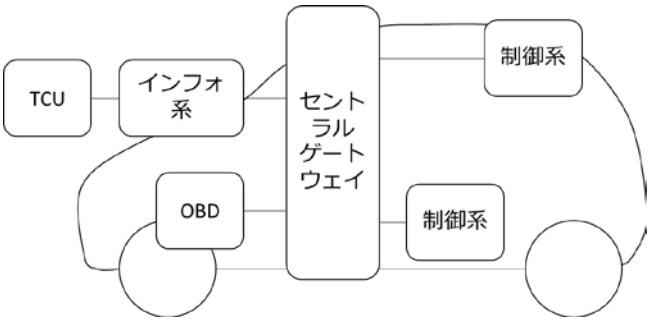

図 1　セントラルゲートウェイによるドメイン分離

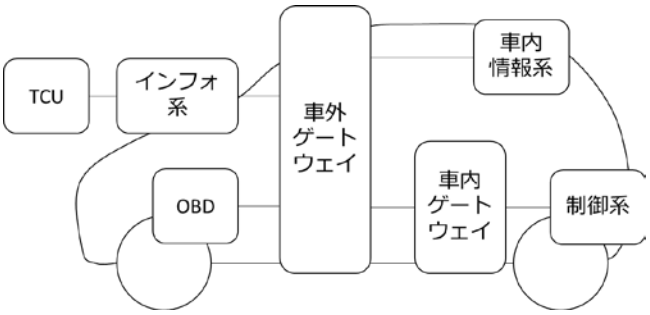

図 2　多層防御のアーキテクチャ

このように，今後の車載ネットワークは，ゲートウェイによるドメイン分離を行い，役割毎に通信範囲を閉じる設計へと変化しつつある．

2.2 エンド ECU

エンド ECU に搭載されるマイコンとしては，セキュリティ機能の搭載されていないこれまでのマイコンに加え，EVITA や SHE の仕様に準拠したマイコンや，Automotive Grade TPM など，セキュリティ機能が搭載されたものが，今後数年間は混在するものと思われる．

3. リモートアップデート

ここからは，リモートアップデートの手法について検討していく．

3.1 既存手法

Klimke らは，セキュアかつシームレスな導入を目指した Over-the-Air のソフトウェアアップデート手法を検討し，システムの要件等を定義している[4]．彼らは，セントラルゲートウェイがストレージを持ち，ファームウェアの検証を実施するモデルを提案している．Idrees らも同じく，Over-the-Air のファームウェアアップデートのセキュリティ要件をまとめており，車載診断ツールに HSM を搭載したマイコンを採用し，エンド ECU と HSM との間で認証を実施することによってセキュリティを担保する仕組みを提案している[5].

3.2 リモートリプログラミングのセキュリティ要件

3.2.1 サーバ～車両間認証

車両からファームウェア配信サーバにアクセスする場合，そのサーバが正規のサーバかを検証する必要がある．一般的には，PKI の仕組みを利用し，サーバの公開鍵証明書を車載機側に配布する．このとき，偽の公開鍵証明書がインストールされないよう，車載機のストレージの設計を考慮しなければならない．

また，車両からサーバに対して情報をアップロードする際は，クライアント証明書も必要となる．ファームウェアの取得を試みる，あるいは不正な端末がサーバにアクセスすることを防ぐために必要である．

3.2.2 ファームウェア暗号化

ファームウェアの暗号化は，リバースエンジニアリングを防ぐために実施する．セキュリティ対策を施していたとしても，ファームウェアに脆弱性が含まれている場合，リバースエンジニアリングによって脆弱性が露呈すると，様々な攻撃の糸口となってしまう．

3.2.3 ファームウェア署名検証

リモートリプログラミングにおいて最も重要なのは，不正なプログラムが書き込まれてしまうことを防ぐことである．チャーリー・ミラーのデモにおいても，インフォ端末上の CAN コントローラが搭載された CPU に不正なプログラムをリプロできたことが，遠隔制御を可能とした最大の原因である．ファームウェアに署名を施し，エンド ECU 側で検証することで，不正リプロを回避する．こちらも，一般的には PKI を利用した公開鍵暗号による署名が考えられるが，エンド ECU では公開鍵暗号が扱えない場合があり，この場合は CMAC 等の共通鍵ベースの手法を用いることとなる．

