平成２８年度
戦略的イノベーション創造プログラム（自動走行システム）：Ｖ２Ｘ等車外情報の活用にかかるセキュリティ技術の研究・開発プロジェクト

平成 29 年 3 月

一般財団法人　日本自動車研究所

平成 28 年度
戦略的イノベーション創造プログラム（自動走行システム）：
V2X 等車外情報の活用にかかるセキュリティ技術の研究・開発
プロジェクト

― 目 次 ―

第1章　はじめに

1.1　事業全体の目標

我が国では、2020 年までに交通事故死者数を 2,500 人以下とし、世界一安全な道路交通を実現することを国家目標として掲げている。交通事故死者数は、2016 年末まで 16 年連続で減少傾向となっているが、2016 年の交通事故死者数は 3,904 人であり、目標達成まで厳しい状況にある。また、交通事故死者数全体に占める 65 歳以上の高齢者の割合は全体の 50%を超える高い水準で推移しており、その対策が急務となっている。さらに、社会問題の一つである交通渋滞は渋滞損失時間を発生させ、経済機会そのものの損失につながっている。これらの課題に対する究極の解決策として期待されるのが自動走行システムであり、欧米各国と ICT 関連企業などの新規参入事業者を巻き込んだ熾烈な競争が繰り広げられている。

自動走行システムでは、様々なセンサによって収集される自動車そのものの動きや人の動きなどのデータが一つの地図基盤上にリアルタイムで統合され、統合されたこれらのデータ等を自動車が認知し、AI 等によって一歩先を読んで判断、動作を制御する自動走行システムの実現により、交通事故や交通渋滞の低減を価値として提供できる。また、技術の適用範囲を拡大することで公共交通機関の定時運行や、誰もがストレスなく移動できる手段等を新たな価値として提供できる。

自動走行システムの基盤となる高度な地図（ダイナミックマップ）のデータや、地図上にマッピングされる自動車、歩行者、インフラ設備等の情報は、主として車外との通信手段を用いて入手することが想定されている。こうした車外との通信を行う自動車は「つながる車（Connected Vehicle）」と呼ばれ、通信手段としては、車車間や路車間の通信を行う V2X の他、スマートフォンの活用や、PHEV/EV の充電スタンドとの通信等、様々な形がある。また、車両情報の外部とのやり取りという点では、車に備えられた OBD ポートを経由するものも通信に含まれる。こうした通信の中で、リアルタイム性の高い V2X 通信では、交差点における信号情報や、前前方車両の急停止等の直接見えていない他の車両の情報、及び効率的でスムースな交通の流れを実現するための情報等をやり取りして協調することが想定されている。

自動走行システムでは、こうした情報を活用するために、車外から通信によって伝えられる情報を、車内の情報系や制御系に、例えばゲートウェイ等を介して繋がるようになるため、従来の自動車にはなかったサイバーセキュリティへの対応を検討する必要がある。これまでにも、車内のネットワークに OBD ポートから直接接続することで、車両の制御を乗っ取る実験が行われた例があるが、2015 年には車両の外部から自動車メーカが提供する通信サービスを経由して、車両の制御をハッキングした事例が報告されている。自動車を安全に運行させるためには、こうしたサイバー攻撃への対策が重要な課題となっている。

自動走行システムの技術開発にあたって、各国および企業間の競争がその技術レベルの

向上に大きく貢献していることは論を待たないが、そのセキュリティの確保にあたっては多くのユースケースを見据えた対策が必要であり、多方面の知見に基づく共通基盤技術として共通評価技術等の開発、およびそこから得られた知見等の共有化を進めることが望ましい。

しかし、各国および企業により研究・開発されているセキュリティ技術は、個々の自動車が持つ機能・特性および必要性に応じて取捨選択され、適用されることが想定されるが、セキュリティ技術の効果・評価について数値化などで比較することは非常に困難である。結果として、個々の自動車に対するセキュリティ対策の妥当性を客観的に証明することは難しい。

