将来の自動車社会に向けた HW/SW セキュリティメカニズム

Automotive System Security Department, High Performance Computing, Analog Power Core Technology Development Division, High Performance Computing, Analog Power Solution Group

Renesas Electronics Corporation

森山　大輔

イントロダクション

自動車に対しての情報セキュリティの需要は、車両の電動化やコネクティビティの普及が進まるにつれてここ数年で非常に高まっている。2000 年前後からインターネットの普及に伴い徐々に発展してきたネットワークセキュリティと比較すると、自動車のサイバーセキュリティ対応の変化はとても急速である。また、2010 年台中盤からセキュリティ対策の不備や脆弱性を突いた自動車へのハッキングや盗難なども報告されるようになってきている。

自動車は今や単なる移動手段としてだけではなく、ノート PC のように情報機器として扱い、コンピュータセキュリティが十分に確保されていることが重要である。ノート PC に積まれている CPU は 1 台だが、自動車 1 台当たりの ECU (Electronic Control Unit)の数は現在 50-100 基程度搭載されていると言われている。将来的には完全自動運転システムのような技術的進歩によりさらに多くの ECU が搭載されることが予想される。パワートレインや ADAS (Advanced Driver-Assistance Systems)といった機能を制御する各 ECU が安全にコミュニケーションを取り、悪意のある攻撃に対抗する必要がある。そのような背景により、自動車向けの半導体製品についても様々な情報セキュリティ技術を導入することが求められている。

本ホワイトペーパーでは、最新の自動車向け半導体のためのセキュリティ技術や将来的な技術に関係する話題を紹介する。特にどのような場面でセキュリティが必要とされ、どのような技術によって守られているかについて個別に分類して説明する。

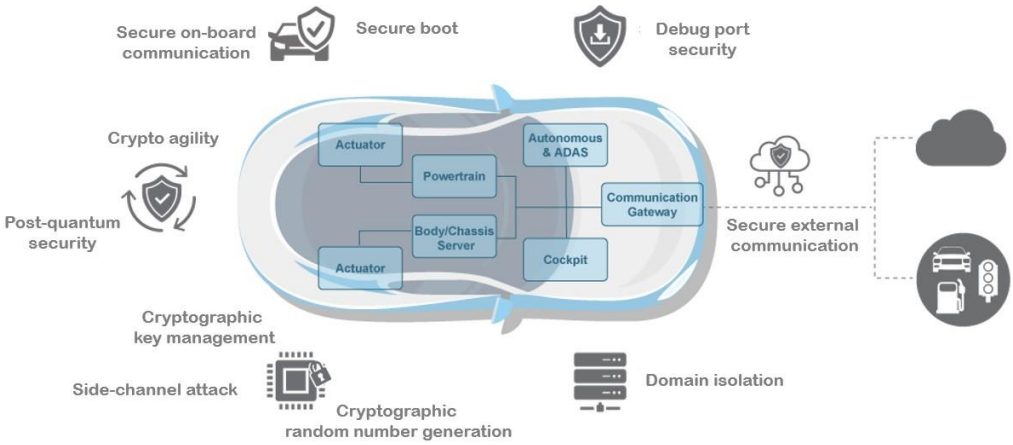

図 1 自動車向けのセキュリティ技術

自動車向け MCU/SoC のセキュリティ要素

最新の自動車向けの MCU/SoC は車内通信の電子化を受けて様々なセキュリティ機能を提供している。これらの製品は起動プログラムの完全性検証、なりすましを防ぐための送受信間での正当性検証、データ盗聴を防ぐための暗号化など様々な脅威に対して必要に応じてハードウェアおよびソフトウェアの対策を有している。

Renesas Electronics を含む多くの半導体ベンダーが販売している自動車向け MCU/SoC ソリューションの中には HSM (Hardware Security Module)が搭載されている。特に高性能な MCU/SoC の HSM には一般的な application CPU とは独立した専用の CPU を備え、またブロック暗号、ハッシュ関数、公開鍵暗号、乱数生成器などの暗号ハードウェアを搭載している。これは欧州委員会によって定められた一番セキュリティレベルの高い EVITA (E-safety vehicle intrusion protected applications) full を満たすための要件である [1]。HSM が Root of Trust として扱われ、チップ全体に対して安全にセキュリティ機能を提供できるよう資産保護やアクセス管理などの責任を受け持つ。

