Vol.2014-ITS-58 No.8
2014/9/19

セキュアブート＋認証による車載制御システムの保護

竹森敬祐† 溝口誠一郎† 川端秀明† 窪田歩†

車の制御システムに組み込まれた Electronic Control Unit(ECU)の取替え，ECU の制御コードの改竄，Controller Area Network(CAN)へのなりすましパケットの送信など，走行に関わる安全性（Safety）が脅かされている．本研究では，(i)ECU のセキュアブート，(ii)ECU の認証，(iii)CAN パケットの認証，(iv)ECU 向けパッチの署名検証を組み込むことで，車の Safety を脅かす攻撃を根本から排除することを目指す．ECU として，信頼の基点（Root of Trust）を作り出せ，安全なデータ実行・管理領域であるセキュアエレメント（Secure RAM/ROM）を H/W サポートするものを選ぶ．その上で，(i)Root of Trust を基点に，ECU の制御コードが完全な状態であることを測定しながら起動させるセキュアブートを実現する．(ii)起動後は，マスタとなる ECU からエンドの ECU に向けて，Secure ROM で管理される鍵を用いて，チャレンジ・レスポンス認証を行い，制御システムの構成を検証する．認証に成功すると，マスタ ECU からエンド ECU に対して，秘密の情報を送付し共有する．(iii)全ての ECU が正常に起動・認証されると，車が走行可能な状態になる．走行を制御する CAN パケットには，Media Authentication Code（MAC）として，Hash（データ，秘密の情報，パケットカウンタ）を挿入することで，パケット毎に，送信データの完全性，送信元の認証，リプレイ攻撃阻止を担保できるようになる．(iv)車検やリコールにおいて ECU の制御コードを書き換える際には，そのパッチに署名を施しておき，セキュアブートされた ECU で署名検証を行った後に，適用する．これにより，エンジン始動から，走行時，メンテナンスまでの車の Safety を H/W レベルの信頼性で実現する．

Protection for Automotive Control System Using Secure Boot and Authentication

Keisuke TAKEMORI† Seiichiro Mizoguchi†
Hideaki KAWABATA† Ayumu KUBOTA†

Attacks on an automotive control system, ex. illegal replacement of Electronic Control Unit (ECU), tampering of ECU firmware, and packet spoofing in a Controller Area Network (CAN), threaten the safety of driving. In this research, we make a secure automotive control system, which is composed of four techniques; (i) secure boot of ECU, (ii) authentication of ECU, (iii) Authentication of CAN packet, and (iv) authentication of ECU firmware. First, we select an ECU that has both the write protection area called “root of trust” and the secure processing/storage called “secure RAM/ROM” using H/W supports. Next, we propose the secure boot mechanism that measures firmware integrity of the ECU from the root of trust when an engine is started. After the engine starting, a master ECU authenticates end ECUs using the challenge response to verify the configuration of the control system. When the authentication to the end ECUs is succeeded, the master ECU issues a secret value to the end ECUs. To check the integrity of the CAN data, to authenticate the sender ECU, and to avoid the replay attack, a media authentication code (MAC) is inserted in the CAN packet. Here, the MAC is calculated as *hash(data, secret value, packets counter)*. The new ECU firmware is issued and signed by a remote authority. The signature is verified when the new firmware is applied to the ECU. Our proposal guarantees the automotive safety of the driving and the maintenance.

1. はじめに

車の制御システムに対する攻撃として，ECU の取替，ECU の制御コードの改竄，CAN へのなりすましパケットの送信，搭載機器からの情報漏洩などが指摘されている[1-3]．例えば，車内に持ち込んだ PC から CAN へ不正な制御パケットを直接送り込むことで制御を乗っ取る攻撃が実証された[1-2]．また，タイヤの圧力センシング情報が脆弱な無線通信で交換されることに着目し，車の通過履歴を盗聴する攻撃も実証された[3]．車に関わるリスクは，大きく分けて，制御システムへの攻撃によって走行の安全性（Safety）を脅かすものと，搭載機器が持つ情報の漏洩がある．特に前者は，人命に直結するものであり，早急かつ堅牢な対策が求められる[4]．

†1 (株)KDDI 研究所
KDDI R&D Laboratories Inc..

ECU の取替と改竄対策として，マスタとなる ECU が制御ステムを構成するエンドの ECU の制御コードに関するハッシュの期待値を管理しておき，エンジン始動時に測定・比較する手法が提案されている[5]．簡易に実現できる有力な手法であるが，測定処理や管理される期待値の保護策がなく，S/W レベルの堅牢性に留まっている．CAN へのなりすましパケットの送信に対しては，各 ECU が CAN を流れるパケットを監視しておき，送信元 ID をなりすまされた ECU がアラートを発する手法[6]，正規 ECU のみが持つ秘密の情報を含めた MAC を CAN パケットに挿入する手法[5]，パケットカウンタ値を CAN パケットに挿入することでリプレイ攻撃を阻止する手法[7]が提案されている．これらは簡易に実現できる有力な手法であるが，処理に対する保護策に踏み込んだ議論がなく，堅牢性に課題が残る．

ここでメインの演算チップから独立し，安全なデータ処理や管理を担えるセキュリティチップ（セキュアエレメント）を ECU に On-board 化する議論が欧州を中心に進めら

れている．E-safety Vehicle Intrusion Protected Applications (EVITA)プロジェクトでは Hardware Security Module (HSM)の議論が[8]，Hersteller Initiative Software (HIS)では Secure Hardware Extension (SHE) の議論がなされてきた[9]．こうした規格を受けルネサスエレクトロニクスから HSM・SHE 準拠の Intelligent Cryptographic Unit (ICU)が設計され[10]，これを搭載した次世代 ECU として RH850F1x が発表された[11]．セキュアエレメントを持つ ECU の登場により，既存の S/W レベルの対策を堅牢化できうる環境が整ってきた．