このため共通的な基本システムに対して対策すべきポイントや、そのポイントに対する対策の評価方法・基準などは、企業間の競争により決められるものではなく、それらを共通課題として共通モデルを構築し、知見等を共有化していくことが必要となる。

なお、競争分野である対策技術開発においても、時間経過と共にある程度一般化されていく（業界として常識化していく）技術もある。一方で、一般化機能との差異化部分で新たな競争領域として進化する技術もある。

本事業では、自動走行システムに向けた自動車のシステムアーキテクチャの共通モデルを構築し、テストベッドの構築や評価基準に関する検討・技術開発、V2X 通信における署名検証の簡略化の研究を行うとともに、V2X 通信に関連する海外の技術・動向等の調査を実施した。

1.2　全体スキーム

本事業の全体実施体制を図 1.2-1　事業の全体実施体制図に示す。

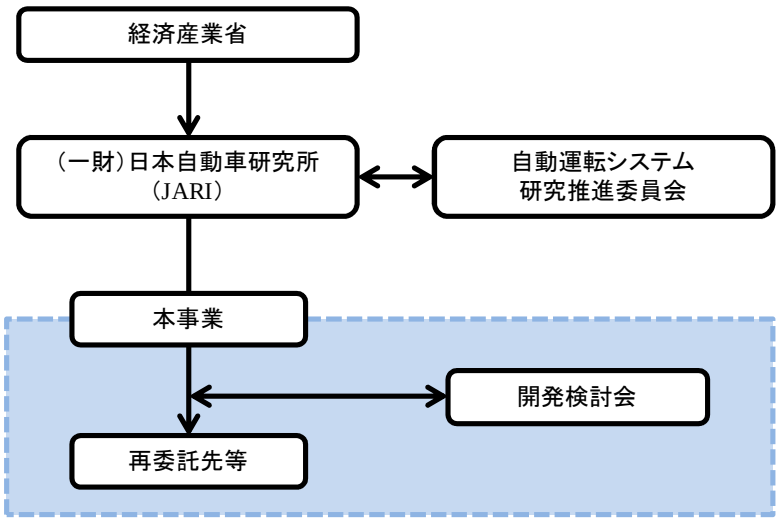


図 1.2-1　事業の全体実施体制図

事業実施にあたっては、成果の最大化、活用促進を図るため、セキュリティ関連の研究等を行っている外部有識者等を含む開発検討会を設置した。また、自動走行に関するプロジェクト全体推進に関して別途設置されている外部有識者や自動車メーカ等からなる自動運転システム研究推進委員会を活用し、助言をいただきながら推進した。

1.3　委員名簿

本事業の研究・開発方針の策定や、事業の推進における課題等を議論する場として、セキュリティ関係の研究を行う機関や、自動車関連団体のメンバーから構成される開発検討会を平成 28 年度に 3 回開催した。委員名簿を表 1.3-1 V2X セキュリティ開発検討会委員名簿に示す。

・第 1 回（平成 28 年 8 月 2 日）:「V2X セキュリティ」プロジェクト概要、計画の説明
・第 2 回（平成 28 年 10 月 31 日）: 研究・開発の進捗報告、課題の議論
・第 3 回（平成 29 年 1 月 30 日）: 研究・開発成果の見通し報告、課題の議論