3.2.4 書込みチェックとセキュアブート

リモートリプログラミングを実施した際に，それが正しく完了したかどうかは，書き込み後のチェック及び，セキュアブートによって確認する必要がある．その場合，サーバに検証結果を通知する必要がある．

4. セキュアなリモートリプログラミング

本章では，セキュアなリモートリプログラミングの実現方法について整理する．IVI にリプロ機能を持たせるモデル，ゲートウェイにリプロ機能を持たせるモデル，ゲート

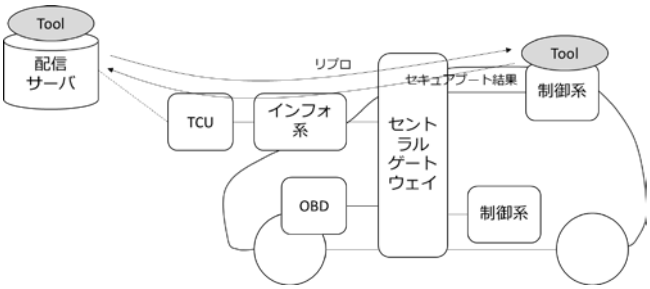

図 5　サーバ～エンド ECU 直接モデル

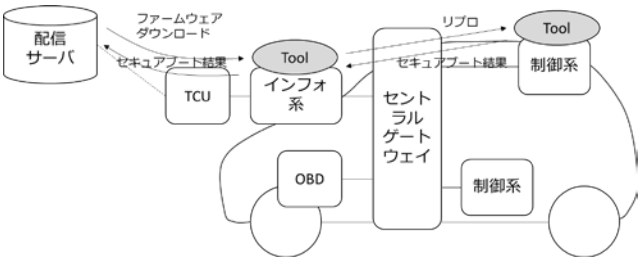

図 6　インフォ端末搭載モデル

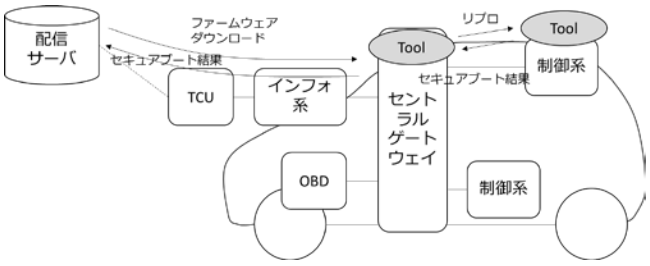

図 4　セントラルゲートウェイ搭載モデル

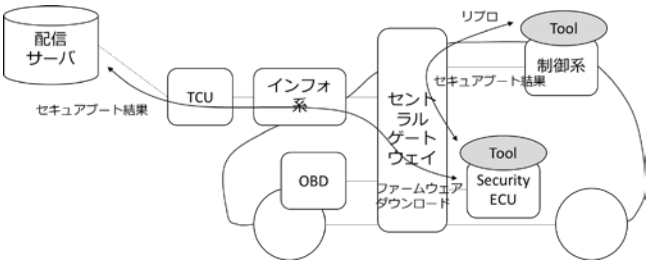

図 3　リモートリプロ用 ECU 搭載モデル

ウェイの内側にリプロ用の機能を持ったモデル，サーバとエンド ECU で直接実施するモデルがある．また，各モデルにおいて，共通鍵ベースや公開鍵ベースなどの実装の違いについても触れる．

4.1 サーバ～エンド ECU 直接モデル

サーバ～エンド ECU 直接モデルでは, Telecommunication Unit を含むインフォ系端末やセントラルゲートウェイはトラフィックの転送だけを行い，サーバとエンド ECU がセキュリティ機能を持つモデルである．

4.2 インフォ端末搭載モデル

インフォ端末搭載モデルは，リプログラミングの機能をインフォ端末に持たせるモデルである. ファームウェアは、インフォ端末に一旦バッファされ, そこからエンド ECU に対してリプログラミングが実施される．

インフォ端末は，比較的 CPU パワーがあり，ストレージも豊富にあるため，ファームウェアのバッファリングや計算コストの大きい暗号処理等を受け持つことができる．

4.3 セントラルゲートウェイ搭載モデル

セントラルゲートウェイに搭載するモデルである．セントラルゲートウェイには，EVITA Medium 以上のセキュリティチップが搭載される可能性があり，これらの機能を用いたリモートアップデート機能を搭載する方式である．