一方、自動車内に搭載される半導体が増えるにつれて個々の MCU/SoC が処理するデータ量も増えてきている。LIN, CAN や CAN-FD といった従来の通信インターフェースだけでなく、Gbps クラスの広帯域 Ethernet も現在は車内間の通信インターフェースとして捉えられている。また自動運転のためのセンサーやカメラデータ、ドライブレコーダに保管される情報にも完全性の担保が望ましい。そのような事情により、1 つのチップに搭載されている 1 つの HSM のみですべての暗号処理を処理することが難しくなってきている。そのため近年の半導体製品では、高いセキュリティ需要に応えられるよう HSM の外側にもセキュリティ処理を担うモジュールを搭載するようになってきている。

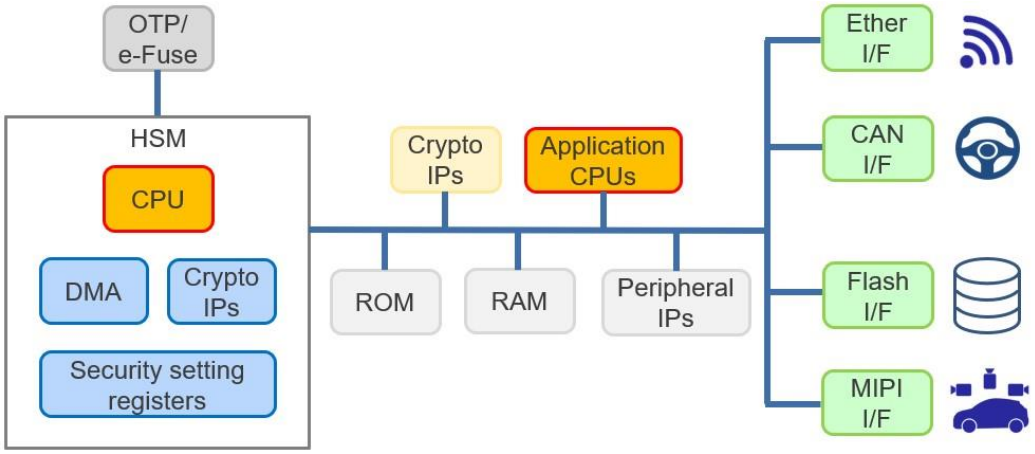

図 2 セキュリティ関連モジュールと利用される通信経路

自動車がその時点での堅牢なセキュリティ対策を維持するためには、どこからどのように攻撃が試みられるのか、何をゴールとしているのかを見定めて、個々に（あるいは共通的な技術により）対抗策を導入する必要がある。次の章ではセキュリティが必要であると想定される場面を MCU/SoC の観点からいくつか取り上げ、どのような対策技術が実装されている（あるいは必要とされている）かについて解説する。

現在および将来考えるべきセキュリティ

[MCU/SoC 内部のセキュリティ]

- ### Secure Boot

安全で堅牢なプログラムを実行するための最初のステップは、最初に起動するファームウェアや基本ソフトウェア自身に対しての信頼性と完全性を最初に確認した上で実行することである。マルウェアが安易にインストールされない環境を構築するためには、起動時に規定されているプログラムがすべて正規のものであるかを確認することが重要である。ハードウェアソリューションとしては、MaskROM や OTP (One Time Program) などの書き換えが行われない領域に一番最初に起動すべきコードを書き込んでおき、HSM のための制御プログラムから順に起動させるべきプログラムの正当性を順に検証することで次のプログラムの正当性をブートチェーンとして保証する。ブートデータの対象を外部領域に保存する必要がある場合は事前に暗号化しておき、ブートする毎に復号することで機密性を確保する。正当性の検証には電子署名やメッセージ認証子などの暗号技術を用いるため、早い段階で HSM と暗号ハードウェアを制御するプログラムの Secure boot を行うことが望ましい。より詳細な Secure boot に関する記事は[2]を参照のこと。

最新の自動車向け MCU/SoC では application 向けにはマルチコアの CPU を搭載しており、それぞれの CPU コア毎に異なる役割が割り当てられ、異なるソフトウェアを実行することが想定される。そのため、優先度が高いプログラムが最短時間で起動されるように、CPU や動作させたい実行プログラムごとに区分けして Trust chain を構築し順に Secure boot を実行することが望ましい。