表 1.3-1　V2X セキュリティ開発検討会　委員名簿

	氏名	組織名　所属／役職
座長	松本　勉	国立大学法人 横浜国立大学大学院 環境情報研究院教授
委員	桑名　利幸	独立行政法人 情報処理推進機構 セキュリティセンター 情報セキュリティ技術ラボラトリー次長
	盛合　志帆	国立研究開発法人 情報通信研究機構 サイバーセキュリティ研究所 セキュリティ基盤研究室 室長
	新国　哲也	独立行政法人　自動車技術総合機構 交通安全環境研究所　主席研究員
	川久保　淳史	一般社団法人 日本自動車工業会 情報セキュリティ WG 副主査
	橋本　寛	一般社団法人 JASPAR 情報セキュリティ技術 WG 主査
	谷口　覚	情報セキュリティ研究開発シナリオ検討 SWG
オブザーバー	一般社団法人　JPCERT コーディネーションセンター	
	経済産業省　製造産業局　自動車課	
	経済産業省　商務情報政策局　サイバーセキュリティ課	
	総務省　総合通信基盤局 電波部 移動通信課（第 2 回より）	
	独立行政法人　情報処理推進機構ソフトウェア高信頼化センター	
	国立研究開発法人 産業技術総合研究所 サイバーフィジカル・セキュリティ研究グループ（第 3 回より）	
	国立研究開発法人 産業技術総合研究所 サイバーフィジカルウェア研究グループ（第 3 回より）	
	e-SYNC 株式会社 （第 3 回より）	
事務局	一般財団法人 日本自動車研究所	

また、本事業を推進するに当たり、自動走行に関する有識者メンバーで構成される自動運転システム研究推進委員会に対して、研究・開発の進捗状況等の報告を実施した。本委員会は平成 28 年度に 4 回開催され、本事業からは、そのうちの第 1 回、第 4 回にて報告を行い、質疑応答および意見交換を実施した。上記推進委員会の委員名簿を表 1.3-2 自動運転システム研究推進委員会委員名簿に示す。