4.4 リモートリプロ用 ECU 搭載モデル

セントラルゲートウェイの内側に，これまでの制御系 ECU と並列して，セキュリティ機能を搭載した ECU を配置するモデルである．セントラルゲートウェイは，診断ポートやインフォ端末系とのインターフェースを持つ上，多くの CAN トラフィックが集中するユニットとなるため，セキュリティと性能面から，リモートリプログラミングや鍵管理の機能を搭載することがはばかられる場合がある．その場合，これらの機能をセントラルゲートウェイから切り出し，ゲートウェイのフィルタリング機能の内側に配置することもできる．

4.5 提案方式

本節では，共通鍵ベースの手法，および公開鍵ベースの手法について，リモートリプロ用 ECU 搭載モデルで設計する．

4.5.1 公開鍵ベースの方式

まず，サーバにおいてファームウェアに署名を施す．サーバの秘密鍵で署名を施すか，あるいはファームウェアのベンダーが事前に署名を施す．

続いて，サーバとセキュリティゲートウェイの間で暗号化通信路を構築するため，サーバ・クライアント証明書の検証をしたうえで，サーバとセキュリティゲートウェイの間でセッションキーを共有し，暗号化経路を構築する．
続いて，前述の通信路を介して署名付きファームウェアをセキュリティゲートウェイに送信し, バッファリングする．セキュリティゲートウェイで，サーバの公開鍵証明書もしくは，ファームウェアベンダーの証明書を用いて検証を実施する．

ファームウェアをダウンロード後は，エンド ECU に対して，車内共通鍵でファームウェアを暗号化しつつ送付す

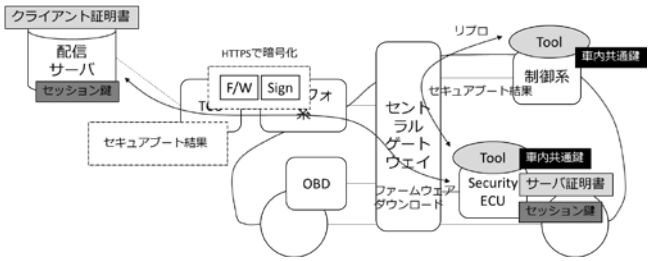

図 7　公開鍵ベースの手法

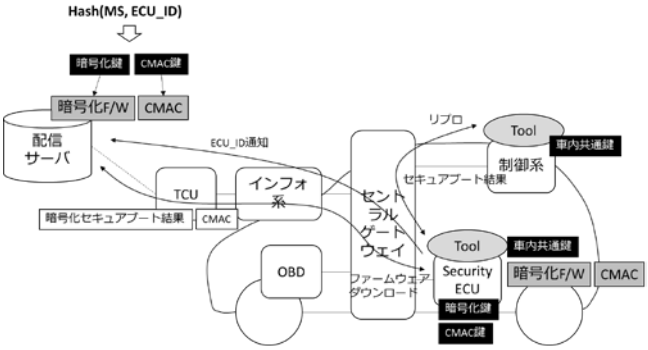

図 8　共通鍵ベースの手法

る．これは，CAN をタップされることによってファームウェアのバイナリが漏えいすることを防ぐためである．あるいは，セントラルゲートウェイの内側はセキュリティが担保されているという前提のもと，セキュリティゲートウェイからエンド ECU へ平文でファームウェアを配信することも考えられる．

最後に，リプロ実施後のセキュアブートにおいて，セキュアブート結果をサーバにアップロードする．このとき，セキュアブートの結果を取得するインターフェースを介して，セキュリティゲートウェイが結果を受け取り，個車毎の秘密鍵を用いて署名を施した上でサーバに送付する．

4.5.2 共通鍵ベースの方式

ファームウェアをセキュリティゲートウェイの共通鍵で暗号化することによって，HTTPS などの暗号化経路を不要とする．署名については，サーバとセキュリティゲートウェイとの間で共有した鍵を用いて CMAC を計算しファームウェアに付与する．これらの暗号化鍵・CMAC 鍵は，Master Secret とセキュリティゲートウェイの ECU ID から導出される鍵を用いる．ゲートウェイ側は，組み立て時に鍵をセットしておき，サーバ側では Master Secret のみを管理することによって，すべての ECU 用の鍵を管理する必要を無くしている．