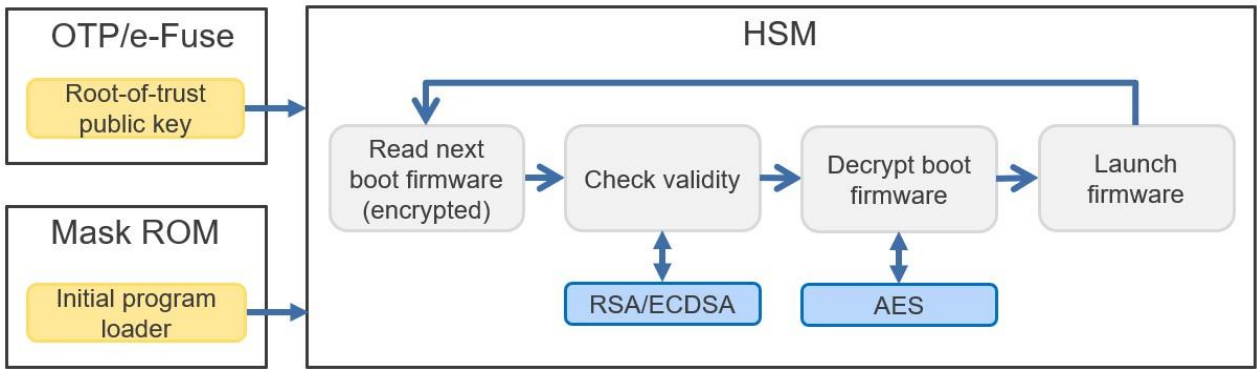

図 3 Secure boot

- ### Domain Isolation

一般に HSM 内の CPU を基点に安全にデータの制御を行うためには、HSM と application CPU とはアクセスできる領域（不揮発性メモリ、SRAM、外部ストレージ、各種内部モジュールのレジスタ）を個別に制御する必要がある。また、application CPU についても、CPU 毎あるいはクラスタ毎に柔軟にアクセス権限を管理し個別データを隔離したい場合は、それぞれのドメインを区別するための一意の ID が必要となる。

それぞれのアクセス先のアドレス領域に対して属性（アクセス不可、リードのみ、リードライト可）が割り当てられ、バスマスターがアクセスする場合に常にハードウェアによって判定が行われる。チップに SRAM や DRAM が接続されている場合であっても、領域分割して個々にアクセス権を設定したいユースケ

ースが多い。この管理は一般的に MPU (Memory Protection Unit)によって制御される。MPU は一定の単位（例えば 4KB 境界）で設定が可能なメカニズムを持っている。このアクセスルール自身を悪意のある攻撃者が書き換えることを防止するため、HSM のみが一連の設定を行うことが望ましい。

- Cryptographic Key Management

自動車社会では Secure on-board communication として主に AES の CMAC モードが用いられる。また不特定多数の端末との信頼性を確認する V2x 通信においては電子署名が用いられることが期待される。このような例だけでなくどのような状況においても、暗号学的秘密鍵は安全に保持し管理しなければならない。EVITA Medium 以上を満たす MCU/SoC であれば、専用 CPU が HSM 内に存在する。そのため、その専用 CPU を用いてすべての秘密鍵をハンドリングすることができる。

秘密鍵を安全に保持する方法については、不揮発性メモリ（例えば Flash）がチップ内に内蔵されているか、あるいはチップ外に備え付けられているかに依存する。もし不揮発性メモリがチップ内に混載されているならば、memory protection unit を用いて一定範囲の領域を HSM のみがアクセスできるよう制限する方法が簡単である。もし外部に存在する不揮発性メモリのみが利用できる場合は（先端プロセスでの Flash 混載は技術的に難しい）、電源が OFF の状態であっても保持すべき秘密鍵は認証暗号で暗号化した上で外付け領域に保存する。また、Root of Trust の起点として利用する秘密鍵は暗号化できないため OTP メモリに保管して、Secure boot 実行中に HSM から読み出されるようにする。

図 2 でも示していたように、最近の自動車向け半導体製品では HSM の外側にも複数の暗号ハードウェアをアクセラレータとして搭載し、Application CPU が利用することで高いスループット/低いレイテンシのセキュリティ処理を行うことを期待している。ただし、このようなセキュリティ機能を追加することによって生じるセキュリティ上の問題はカバーしなければいけない。もし複数の application CPU が同じ暗号アクセラレータにアクセスすることを許容するのであれば、ある秘密鍵が異なる用途で利用されないよう一連の利用が終了した後で必ずそのモジュールの内部状態をリセットしなければいけない。