・第 1 回（平成 28 年 9 月 29 日）: プロジェクト概要、計画の説明

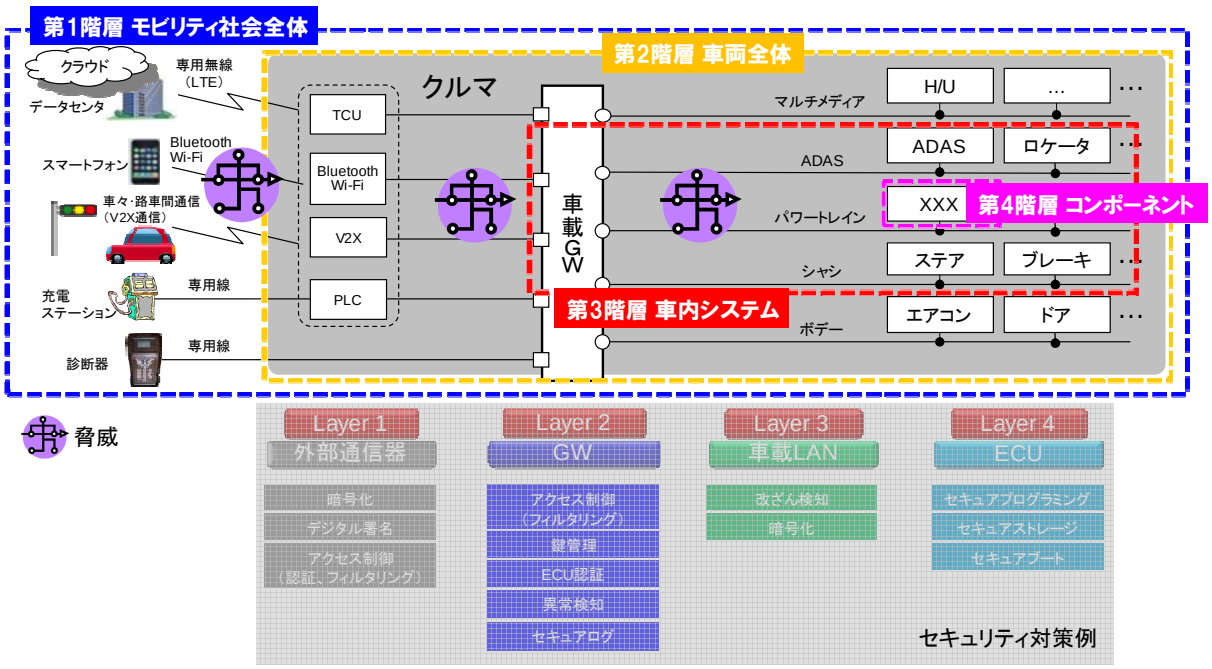
・第 4 回（平成 29 年 2 月 27 日）: 研究・開発成果の見通し報告

表 1.3-2　自動運転システム研究推進委員会　委員名簿

	氏名	組織名　所属／役職
委員長	伊藤　誠	筑波大学システム情報系 教授
委員	石　太郎	早稲田大学環境総合研究センター 参事
委員	大前　学	慶應義塾大学大学院政策・メディア研究科 教授
委員	北﨑　智之	国立研究開発法人 産業技術総合研究所 自動車ヒューマンファクター研究センター 研究センター長
委員	葛巻　清吾	SIP-adus プログラムディレクター （トヨタ自動車株式会社 CSTO 補佐）
委員	菅沼　直樹	金沢大学新学術創成研究機構 准教授
委員	須田　義大	東京大学 生産技術研究所 教授 次世代モビリティ研究センター長
委員	横山　利夫	日本自動車工業会自動運転検討会主査 （株式会社本田技術研究所 上席研究員）
委員	高田　広章	名古屋大学未来社会創造機構　教授
オブザーバー	経済産業省製造産業局自動車課	
オブザーバー	国土交通省自動車局技術政策課	
事務局	一般財団法人日本自動車研究所	

第2章　技術開発項目

本事業において、セキュリティを検討していく上で、自動車として検討すべき対象の階層を図 2-1 の様に分類することとした。第 1 の階層としては、今後繋がる車を含む「モビリティ社会全体」であり、クラウド等を含んでいる。第 2 の階層は、車両外部との通信端末を含む「車両全体」、第 3 の階層は、外部からの情報がゲートウェイを経由して接続される車内のネットワークであり、ここでは「車内システム」と呼ぶ。第 4 の階層は、ECU 等の個別の「コンポーネント」である。



TCU: Telematics Communication Unit, PLC: Power Line Communication,
GW: Gateway, H/U: Head Unit,
ADAS: Advanced Driver Assistance Systems,

図 2-1　本事業の対象の定義

本事業では、主として第 2 階層以下を対象として検討を行った。但し、第 1 階層において発生したセキュリティ上の脅威は、外部通信（V2X、WiFi、スマートフォン等）を経由して第 2 階層以下への入力となるため、これも脅威として扱った。

また、V2X 通信については、車両間の通信（V2V）もあり、自動車向けのセキュリティ技術として検討されているため、本事業の対象に含んでいる。V2X 通信に関しては、国内と海外ではセキュリティ仕様が異なっているため、海外のセキュリティ技術やプロジェクト等の動向についての調査を実施した。

本事業における 4 つのテーマにおける具体的な実施内容については以下の通りである。

2.1 自動走行の共通モデルの構築と、それに基づく脅威分析、セキュリティ要件及び対策の検討（テーマ①）

セキュリティ対策の重要な要素技術の一つが脅威分析である。セキュリティの対策を立案するために、どのような脅威が存在し、その脅威に対する対策（セキュリティ要求）として何が適切であるかを考え、その予想されるリスクを十分低減しているかを明確にするために実施されるのが脅威分析である。

脅威分析は、以下の要素を基に実施するのが一般的である。

・システムアーキテクチャ
・（システムの利用方法に関する）ユースケース（想定利用例）
・脅威分析手法
・リスクアセスメントのための基準
・対策（セキュリティ要求）の作成

これらの技術の開発は自動車メーカの各社が個別に対応するのではなく、自動車産業全体で共有されることで、効率的に、より安全で、セキュアな自動車の開発が可能になる。個社別のアーキテクチャや、ユースケース、脅威分析手法、アセスメントのための基準ではなく、自動車業界で共通に利用できるものとするためには、共通に議論するためのモデルの構築が必要になる。