ところで，ARM アーキテクチャ[12]を採用するスマートフォンにおいては，Trusted Computing Group (TCG) が規格化した Trusted Platform Module (TPM) [13]を，セキュアエレメントとして活用した H/W レベルの堅牢性を有するセキュアブートが実証された[14]．その後，TPM を SIM/UIM に置き換えたセキュアブートも実証され[15]，セキュアエレメントを内包する ECU 単独でのセキュアブートが現実味を帯びてきた．

本研究では，次世代 ECU を用いて，(i)ECU のセキュアブート，(ii)ECU の認証，(iii)CAN パケットの認証，(iv)ECU 向けパッチの署名検証を組み込むことで，車の Safety を脅かす攻撃を根本から排除する制御システムを実現する．ECU として，書換えを行えない Write Protection を施した Root of Trust を作り出せ，メインの演算チップから直接的な Read/Write/Execution 不能なセキュアエレメント（Secure RAM/ROM）を H/W サポートする RH850F1L を選ぶ．その上で，(i)Root of Trust を基点に，ECU の制御コードが完全な状態であることを測定しながら起動させるセキュアブートを実装する．(ii)セキュアブート後は，マスタとなる ECU からエンドの ECU に向けて，Secure ROM で管理される鍵を用いて，チャレンジ・レスポンス認証を行い，制御システムの構成検証を実現する．エンド MCU が本物であることが認証されると，マスタ MCU からエンド MCU に対して，秘密の情報を送付し，共有する．(iii)全ての ECU が正常に起動・認証されると，車が走行可能な状態になる．走行を制御する CAN パケットには，MAC として，Hash（データ，秘密の情報，パケットカウンタ）を挿入することで，1 パケット単位で，送信データの完全性，送信元認証，リプレイ攻撃阻止を担保する．また，(iv)車検やリコールにおいて ECU の制御コードを書き換える際には，その制御コードに署名を施しておき，セキュアブートされた ECU で署名検証を行った後に，適用する．これにより，エンジン始動から，走行中，メンテナンスまでの車の Safety を担保できるようになる．

2. 車の制御システムへの攻撃

本論文で対象にする Safety を脅かす車載制御システムに対する攻撃マップを図 1 に示す．

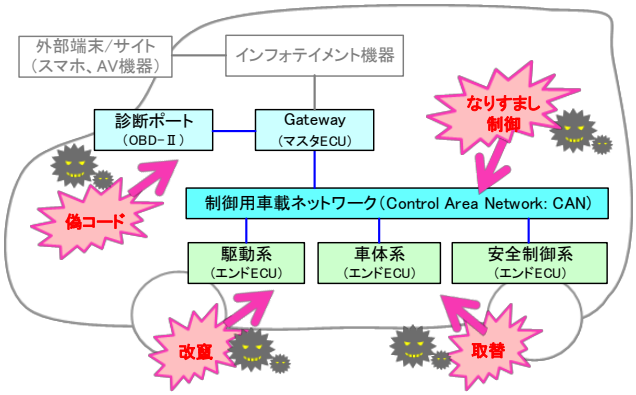

図 1　車載制御システムに対する攻撃

Figure 1　Attack Model on Automotive Control System.

2.1 ECU の取替・制御コードの改竄・偽コードの適用

青空駐車場を狙った窃盗団や愉快犯，町工場での車検やメンテナンスにおける悪意の工員，そして所有者自らによる，ECU の取替，制御コードの改竄が懸念される．また，偽のコードを掴まされ，誤って適用してしまうミスも想定される．ここで ECU 取替は，車内から直接抜き差しすることになる．制御コードの書換えは，ハンドル配下にある診断ポート（On-board Diagnostics Ⅱ：OBD-Ⅱ）に専用の機器を接続して行われる．いずれもローカル攻撃である．

現在，町工場の工員を信頼するモデルで車検やメンテナンスが行われていること，所有者による勝手な改造を見逃していることなど，リモート局から統括的に車の ECU 状態を把握できない状況にある．

2.2 なりすまし制御

偽の CAN パケットによるなりすまし制御として，送信元 CAN の ID を偽る攻撃と，正規 CAN パケットをキャプチャして再送するリプレイ攻撃がある．いずれも車内から直接 CAN に偽のパケットを送り込むローカル攻撃である．

現在の CAN 通信プロトコルは，送信元 ID を CAN フレームに記載してブロードキャストし，各 ECU が受信すべき送信元 ID が記された CAN パケットを取り込む仕様である．送信元 ID に対する認証は行われず，CAN パケットに記載された送信元 ID を信じるモデルである．

3. 既存の対策技術

3.1 ECU の制御コードの改竄検知

マスタ ECU が車の制御システムを構成する ECU の制御コードに関するハッシュの期待値を管理しておき，エンジン始動時に測定・比較する手法が提案されている[5]．

簡易に改竄検知を行える有力な手法であるが，測定処理や期待値に対する改竄攻撃に対して，保護策が考慮されておらず，S/W レベルでの堅牢性に留まっている．

ARM アーキテクチャから構成されるスマートフォンにおいては，eMMC フラッシュメモリ[16]と TPM を活用することで，H/W レベルの堅牢性を有するセキュアブートが提

案・実装されている[14]．これは，e-MMC の Write Protect された固定領域を Root of Trust にして，ここに Boot Loader とハッシュ測定処理を置くことで，Boot 処理とハッシュ測定処理に対する改竄攻撃を防ぐ．また，外部からアクセス不可な TPM の演算/データ管理（Secure RAM/ROM）領域で，カーネル層より上層の制御コードのハッシュの期待値を管理しておく．そして，起動時に Root of Trust から TPM へ測定値を渡して，TPM 内で期待値と比較することで，スマートフォンが完全な状態で起動したことを保証する．