- Cryptographic Random Number Generation

暗号技術には乱数が欠かせない要素の一つとなる。例えば公開鍵暗号における鍵生成のシードや、共通鍵暗号の秘密鍵として直接利用される。もしこの値が簡単に予測可能である場合、どのような理論的に安全な暗号アルゴリズムを実行していたとしても安全性は保証されない。暗号に利用することができる乱数の生成方法には大きく分けて TRNG (Truly Random Number Generator)と DRNG (Deterministic Random Number Generator)の 2 つが存在する。TRNG は電源のノイズや意図的なメタステーブルになる回路から、物理的な現象を利用することで乱数を生成する。TRNG 出力の偏りは、製造時のばらつきや温度のような動作環境にも依存することがある。DRNG は、一定のエントロピーを持ったシード値（一般的には TRNG からの出力）を利用し一定の計算アルゴリズム（ブロック暗号やハッシュ関数といった暗号アルゴリズム）を利用することで別の長い乱数系列を導出する。

乱数生成に関する国際的な標準規格として 2 つの有名なものを取り上げる。米国の政府系機関 NIST

(National Institute of Standardization Technology) が発行している文書のうち SP800-90A, SP800-90B に TRNG や DRNG について達成すべき要件や構成例が述べられている [4][5]。また、ドイツの BSI (Bundesamt für Sicherheit in der Informationstechnik) は TRNG/DRNG に対して必要とされる性質に依存したレベルを設けている [6]。また上記の他に、NIST SP800-22 には乱数生成器の出力の妥当性を評価するための統計テスト方法が述べられている [7]。

NIST や BSI の規格を満たす乱数生成器を搭載することが望ましいが、実際の IT システムにおいて乱数生成器の乱数性が十分でないことが原因の脆弱性が度々報告されている。そのため、一定の乱数性が保たれていることをその都度内部的にテストするような機構を設けるか、あるいは第三者評価機関などから認証を得ることが推奨されるであろう。

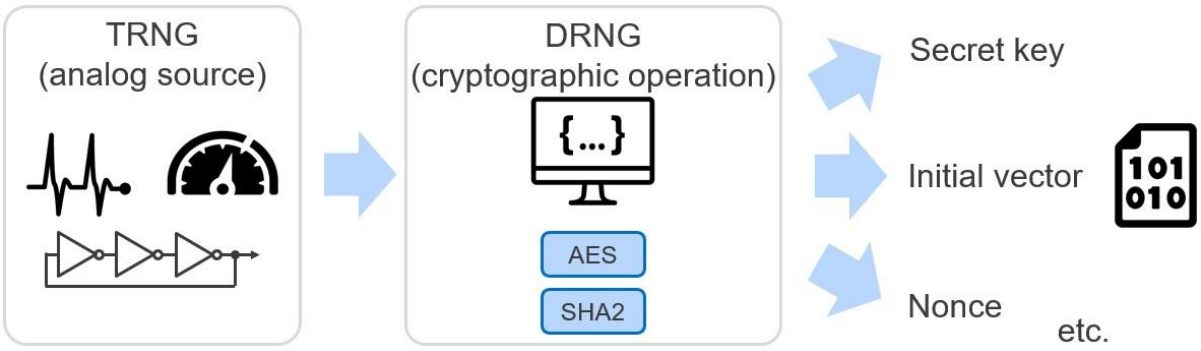

図 4 安全な乱数生成器の構成

- Side-Channel Attack/Physical Attack

セキュリティ機能そのものは、アクセス管理や暗号化、認証などの一定の手順によって担保することができる。一方で機能自体の脆弱性や実装によるバグとは独立に、挙動の観測や直接的な悪意のある信号を挿入することで機密情報を搾取する攻撃は実際に存在する。どの対象に対してどのような攻撃手法が成功するのかを特定し、どのような対策が適切であるかを特定することは難しい。以下によく知られているサイドチャネル攻撃や物理攻撃と一般的な対策手法を紹介する。

一番シンプルな攻撃はタイミング攻撃である。秘密鍵の値に依存した暗号アルゴリズムの処理時間の変化や、CPU のキャッシュに機密情報が残っているかどうかによるアクセス時間の差分を利用することで、機密情報を搾取する。タイミング攻撃はリモート機器からでも観測することができるため、比較的容易な攻撃手法である。そのためには処理時間が一定になるよう暗号処理にダミー演算を追加する、CPU 内のキャッシュはそれぞれの CPU で隔離し、遅延の差が起こらないよう必要な処理が終わったら消去するといった制御が必要となる。