本事業では、脅威分析のための共通モデルを構築し、そのうえで、産業界において共通に利用できる脅威分析のための共通プラットフォームになるツールの開発を行うものであり、平成 27 年度においては、自動走行の共通モデルとして、共通システムアーキテクチャ、共通ユースケース、脅威分析手法、脅威リスクアセスメント基準の調査を行った。

これらの結果を基に平成 28 年度は以下の方法論の開発、ツールの設計を行った。また、本事業で開発するツールについては、業界で使いやすいものとするために、関連する団体等との意見交換を行って仕様の見直し等の検討を行っていくことにした。

2.1.1 共通システムアーキテクチャ

共通システムアーキテクチャは、脅威分析において、アセットの同定、アタックサーフェイスの同定、脆弱性の分析等に用いられる。そのためには、上記の作業が容易な形のアーキテクチャ図として、産業界の誰でも参照、利用可能なデータとして提供できる形が望ましい。そこで、自動車システムのアーキテクチャの記述を SysML 等の汎用的な記述言語を利用して統一的に記述した。

2.1.2 共通ユースケース

共通ユースケースは、脅威分析手法の中で効率的な利用を可能にする必要がある。そのためには、参照可能なデータとしてデータベースに格納できることや、その参照が効率的に行えること、脅威分析手法の中で容易に利用可能なことが求められる。

平成 27 年度に実施した共通ユースケースの調査結果をデータベース化するためのデータフォーマットの設計を行い、共通脅威分析プラットフォームにおいて利用可能な形にした。脅威分析手法との連携方法については、脅威分析手法の研究において実施した。

2.1.3 脅威分析手法の開発

平成 27 年度に実施した脅威分析手法の調査の結果を基に、新たな脅威分析手法の開発を行った。

本脅威分析手法は、調査の中で明らかになった既存の手法の欠点を補い、セキュリティ脅威の分析において困難である課題（多段攻撃や多重防御の記述と分析、脅威に対する複数の対策があった場合のリスクアセスメント方式等）を解決するものを開発した。

2.1.4 共通脅威リスクアセスメント

脅威リスクアセスメントは、セキュリティ上の脅威がどれだけの被害を与えるか、攻撃がどれだけ容易か、といった点について評価をするものであり、セキュリティ上の脅威に対する対策（セキュリティ要求）がどれだけリスクを軽減しているかを判定するためにも用いられる。

平成 27 年度の調査結果を踏まえて、セキュリティのアセスメント基準（セキュリティメトリックス）の基本原則を明らかにし、国内の標準化検討グループと協議を行った。共通プラットフォームとしては、リスクアセスメントを、アタックツリーで計算する方式案を策定した。

2.1.5 セキュリティ要求の導出

セキュリティ要求は、同定されたセキュリティ上の脅威に対して対抗する手段であり、同定された全ての脅威に対して導出する必要がある。平成 28 年度は、複数の脅威分析手法を用いた多面的な分析によるセキュリティ要求の導出を行える仕組みを策定した。

2.2 車両への攻撃に対する対策の評価手法・認証の調査・研究（テーマ②）

車両に対するサイバーセキュリティ攻撃への対策が現実的な課題になりつつある現状、こうした攻撃に対する対策技術が有効に機能しているかを確認することが重要になっている。確認のための評価を行う環境として、業界が共通して利用できるものを構築することが望ましい。本事業では、テーマ①の共通モデルも踏まえつつ、共通して利用することが出来る評価方法、評価基準について研究を行い、評価環境の構築について検討を行った。また、第三者認証については、海外の動向等も踏まえ、認証のあり方を検討した。