車においても，ECU に TPM を搭載して堅牢化を図る議論がなされている[17]．しかし，セキュアエレメントとして ECU の制御コードの完全性の検証やパッチに付された署名検証のフレームは提案されているものの，Root of Trust の議論が欠けている．これによりセキュアブートが H/W レベルの堅牢性で担保されなくなり，その上に構築される署名検証や認証処理の信頼性が揺らぐ．また，各 ECU に TPM を追加・搭載することへのコストの増加も懸念される．

その後，スマートフォンにおいては TPM を SIM/UIM に置き換えたセキュアブートが提案・実装された[15]．これは，ハンダ付けされた TPM の代わり，挿抜可能な SIM/UIM を Secure RAM/ROM として用いる技術であり，ARM の Trust-zone に認証用の鍵を管理しておき，SIM/UIM 側からチャレンジ・レスポンス認証を行うことで，SIM/UIM を ARM ボードにバインドさせる．これにより，新たなデバイスの追加なく，スマートフォンにおける堅牢なセキュアブートと認証・署名検証を実現できるようになった．

3.2 ECU の真贋判定

電子回路固有の物理特性として，計算時間や電流量などの僅かな差に着目し，個々の ECU を区別する Physical Unclonable Function (PUF)と呼ばれる真贋判定技術がある[18]．秘匿すべき認証プログラムが漏洩・コピーされたとしても，電子回路固有の物理特性までコピーすることが困難なことに基づく真贋判定の手法である．

ECU の取替え検知に有力視されているが，真贋判定のための特殊な測定装置が必要である．事故などのインシデントにおける事後分析には利用できるものの，日常的なエンジン始動時の検査には適していない．

3.3 なりすまし CAN パケットの監視

各 ECU がブロードキャストされる CAN パケットの送信元 ID を監視しておき，自身が発信していないときに，自身の ID と一致する CAN パケットを受信すると，周囲の ECU に異常を知らせる手法が提案された[6]．

簡易になりすまし ECU を検知できる有力な手法であるが，アラートを発信すべき本来の ECU が取り外されてしまうと機能しない．

3.4 MAC による CAN パケット認証

各 ECU がセキュアブートした後に，マスタ ECU からエンド ECU に秘密の情報を送付・共有し，これを用いた MAC を算出して，CAN パケットに挿入する手法が提案された[5]．

MAC＝Hash（前 4 つ分の CAN データ部，秘密の情報）

MAC は，4 つ分の CAN パケットのデータ部と秘密の情報から算出し，64bit のデータを得る．これを，その後の 4 つの CAN パケットの CRC 部（16bit）に埋め込む．受信側は，4 つ分の CAN パケットを受け取ると，MAC を算出して，データに改竄がないこと，正しい ECU から送信されていることを確認する．

簡易に CAN パケットの改竄検知と送信元認証を行える手法であるが，4 パケット揃った後に検証を行うため，リアルタイム性が要求される車載制御においては，遅延は許容し難い．

3.5 CAN リプレイ攻撃の阻止

正規の CAN パケットを盗聴して，これをコピー・送信するリプレイ攻撃への対策技術が提案されている[7]．これは，ECU に送受信したパケット数をカウントする機能を持たせておき，CAN パケットを送信した後に，

Hash（CAN データ部，カウンタ値）

を，次の CAN パケットに挿入して，送信する手法である．

簡易にリプレイ攻撃を阻止できる有力な手法であるが，パケット数が 2 倍に増加してしまうこと，1 パケット分の遅延が生じてしまう問題がある．

4. 提案・実装

本論文で提案する車の制御システムに対する攻撃対策の概要を図 2 に示す．(i)ECU のセキュアブート，(ii)ECU の認証，(iii)パケットの認証，(iv)コードの署名検証である．これらを，H/W レベルの堅牢性で実現することを本論文の目標とする．はじめに ECU に用いるマイコンや開発環境の選定を行い，その上で(i)～(iv)を構築していく．

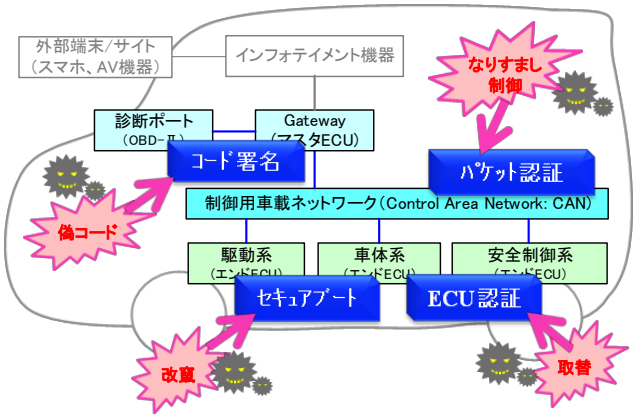

図 2　車の制御システムに対する攻撃対策

Figure 2　Countermeasures against Automotive System Attack.

4.1 エンド ECU の選定・環境

HSM [8]や SHE [9] 準拠の ICU [10]を搭載する ECU として，ルネサスエレクトロニクス製の RH850F1L [11]がある．

これはローエンドな ECU であり，制御システムを構成する ECU の多くに適用が見込まれる．この ECU でセキュアブートや認証を実現できると，これよりハイエンドな ECU の全てにおいても同様な実装を行えることになる．

RH850F1L の開発・評価用ボードとして，テセラ製 FL-850/F1L-176-S を用いる[19].

ソフトウェア開発環境として，ルネサスエレクトロニクス製 CubeSuite+ for v850 を用いる[20].