物理的なチップにアクセスすることができる場合は、他にも攻撃経路が存在する。受動的な非侵入型の攻撃として電力消費や電磁波放射を観測する方法がある。攻撃者は一つの秘密鍵に対して様々な入出力データを試行したときの電力や電磁波の遷移から、秘密鍵を得ようと試みる。これらの攻撃に対して一番効果的な対策技術として現在考えられているのはマスキングである。この方法では秘密鍵に乱数を混ぜ合わせ、最終的

な出力が通常動作と同じになるよう特別に改良された暗号アルゴリズムを実行する。

能動的な非侵入型の攻撃としては、Fault injection 攻撃が挙げられる。この攻撃は、チップの外部ピンから接続されているクロックや電圧を対象としてグリッチ信号を挿入することで結果的に意図していない挙動を引き起こす。例えば、この攻撃はハードウェアにおけるセレクタ論理やソフトウェアにおける”IF”構文の分岐処理を攻撃対象とし、本来の判定結果とは逆の後続処理を実行させることを試みることが可能である。例えば、署名検証や Challenge-Response 認証が正しく実行されているにもかかわらず、判定結果が迂回される可能性がある。Fault Injection 攻撃への対策例としては、影響を強く受けるハードウェアモジュールに対して専用の検知回路を載せるか、あるいはハードウェア/ソフトウェアどちらかにより二重チェックを実行することである。

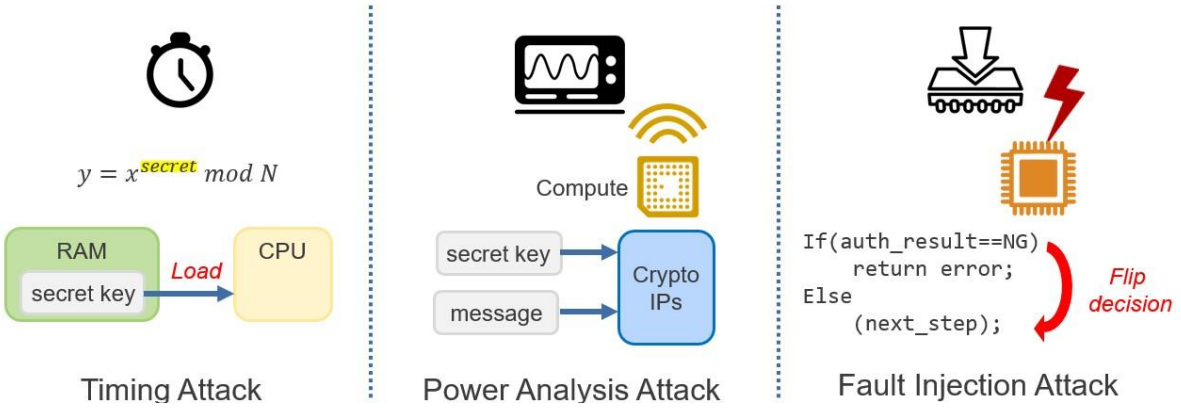

図 5 Side-Channel Attack/Fault Injection Attack

[MCU/SoC との通信に関わるセキュリティ]

- **Secure External Communication**

  自動車内部のソフトウェアのシステムアップデートや、車内エンターテイメントシステム、さらには自動運転のための V2x (Vehicle to Everything)をサポートするためには、5G 回線や Wi-Fi、近距離無線などの車外との通信が必要不可欠になる。そのときの通信路における盗聴や改ざん、なりすましへの攻撃に対処するため、通信路は十分に安全でなければいけない。既存のインターネット向けのセキュリティ対策と同じように、必要に応じて TLS (Transport Layer Security) や VPN (Virtual Private Network) を用いて標準的なセキュア通信路を確立した上で通信を行うことが望ましい。もし DoS (Denial of Service)攻撃や不正パケットが基本的な自動車制御システムを阻害する可能性を考慮し脅威と捉えるのであれば、Firewall や IDS/IPS のような機能を特定のインターフェースに備え、悪意のある通信パケットをフィルタリングする必要も出てくるであろう。

- **Secure On-board Communication**

  車外とのネットワーク通信を十分に保護すれば安全な環境構築が可能になるわけではない。例えば近年の自動車盗難方法の 1 つである CAN Invader は、専用のハッキング装置を物理的に通信 I/F へ有線接続し、ド