セキュリティゲートウェイがファームウェアをダウンロード後，暗号化鍵を用いてファームウェアを復号する．また，CMAC の検証を実施する．検証が成功したのち，ファームウェアをエンド ECU に対して送付する．このとき，車内共通鍵でファームウェアを暗号化しつつ送付するのが望ましい．エンド ECU では，ブロック毎に復号しつつ書き込みを行う．

エンド ECU では，セキュアブート後，セキュアブートの結果をメモリ上に保存しておく．そして，セキュリティゲートウェイが適当なタイミングで保存されたセキュアブートの結果を取得し，セキュリティゲートウェイの鍵で暗号化してサーバに送付する．

4.6 考察

公開鍵ベースの手法では，サーバの公開鍵証明書をインストールしておくことで，サーバ～車載機間の暗号通信路を構築することができるため，共通の鍵を持つ必要がない．一方，セキュアブートの結果を送信する際は，個車を区別するために，個車毎の秘密鍵と公開鍵証明書を準備する必要がある．

共通鍵ベースの場合，ECU_ID で導出される鍵によって暗号化と CMAC 計算が行われるため，サーバ側の鍵管理におけるストレージの問題が解決される．

5. 実装：UDS ベースのリプログラミング

我々は，セキュリティゲートウェイからエンド ECU に対してリプログラミングを実施する際に，Unified Diagnosis Services (UDS) に基づいた手法を検討した．

5.1 Unified Diagnosis Services (UDS)

UDS は，ISO-14229 に規定されている，一般的な診断システム構築のための規格である．2013 年の第 2 版では，第 15 章に Non-volatile server memory programming process の項目が設けられ，リプログラミングにおいてやり取りすべきパラメータ等が規定されている．

5.2 クライアント・サーバ

UDS では，リプログラミングを受ける側がサーバ，リプロを指示する側をクライアントと規定している．本稿では，エンド ECU がサーバ，リプロを実施するツールがクライアントとなる．

5.3 ファームウェアの暗号化と復号

4.5 節の方式にてエンド ECU にファームウェアを送付する場合，暗号化の処理が行われる．この処理は，メモリに書き込む単位で行われることが望ましい．UDS では，Programming フェーズ内において，Request Download → Transfer Data → Request Transfer Exit という流れでデータ送信と書き込みが行われる．この Transfer Data 内で，書き込み単位までデータを受信した場合に，データを復号しダイジェストを計算した上で書き込みを行うように実装する．クライアント側では，書き込み単位毎に分割し暗号化した

データを用意しておけば良い．車内共通鍵で暗号化する場合は，セキュリティ ECU 内で書き込み単位毎の分割と暗号化を実施する必要がある．

5.4 セキュアブートの結果取得

リプログラミング後のリセット時に，セキュアブートを実施し，その結果を暗号化鍵等で暗号化し，メモリ上に保存する．そして，セキュリティゲートウェイが同じく UDS を利用して，結果の取得を行う．サーバ側は，エンド ECU の ID から暗号化・CMAC 鍵を導出し，セキュアブート結果の検証を行う．このとき用いる UDS のサービスは，Read Memory by Identifier など，情報取得のために用いるサービス ID を用いる．

6. おわりに

本稿では，車載ネットワークのアーキテクチャについて整理を行い，この上で実施するセキュアリモートリプログラミング方式を提案した．また，本手法の実現性を，UDS に準拠した設計で検証した．

今回は，設計だけとどめたため，今後の課題として，本手法の実装によるパフォーマンス評価等を行っていく．

参考文献

[1] “HACKERS REMOTELY KILL A JEEP ON THE HIGHWAY – WITH ME IN IT”, https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/, (参照 2016-08-12)

[2] 柳川博彦, “車載情報プラットフォームにおけるセキュリティの研究開発”, デンソーテクニカルレビュー, Vol.8, No.1, http://www.denso.co.jp/ja/aboutdenso/technology/dtr/v08_1/, (参照 2016-08-12)

[3] 平川浩史, “つながるクルマのセーフティ&セキュリティ”, 第 12 回クリティカルソフトウェアワークショップ (12thWOCS2), 1 月, 2015.

[4] Martin Klimke, “Secure and seamless integration of Software Over The Air (SOTA) update in modern car board net architectures”. 13th ESCAR EU, November, 2015.

[5] Idrees, “Secure Automotive On-Board Protocols: A Case of Over-the-Air Firmware Updates”, In Communication Technologies for Vehicle, pages 224—238, Springer, 2011.