RH850F1L 向け OS として，TOPPERS ATK2 [21]の最新のコードを独自にチューニングして適用することにする．

エンド ECU である RH850F1L を搭載した 2 つの FL-850/F1L-176-S を CAN 接続した開発の様子を図 3 に示す．

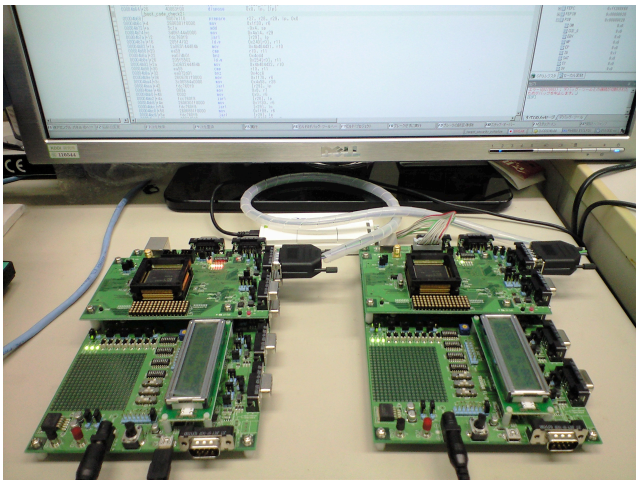

図 3　開発に用いた RH850F1L と評価ボード
Figure 3　RH850F1L on the Development Board.

4.2 マスタ ECU の選定・環境

同じく ICU [10]を搭載する ECU として，RH850F1M が候補としてあるが，まだ研究開発に利用できる段階ではない．そこで[15]で実績のある，ARM 系ボード Freescale i.MX6Q Sabre-SD [22]に Qualcomm Gobi 2000 Mini-PCI Express 3G モデムカード経由で SIM/UIM(Java card)を接続した．OS として，Android 4.2.2（Linux3.0.35）を適用した．

将来的には，挿抜可能な SIM/UIM から On-board 化できうる eSIM に置き換えることや，RH850F1M に置き換えたい．参考までに，eSIM を内包した通信モジュールとして KYM11 がリリースされている[23].

4.3 セキュア ECU の設計

図 4 に，マスタ ECU とエンド ECU の機能設計を示す．

マスタ ECU である i.MX6Q は拡張性が高く，所望の処理を柔軟に組み込める[15]. 本論文では，非対称鍵暗号を H/W サポートしないなどの制約の多いエンド ECU に注目した議論を進める．

エンド ECU となる RH850F1L に，Write Protection を施した Root of Trust を用意し，不変な Boot Loader とセキュアエレメントとのインタフェース(IF)を，Write Once で書き込む．

RH850F1L には，安全な実行処理・データ管理領域であるセキュアエレメント（ICU-S と呼ばれる Secure RAM/ROM）が標準搭載されている．Secure RAM には，対称鍵暗号である AES，ファイル測定のための Cipher-based MAC (CMAC)，乱数生成の機能が H/W 実装されている．また，起動後に ECU 間で共有される秘密の情報を管理する機能を設ける．Secure ROM には，複数の暗号鍵と，制御コードの期待値を管理する領域が H/W 実装されている．

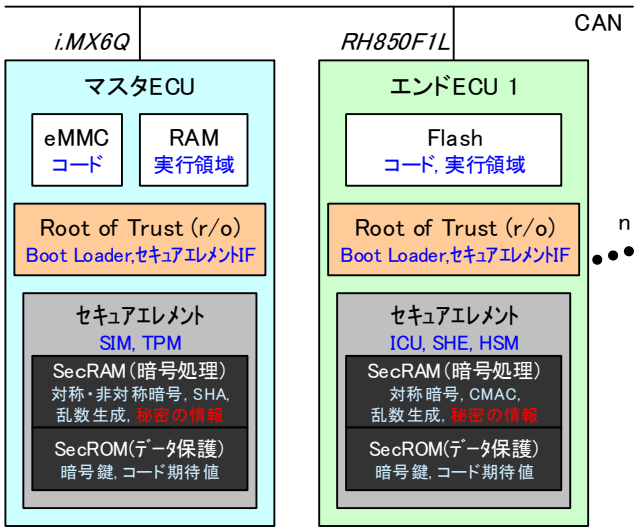

図 4　セキュア ECU システムの設計
Figure 4　Design of Secure ECU System.

4.4 ECU のセキュアブート

エンジン始動時の RH850F1L のセキュアブートを以下の手順で実施する(図 5). 前提として，制御コードの測定には，BOOT_MAC_KEY と呼ばれる鍵を用いた CMAC 演算が Secure RAM で行われる．Secure ROM には BOOT_MAC と呼ばれる領域に，CMAC の期待値が予めセットされている．

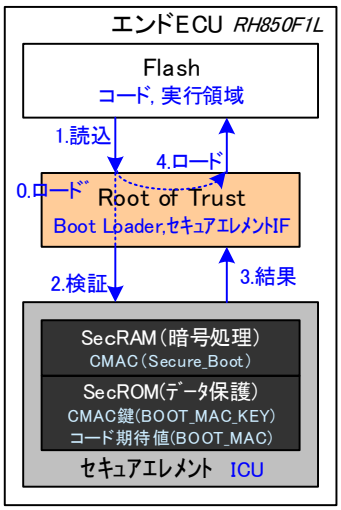

図 5　エンド ECU (RH850F1L) のセキュアブート
Figure 5　Secure Boot for End ECU (RH850F1L).

Step 0) Boot Loader とセキュアエレメント IF をロードする.
Step 1) 可変の制御コードが Flash メモリから, Root of Trust のセキュアエレメント IF を通じて, Secure RAM の CMAC 処理に渡される.
Step 2) Secure ROM の CMAC 鍵(BOOT_MAC_KEY)を用いて Secure RAM で制御コードの CMAC(Secure_Boot) 演算が行われる. この値と, Secure ROM で管理される CMAC の期待値(BOOT_MAC)を比較する.
Step 3) 一致/不一致の結果を Root of Trust に返す.
Step 4) 結果が一致していれば, 制御コードが完全であると判断され, Flash 上の制御コードが書き込まれている番地にジャンプ(ロード)する. 不一致であれば, 起動を停止するなど, エラー処理に進む.