アの開錠を指示するコマンドを送ることを試みる。そのような実際の車内の機器とは別の端末から受信したデータが実行されることを防ぐためには、車内ネットワークに対して常に送信者の完全性検証を実行する必要がある。

AUTOSAR (Automotive Open System Architecture) では AES の CMAC モードを車内の通信に適応することを規定しており [3]、ルネサスを含む多くの半導体企業は、自動車向け製品の HSM に AES ハードウェアを提供している。車内ネットワークは静的であり自動車オーナーによって変更されるものではないため、CMAC に用いられる秘密鍵はネットワーク構成をベースにし事前に工場内で書き込んでおくことが一般的である。N 個の通信相手に対しては N 個の秘密鍵を保持する必要があるため、将来的に車内のチップが極めて増加する場合には、事前共有鍵ではなく鍵交換が必要とされる可能性があることが予想される。

On-board communication に求められる安全性のうちのキーポイントが完全性であるとシンプルに述べても、必要とされるスループットや遅延量は使い方によって強く依存する。例えば、ステアリングやブレーキといった走行制御の情報は明確に遅延が極力少なくなるよう設計されていることが必要であるが、データ量としては多くないことは想像しやすい。一方で、自動運転に必要不可欠なカメラからの動画データは圧縮技術が発展しているとはいえ、リアルタイムにデータを流し続けるものであるため十分な帯域とそれを処理するだけの処理能力が完全性を付加する暗号エンジンにも要求される。これらの例から、車両内のチップの中でも利用用途に応じて、十分に処理を賄うことができる様々なタイプの暗号エンジンが搭載されたものが望ましい。

- Debug Port Security

一般的に半導体製品では debug port を設けており、チップ内の様々な資源（不揮発性メモリ、SRAM やレジスタ）にアクセスすることができる。この機能を自動車オーナーが利用する場面は想定されていない。一方で OEM や Tier1、自動車向けソフトウェア等の設計や検証を行う立場では、開発段階の不具合やバグの解析を用いるために必要不可欠である。一方、debug port に誰でもアクセスできる状態のまま出荷されると、簡単に悪意のあるハッカーが内部に格納されている任意の情報にアクセスすることができ、通信ログや秘密鍵といった安全に格納されている情報を盗むことができる。そのため、debug port 自身を出荷前に永久に無効化するか、あるいは ID 認証や Challenge-Response 認証を設けて信頼されたユーザ（設計・製造に関わっている特定の企業のみ）のみがアクセスできるように厳重な保護機構を設けておくことが必要である。

近年の自動車向けのハッキングにおいても、自己診断ポートを経由した車内ネットワークの情報解析やなりすましコマンドの送付などに利用されることがある。そのような攻撃を防ぐためには、非正規の端末からのアクセスができないよう上記のようなアクセス保護メカニズムを備える必要がある。

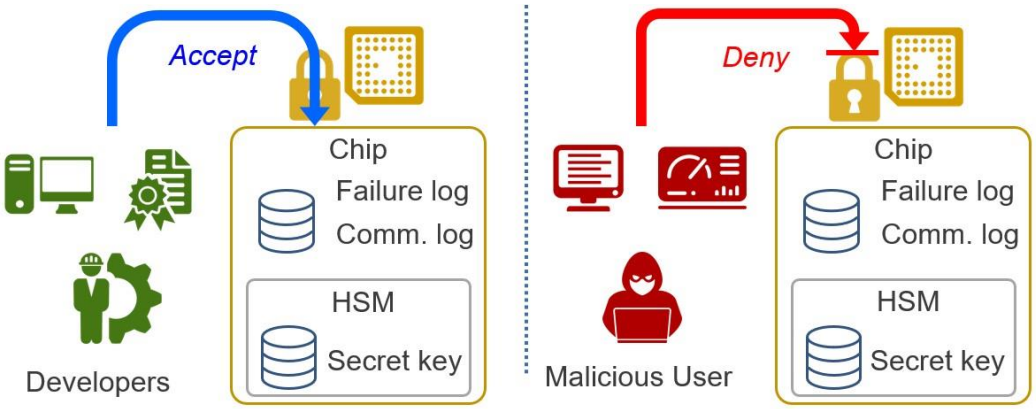

図 6 Debug ポートのセキュリティ

[将来の技術変化に向けたセキュリティ]