研究・開発にあたっては、以下の 3 点について留意する。

① ICT 分野での事例を収集し、対策すべきポイントを明確にするために情報セキュリティに関する中立的な研究機関の協力が得られるよう取り組む。

② 評価方法・評価基準・評価環境などについては世界的な動向に留意しながら、日本発の提案の盛り込みに貢献できるよう取り組む。

③ 評価方法・評価環境などがコスト面でも事業性を有するものとなるよう留意する。

評価技術に関する研究、認証に関する調査については、以下の 6 項目に分けて実施した。

2.2a　セキュリティ評価用車両模擬システム構築（テーマ②a）

車外連携および車内システムの評価として、本事業で策定する自律型自動走行システムの共通システムアーキテクチャを踏まえた評価環境の構築について検討した。評価対象のシステム仕様としては、車両本来機能とセキュリティ機能とを検討する。一方、車外連携および車内システムの評価の難しさとして、実走行車両に対して評価しようとした場合、実機評価の安全性の確保や、実機評価の再現性の保証などの問題が挙げられる。

そこで、これらの実機評価をシミュレータベースで行うことを検討した。具体的には HILS（Hardware-in-the-Loop Simulator）のように車両挙動等を模擬してシミュレーションを行うことが可能な環境を用いることを検討した。

2.2b　コンポーネント・車内システムにおける評価技術の検討（テーマ②b）

コンポーネントと車内システムに対して、セキュリティ対策の妥当性を定量的に確認可能な評価技術を開発した。

従来は標準的なモデルを定義せずに製品毎に個社でセキュリティの評価を行っていたので、評価基準のばらつきや評価の効率化が課題となっていた。このため、標準的なコンポーネント（標準 ECU）と車内システムによる自動車業界で活用可能な評価方法・評価基準の策定、および前記評価基準の実機検証と効率化のために評価環境の開発が必要である。

本テーマでは、コンポーネントにおける評価対象としての標準的な ECU ならびに評価環境を開発した。また、車内システムにおいては、標準的な ECU 連携機能として鍵管理を選定し、その評価対象を開発するとともに、評価方法・評価基準の策定に必要となる要件について調査した。

(1) コンポーネント（リプログラミング）に対する評価方法・評価技術の検討

評価対象となる標準 ECU として、平成 27 年度はセキュリティ上、重要な機能であるソフトウェア更新機能（リプログラミング）について、標準的なマイコンに搭載可能なソフトウェアを開発した。また、そのソフトウェア上には、攻撃の発生可能性を定量化するためにセキュリティレベルの異なる 4 種類の脆弱性を設定し、実機評価を行った。

平成 28 年度は、さらにセキュリティレベルを上げて評価基準の検討を行った。そのため

に、脆弱性を評価する指標となる「脆弱性水準」の数を追加・拡張し、標準 ECU へ搭載するマイコンのセキュリティ IP の活用を検討した。

車内システムの評価対象としては、自律型自動走行システムを踏まえ、ECU 連携制御における車内ネットワークに対する脅威への評価および車内ネットワークのセキュリティ対策として、ECU のなりすましやメッセージの改ざんを防御する機能を検討した。

(2) 車内システム（鍵管理）に関連する他業界の事例調査（評価方法・評価基準）

コンポーネントの評価方法・評価基準は、平成 27 年度に策定した評価方法・評価基準ドラフトに対しての更新を行った。具体的には、(1)で開発したセキュリティレベルを上げた評価対象を実機評価することで、発生可能性の定量化に関する評価結果を充実させた。

車内システムの評価方法・評価基準の検討に当たっては、鍵管理をテーマに設定してコンポーネントの評価方法・評価基準策定と同様に他業界の事例を調査した。コンポーネント同士が車内ネットワークを介して接続したシステムに関する評価方法・評価基準を調査・抽出し、本調査結果と自動車業界の考え方を踏まえて必要な事項を整理した。

(3) 車内システム（鍵管理）に対する評価方法・評価技術の検討

今後、自動車業界で標準的に使用される可能性がある鍵配布のしくみを搭載した標準 ECU と、鍵を生成して該 ECU へ鍵を配布する模擬ツールを開発した。開発した評価環境を利用して、鍵の生成・配布・保管・利用・廃棄といったシステムの全体像を模擬することができ、技術面だけでなく、管理・運用面の脆弱性についても評価できる環境を構築した。