4.5 ECU の認証

セキュアブートが完了し, 個々の ECU が完全な状態で起動すると, 制御システムに偽の ECU が混入してしないか, マスタ ECU からエンド ECU に向けてチャレンジ・レスポンス認証を行い, 構成検証を進める（図 6）.

尚, 認証には公開鍵・秘密鍵で処理する非対称暗号が望ましいが, RH850F1L で H/W サポートが無いこと, S/W 実装しても処理速度が望めないことから, H/W サポートのある対称鍵(K)を用いて行うこととする. 対称鍵(K)は予め Secure ROM にセットされていることを前提とする.

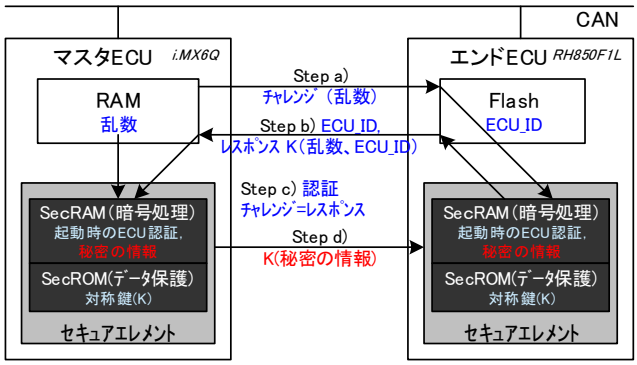

図 6 ECU 認証と秘密の情報の共有
Figure 6 ECU Authentication and Secret Share.

Step a) マスタ ECU の RAM で乱数を生成し, エンド ECU へチャレンジとして送付する.
Step b) エンド ECU は, 受け取った乱数と ECU_ID を Secure RAM に渡し, Secure ROM で管理される対称鍵 K で暗号化 K(乱数, ECU_ID)する. これをレスポンスとしてマスタ ECU へ返信する. このとき CAN パケットには, 送信元を示す ECU_ID が付される.
Step c) マスタ ECU は, 受け取った K(乱数,ECU_ID)と ECU_ID を Secure RAM に渡し, Secure ROM で管理される対称鍵 K で K(乱数,ECU_ID)を復号する. そして送信した乱数と復号した乱数, ECU_ID が一致することを確認し, エンド ECU を認証する.
乱数＝K･K(乱数), ECU_ID＝K･K(ECU_ID)
Step d) 認証に成功するとマスタ ECU の Secure RAM で, 新たな乱数である秘密の情報を生成し, Secure ROM で管理される対称鍵 K で暗号化して, エンド ECU へ送付する.
エンド ECU は受け取った K(秘密の情報)を, Secure ROM で管理される対称鍵 K で復号し, Secure RAM で管理する.

Step a)～d)により, 制御システムを構成する ECU 群の認証が完了し, マスタ ECU が生成した秘密の情報を正規の ECU のみが安全に共有することになる. この秘密の情報は, エンジン始動毎にマスタ ECU が生成した乱数とし, 1 つの車両を構成する全ての ECU で共通の値とする.

4.6 CAN パケットの認証

ECU は, 送信する全ての CAN パケットに MAC を挿入する(図 7). MAC には, データの完全性と送信元認証を担保するために, CAN フレームのデータ部と事前に共有した秘密の情報を含める. また, リプレイ攻撃を阻止するために, ECU の RAM で, 自身が送信したパケット数"s"をカウントし, これも MAC に含める. 各 ECU から受信したパケット数"r"もカウントしておき, MAC 検証に利用する.

MAC＝Hash（データ, 秘密の情報, カウンタ）

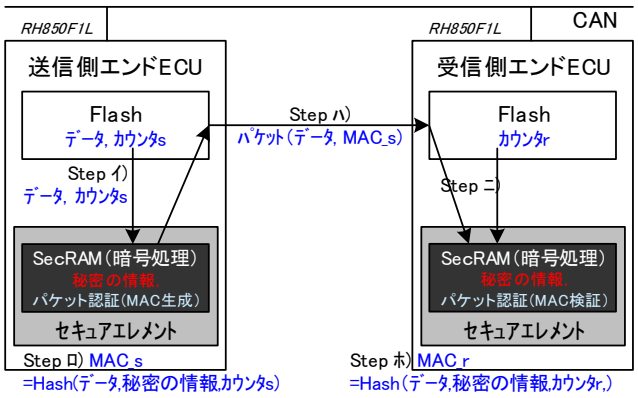

図 7 MAC による CAN パケット認証
Figure 7 CAN Packet Authentication using MAC.