- **Crypto Agility**

  暗号アルゴリズムを利用する場合、最適な解読アルゴリズムあるいは全数探索が現実的な時間内で実行されたとしても十分に安全性が確保されることを想定しなければいけない。一方、コンピュータの計算能力は年々増加しているため、一定の期間が経つとより強度の高い鍵長や暗号アルゴリズムへと移行する必要がある。NIST は 112-bit security（2,048-bit の RSA や SHA-224 等）は 2030 年以降に利用しないようにガイドラインを出している。CRYPTREC はさらに細かい指標を表 1 のように示している [8]。もしこれらの鍵長や暗号アルゴリズムを現在のシステムにおいて利用している場合、将来的に 3,072-bit RSA や P-256 曲線を使った楕円曲線暗号、SHA-256 などの 128-bit security を満たす暗号アルゴリズムに移行する計画を準備する必要がある。

  もし暗号アルゴリズムがハードウェア実装されている場合であっても、128-bit security 以上のものを含め複数の安全性レベルをサポートしていれば、ファームウェア更新により利用されるアルゴリズムに切り替えるのは容易である。それよりも注意しなければいけないのは、セキュリティアーキテクチャの中で一部でも弱い安全性のものが含まれていると、その部分が侵入経路の起点となりうることである。そのためセキュリティネットワーク全体において一定水準を満たす暗号が利用できるよう適宜更新されなければいけない。

  自動車社会において、ドライバーが自動車を 10 年以上使い続けることは珍しいことではない。自動車自身の物理的なパーツの経年劣化も明らかに存在するが、セキュリティについてもより高いレベルへの移行や、低い安全性レベルが不用意に選択されないように考慮する必要がある（ダウングレード攻撃）。ハードウェアが FPGA (Field Programmable Logic Array)でない限り、製造された回路を変更することはできない。暗号アルゴリズムをハードウェア実装する際は高いセキュリティレベルを事前にサポートし、世の中の動向に応じてソフトウェアでセキュリティレベルを制御できるようにしておくことが望ましいアプローチである。

表 1 Crypto Agility (from CRYPTREC [8])

		-2030	2031-2040	2041-2050	2051-2060	2061-2070
112-bit security	Encrypt/Sign	Transition	Unavailable	Unavailable	Unavailable	Unavailable
	Decrypt/Verify		Permit			
128-bit security	Encrypt/Sign	Available	Available	Transition	Unavailable	Unavailable
	Decrypt/Verify				Permit	
192-bit security	Encrypt/Sign	Available	Available	Available	Available	Available
	Decrypt/Verify					
256-bit security	Encrypt/Sign	Available	Available	Available	Available	Available
	Decrypt/Verify					

- Post-Quantum security

  近年急速に発展している量子コンピュータに対しての暗号の安全性は、PC やクラウドサーバといった古典コンピュータとは独立に評価される。RSA や楕円曲線暗号といった公開鍵暗号は、十分な計算能力が備わっている量子コンピュータを用いると容易に破ることができる。NIST は耐量子安全性を有する公開鍵暗号の標準化プロジェクトを遂行しており、現時点では公開鍵暗号方式として Kyber、電子署名方式として Dilithium, FALCON, SPHINCS+が選定されている [9]。NIST は将来これらの選定アルゴリズムの最終仕様を決定させた後に、FIPS ドキュメントとして規格化を進めるプランを立てている。そのため、上記の”Crypto Agility”とは独立に、耐量子安全性を考慮して公開鍵暗号方式を置き換えることを計画しておくことが推奨される。より詳細な耐量子暗号に関する記事は[10]を参照のこと。

  共通鍵暗号は、公開鍵暗号と異なり量子コンピュータによってすぐに破られるわけではない。その代わりに本来の古典コンピュータ向けの安全性と比較して安全性レベルが半分に低下する。例えばブロック暗号ならば AES-128 ではなく AES-256 を選択し、ハッシュ関数ならば SHA-256 ではなく SHA-512 を選択されるよう変更することが望ましい。公開鍵暗号よりはこのような変更の影響度は低いと推測される。それでもなお、その中でも高いセキュリティレベルのものを選択する必要はあるため、一定のソフトウェアの更新は必要になるであろう。