Step イ）送信側 ECU は, Flash からデータ, パケットカウンタ"s"を Secure RAM に渡す.
Step ロ）Secure RAM は, 起動時に共有しておいた秘密の情報を加えて, MAC を算出する.
MAC_s=Hash（データ, 秘密の情報, カウンタ"s"）
Step ハ）Secure RAM は, 算出した MAC を Flash に渡し, CAN にブロードキャストする.
受信側 ECU は, 受け取るべき CAN パケットの送信元 ECU ID を監視しておき, 所望の CAN パケッ

トを取り込む．

Step ニ）受信側 ECU は，CAN パケットのデータ，MAC_s，送信元 ECU ID に該当する受信パケットカウンタ”r”を，Secure RAM に渡す．

Step ホ）Secure RAM は，データ，起動時に共有しておいた秘密の情報，パケットカウンタ”r”から MAC を算出する．そして，MAC_s＝MAC_r を検証することで，データの完全性，送信元認証，リプレイ攻撃阻止を担保する．

MAC_r＝Hash（データ，秘密の情報，カウンタ”r”）

ここで，現在の CAN フレームにないパケットカウンタを新たに設けることで，送信側と受信側のカウンタ値がずれてしまうことでの同期外れを防ぐ必要がある．また，CAN フレームのサイズは小さく，挿入できる MAC のビット数が限られる問題がある．この 2 つの課題を解決するために，MAC を小さく分割し，カウンタ値に従った分割位置の X ビットを抽出・比較する手法を，以下に提案する．

【MAC 分割・抽出・比較】

パケットカウンタを上位 L-1-n ビットと，下位 n ビットに分割する．MAC に含めるカウンタは上位ビットとする．算出された MAC を K 分割し，下位ビットが指す値を分割された MAC フレームの抽出位置とする．

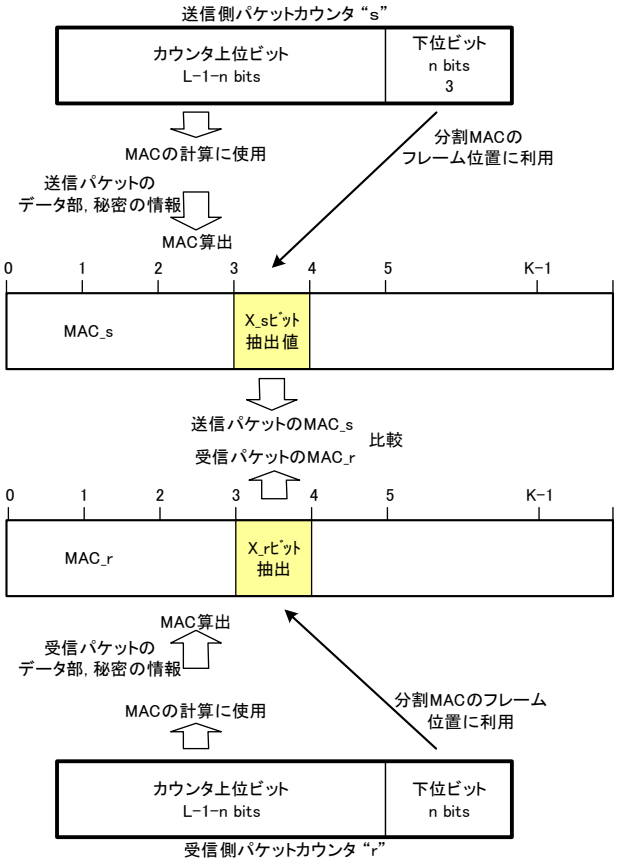

図 8　MAC の分割・抽出と比較

Figure 8　Division and Comparison of MAC.

送信側の ECU でカウンタ”s”から算出・抽出された X_s ビットと，受信側 ECU でカウンタ”r”から算出・抽出された X_r ビットを比較し，一致すれば，MAC 認証が成功したとみなす．この様子を図 8 に示す．

【パケットカウンタ同期】

もし下位ビットが指すフレーム位置の X ビットの比較において不一致が生じると，カウンタの同期外れか，攻撃が行われたことになる．そこで，受信側 ECU では，分割・抽出した X ビットの隣のフレームを比較する．この作業を，最大 K-1 回実施する．もし一致する X ビットが検出されれば，カウンタの同期外れとみなして，一致したフレーム位置を新たなカウンタ”r”として補正する．もし一致するフレームが無い場合には，攻撃がなされたと判断する．この様子を図 9 に示す．

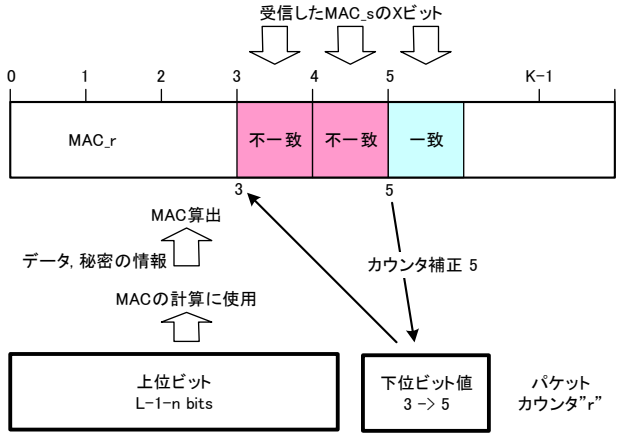

図 9　パケットカウンタの同期補正

Figure 9　Synchronism for Packet Counter.

ここで，カウンタを上位と下位ビットに分割して，上位ビットで MAC を算出し，下位ビットでフレーム位置を特定するアルゴリズムに注目されたい．上位ビットは，最終の K-1 フレームまで変化しないため，1 フレームずつスライド比較する処理において，MAC の再計算を要しない．戻って 0 フレームから比較するときのみ，下位ビットの桁上がりが生じて，新たな上位ビットを用いた MAC の再計算を 1 回行う．このように，同期補正に要する処理負荷を軽減した方式になっている．

サイズ制限のある CAN フレームに挿入するために，K 分割しているが，これに伴い挿入ビット数が短くなる．これは，MAC の衝突確率が増すことを意味する．例えば，16 ビットの場合，2 の 16 乗＝1/65536 の確率で衝突する．ここで車載制御を考える．例えばアクセルを踏み続ける限りにおいて，連続的に CNA パケットが送信される．連続的に MAC が衝突する確率は，指数関数的に減少する．このため，攻撃者が連続してアクセルを踏み続ける不正制御に成功する確率は格段に下がる．暗号学においては見逃せない数値であるが，車載制御においては許容できるレベル

ではないであろうか．CAN フレームに MAC を挿入する場合，64 ビット長のデータ部もしくは 16 ビット長の CRC 部が候補になる．許容できる衝突確率と，挿入サイズのバランスを加味して，ビット長を決めると良い．

4.7 ECU 制御コードの署名・検証

車検やリコールなどのメンテナンスの際に，ECU の制御コードを書き換える必要がある．このとき，正しい制御コードのみが適用されるよう，制御コードに署名を付しておき，ECU で署名検証を行った後に適用する方式を提案する．署名は，車メーカなどのセンタ局側で付し，検証用の鍵は ECU の Secure ROM に保存しておく．

署名検証処理が改竄されると，検証の信頼性が担保されない．そこで ECU のセキュアブートにおける測定対象として，署名検証処理も含めることにする．

尚，ECU 認証，CAN パケット認証に関わる処理についても，セキュアブートの測定範囲に含める．この様子を図 10 に示す．セキュアブートと認証によって，エンジン始動時，走行時，メンテナンス時をトータルに，Safety を担保できるようになる．

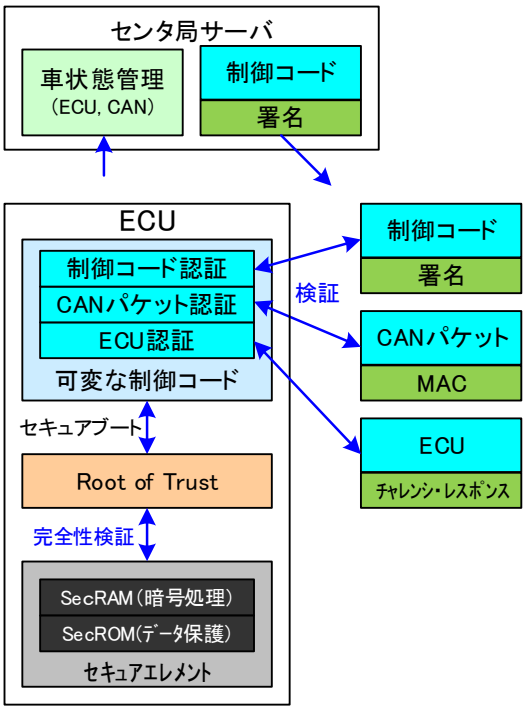

図 10　セキュアブートと制御コードの署名検証

Figure 10 Verification of ECU_code Signature on Secure Boot.

4.8 センタ局によるリモート管理

車メーカなどが運営するセンタ局では，出荷した車の状態を統合管理できるよう，図 11 に示すような状態監視システムを構築する．ここでは，個々の車の状態として，CAN 通信，ECU の正常/異常を一覧表示している．これにより，異常な車の検知，車検やリコールの進捗確認，事故が発生したときの責任分解が行えるようになる．

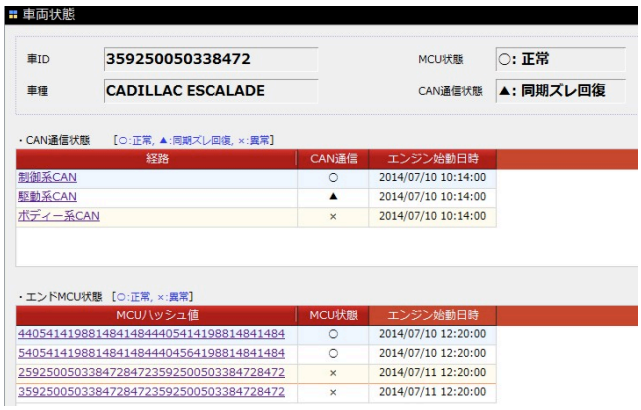

図 11　車のリモート管理局

Figure 11　Remote Management System for Automotive.

5. 評価

評価として，エンジン始動時に行われるエンド ECU である RH850F1L の CMAC を用いたセキュアブートに要する処理時間を測定した．測定対象は，TOPPERS ATK2 であり，43.6Kbyte である．結果は，60.0msec であった．CMAC 演算に H/W サポートがあり十分な高速性を達成できている．

将来的には，マスタ ECU を，RH850F1M へと置き換え，セキュアブートの処理時間の計測と，マスタ ECU からエンド ECU に対する認証の処理時間も測定したい．

6. おわりに

本論文では，車の制御に関わる ECU の全てをセキュアブートさせ認証すること，CAN パケット単位で認証すること，パッチに署名を付し検証することで，エンジン始動時から，走行時，メンテナンス時の Safety を H/W レベルの信頼性で実現する手法を提案した．この信頼の連鎖の様子を図 12 に纏める．これは，昨今報告されている ECU や CAN に対する攻撃を根本解決できると伴に，青空駐車場，町工場の工員，所有者を信頼することを前提にした車の利用・保守モデルに対して，リモートから監視・検証できる新たなフレームとして期待される．

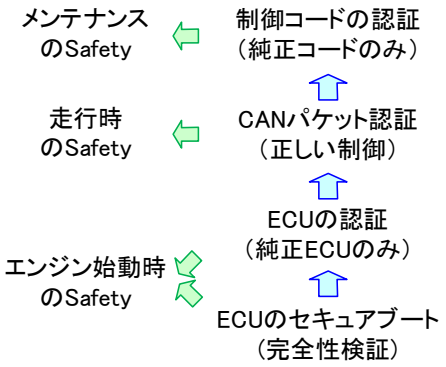

図 12　始動・走行・メンテナンスの安全性

Figure 12　Safety of Start, Driving, and Maintenance.