ハードウェア対策 v.s. ソフトウェア対策

ソフトウェア対策よりもハードウェア対策を実装するべきかについては、上記すべてのセキュリティ対策技術の導入に対し普遍的な課題であり、本ホワイトペーパーでも別の観点として触れておくことにする。PC やサーバのように汎用的な利用シーンでは、必要最低限の Root-of-Trust となるブートにのみ ハードウェアセキュリティを採用し、その他の一般的なセキュリティについてはそれぞれの使い道によって自由に対策を選択できるようソフトウェアで対処する方が便利である。バグのない高度な機能のプログラムを書くことは非常に難しい（ハードウェア/ソフトウェア両方）が、ハードウェアは開発後には修正できないことから、一般用途の端末ではハードウェア起因のバグの可能性は極力減らす必要がある。万が一ソフトウェア側に脆弱性が見つかったとしても、更新プログラムによってリスクの低減は可能である。

一方で、特定の用途や通信でのみ利用され、かつソフトウェアのみでは求められる性能や遅延が満たされない場合は、ハードウェア実装が必要不可欠である状況が考えられる。例えば AES を 32-bit CPU を用いてソフトウェア実装するとおおよそ 1,500-2,000 サイクル程度かかるが、一般的なハードウェア実装は 11-12 サイクルしかかからない。ブレーキやステアリング制御に関わるデータ転送にある暗号アルゴリズムが適応されることを仮定すると、セキュリティ対策が原因となる致命的な遅延は車両事故の可能性を最小限にするため起きてはいけない。もちろんセキュリティがハードウェアのみで完結するわけではない。これらはソフトウェアで制御されて初めて処理を開始する。そのためソフトウェアが脆弱性を引き起こさないよう動作させなければいけない。

ハードウェアによるセキュリティ対策が確かに特定のシナリオにおいては有益である。ただしハードウェア起因の致命的なバグが見つかる度にチップを回収し再設計するのは現実的ではない。実際、半導体企業は非常に多くの時間を組み合わせも含めた機能の検証に費やしている。一方、予想していない脅威が技術の進歩によって顕在化することを予測するのは不可能である。そのため、どのセキュリティ機能をハードウェア/ソフトウェアで実行するのかは十分に整理した上で一番都合が良い状態の製品をエンドユーザに提供できるよう努めることが望ましい。

まとめ

近年の自動車における技術の変化に伴い、現在様々なセキュリティ技術の導入が自動車半導体製品に求められている。ブート手順から自動車内外との通信、物理的攻撃から次世代に向けた暗号まで、十分なレベルのセキュリティ技術を選定・実装することが望ましい。そして常にセキュリティ意識を維持し、エンドユーザが常に安全な環境を享受できるよう努力することが重要である。

一般的に完璧なセキュリティ対策というのは存在しない。攻撃手法は日々進化しており、今日は安全であったものが明日から安全でなくなる可能性が存在するかもしれない。将来的には自動車の IVI (In-Vehicle Infotainment) を狙ったランサムウェアや自動車バッテリー消費を目的とした暗号資産のマイニングを行うマルウェア等が登場するかもしれない。それでもなお、世の中の技術動向を踏まえて現時点で考えられる既知の脅威から必要とされるセキュリティの分析を行うことが重要である。我々は現在および将来の半導体ソリューションに実装されるセキュリティ技術が全体のセキュリティリスクを低減することに活用されていることを願っている。

[参考資料]

[1] https://www.evita-project.org/

[2] https://www.renesas.com/jp/ja/blogs/introduction-about-secure-boot-automotive-mcu-rh850-and-soc-r-car-achieve-root-trust-1

[3] https://www.autosar.org/fileadmin/standards/R22-11/FO/AUTOSAR_TR_SecureHardwareExtensions.pdf

[4] https://csrc.nist.gov/publications/detail/sp/800-90a/rev-1/final

[5] https://csrc.nist.gov/publications/detail/sp/800-90b/final

[6] https://www.bsi.bund.de/EN/Themen/Unternehmen-und-Organisationen/Informationen-und-Empfehlungen/Kryptografie/Zufallszahlengenerator/zufallszahlengenerator_node.html

[7] https://csrc.nist.gov/publications/detail/sp/800-22/rev-1a/final

[8] https://www.cryptrec.go.jp/report/cryptrec-gl-3004-1.0.pdf

[9] https://csrc.nist.gov/projects/post-quantum-cryptography

[10] https://www.renesas.com/jp/ja/document/whp/latest-trends-post-quantum-cryptography?r=1601456

(Rev.1.0 June 2023)

本社所在地

〒 135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

https://www.renesas.com

お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄りの営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

http://www.renesas.com/contact/