今後は，認証・署名検証に用いる暗号鍵の発行・配布，廃棄の仕組みについて提案・実装を進めていく．

CAN は枯れた・信頼された車載制御ネットワークとして広く普及し，その仕様変更は難しい．高速・大容量化を目指した次世代の車載ネットワークとして，CAN-FD [24]，Ethernet-AVB [25]，Time-Sensitive Networking (TSN) [26]などの規格化が進められている．こうした新規格の制御ネットワークに本パケット認証の仕組みの適用を目指したい．

謝辞 本研究を進めるにあたり，適切なコメントならびに開発支援を頂いた ルネサスエレクトロニクス株式会社 押田様，山梨様，塩田様に，謹んで感謝申し上げます．

参考文献

1) Karl Koscher, Alexei Czeskis, Franziska Roesner, Shwetak Patel, and Tadayoshi Kohno, "Experimental Security Analysis of a Modern Automobile", IEEE Symposium on Security and Privacy, May, 2010.
2) Stephen Checkoway, Damon McCoy, Brian Kantor,Danny Anderson, Hovav Shacham, and Stefan Savage "Comprehensive Experimental Analyses of Automotive Attack Surfaces", USENIX Security, August, 2011.
3) Charlie Mille and e Mille, "Adventures in Automotive Networks and Control Units", DEF CON 21, August, 2013.
4) 高田広章，松本勉，"載組込みシステムの情報セキュリティ強化に関する提言"，IPA，2013 年 9 月．
https://www.ipa.go.jp/files/000034668.pdf
5) 吉岡顕，小熊寿，西川真，繁富利恵，大塚玲，今井秀樹，"構成証明機能を持つ車内通信プロトコルの提案"，情報処理学会，DICOMO2008，pp.1270-1275，2008 年 7 月．
6) 畑正人，田邉正人，吉岡克成，大石和臣，松本勉，“不正送信阻止：CAN ではそれが可能である”，情報処理学会，CSS2011，pp.624-629，2011 年 10 月．
7) 特許：小熊寿，松本勉，畑正人，田邉正人，吉岡克成，大石和臣，"通信システムにおけるメッセージ認証方法および通信システム",
https://www.google.com/patents/WO2013065689A1?cl=ja&dq=CAN+message+authentication+ECU&hl=ja&sa=X&ei=0VEhU5OZMcmulAWp4YG4Ag&ved=0CDgQ6AEwA
8) EVITA Project, Hardware Security Module,
http://www.evita-project.org/
http://www.evita-project.org/EVITA_factsheet.pdf
9) HerstellerInitiative Software (HIS), Secure Hardware Extension (SHE),
http://portal.automotive-his.de/index.php?option=com_content&task=view&id=31&Itemid=41&lang=english
10) ルネサスエレクトロニクス，"Security in Automotive Applications" and "ICU", DevCon 2013,
http://www.renesasinteractive.com/file.php/1/CoursePDFs/DevCon_2012/Security/BC05I_FabricePoulard_SecuritySolutionsfortheAutomotiveIndustry_0920_final.pdf
11) ルネサスエレクトロニクス, RH850F1L,
http://japan.renesas.com/products/mpumcu/rh850/rh850f1x/rh850f1l/index.jsp
12) ARM，http://www.arm.com/ja/
13) TCG (Trusted Computing Group),
http://www.trustedcomputinggroup.org/
14) 竹森敬祐, 川端秀明, 磯原隆将, 窪田歩, "Android(ARM)+TMP によるセキュアブート"，電子情報通信学会，SCIS2013，4C1-4, 2013 年 1 月．
15) 竹森敬祐，川端秀明，窪田歩，"ARM＋SIM/UIM によるセキュアブート"，電子情報通信学会，SCIS2013，1Ba-2，2014 年 1 月．
16) eMMC，東芝 セミコンダクター&ストレージ社,
http://www.semicon.toshiba.co.jp/product/memory/selection/nand/mlc/emmc/index.html
17) TCG TPM 2.0 Automotive Thin Profile, June, 2014,
http://www.trustedcomputinggroup.org/files/static_page_files/BAA6C75F-1A4B-B294-D0DBC6E5EBDCDD85/TPM%202%200%20Library%20Profile%20for%20Automotive-Thin_v0.91.pdf
18) 中野将志，鵜飼慎太郎，柴谷恵，久保田貴也，汐崎充，藤野毅，"サイドチャネル攻撃対策 AES 暗号と PUF 技術を用いた車載向け耐タンパ認証システムの設計と実装"，信学技報, vol. 113, no. 498, DC2013-93, pp. 139-144, 2014 年 3 月．
19) テセラ，FL-850/F1L-176-S,
http://www.tessera.co.jp/fl/f1l-176.html
20) ルネサスエレクトロニクス，CubeSuite+ for V850
http://www.digikey.jp/product-search/ja?vendor=0&keywords=CUBESUITE+for+V850
21) TOPPERS ATK2，https://www.toppers.jp/atk2-download.html
22) freescale i.MX6 SABRE-SD,
http://www.freescale.com/ja/webapp/sps/site/prod_summary.jsp?code=i.MX6Q
23) KDDI, KYM11,
http://www.kddi.com/business/mobile/m2m-solution/domestic-m2m/product/kym11/
24) CAN-FD，bosch, “CAN with Flexible Data Rate”, April, 2012.
http://www.bosch-semiconductors.de/media/pdf_1/canliteratur/can_fd_spec.pdf
25) Ethernet AVB, IEEE 802 Audio Video Bridging Task Group,
http://www.ieee802.org/1/pages/avbridges.html
26) TSN，IEEE 802 Time-Sensitive Networking Task Group,
http://www.ieee802.org/1/pages/tsn.html
27) IPA，”2011 年度自動車の情報セキュリティ動向に関する調査”，http://www.ipa.go.jp/files/000024413.pdf
28) IPA，”自動車の情報セキュリティへの取組みガイド”
http://www.ipa.go.jp/files/000027273.pdf