2.2c　車外連携システム・車両レベルにおける評価技術の検討（テーマ②c）

自動走行を想定した車外連携システムと車両全体に対して、ICT におけるサイバー攻撃の事例などを元にセキュリティ対策を行うべきポイントを抽出し、そこに適用される対策技術の評価方法・基準について研究・開発を行った。

一般的な対策技術について、この評価方法・基準を適用して評価することにより、評価方法・基準としての妥当性の検証を行った。

この結果を基に、自動走行を想定したシステムと自動車全体に対しての評価方法・基準のガイドライン化を目指す。

本テーマではこれらの実験段階として、以下の研究を実施する。

(1) 対策すべきポイントとその評価方法の見直し

平成 27 年度に仮定した対策すべきセキュリティポイント（侵入口）とその評価基準について、昨今のサイバーセキュリティの事例を参考にし、リモートリプログラミングなどのユースケースも想定し、見直しを実施した。

(2) 対策技術の評価方法・基準についての評価環境の構築

(1)の検討内容を検証するための実験システムを構築し、3 つ以上の評価基準を設定し、その机上検討と実験システムでの結果について比較・検討を行った。構築に関しては今回の研究では実際の走行は行わないが、機能的には走行に必要な機能は具備している自動走行のモデル車両を使用して、(1)で設定したモデルの車内ネットワークを構築し、必要と考えられる標準的なセキュリティ対策技術（ハードウェア、ソフトウェア）を搭載し実装した。

実装対象としては、『走る・曲がる・止まる』の 3 カ所を代表する ECU、クラウド連携を想定した外部接続の ECU、車内ネットワークの監視 ECU の 5 カ所を想定した。また、(2)の環境において(1)の評価基準を検証し、その妥当性や課題について検討を行い、その結果に基づき必要な評価基準へのフィードバックと必要であれば再度の検証を実施した。

(3) サイバー攻撃の情報共有検討

サイバー攻撃情報として JPCERT/CC などに提供される善意の情報をどのように扱うかについて、平成 27 年度の調査結果に基づいて、組織に跨がる情報共有の際に必要となる調査・検討を実施した。

2.2d　車内通信プロトコルの仕様に基づく評価方法の検討（テーマ②d）

自動走行に向けて運転支援システム等が高度化することに伴い、これまで以上に車両内の ECU やセンサ等のコンポーネントが協調動作することの重要性が増してくる。この協調動作には車内のネットワークが欠かすことができず、当該ネットワーク上の通信プロトコルがより重要な役割を担い多く利用されると考えられる。従来の車内ネットワークの通信プロトコルに対する研究では特定の通信プロトコルへの攻撃方法のみが検討の対象となっており、自動走行が実現された際に利用されることが予測される通信プロトコルに対しての評価方法・評価基準の調査及び検討はほとんど行われておらず、脆弱性の有無及び評価方法・評価基準は明らかになっていなかった。また、既知の脆弱性や攻撃方法を評価することが可能な評価環境も確立されていなかった。そのため、自動車に対するサイバー攻撃により車内ネットワークの通信プロトコルがどのような影響を受けるかを明らかにすることは、車両のセキュリティ対策に必要である。

本テーマでは、自動走行が実現された際に主流になることが予測される車内ネットワークの通信プロトコルを主な対象とし、通信プロトコルのセキュリティ対策を評価することが可能な評価環境の開発に必要となる要件を検討することとしている。

平成 28 年度は、通信プロトコルにおける既存の脆弱性、および当該脆弱性を利用した攻撃方法に基づく評価方法を検討した。その結果を基に、車内通信プロトコルに対するセキュリティ対策が攻撃の耐性を有しているかを調べるための方法として、シミュレータ上での評価方法が有効であるかの検証を実施した。

(1) 評価方法の検討

脆弱性及び攻撃方法の調査に基づき、以下の観点を踏まえて評価方法の検討を実施した。具体的には、いくつかの攻撃方法を選定し、シミュレータ上で攻撃を行うための詳細な手順について検討を行った。

・通信プロトコル仕様の特徴を利用する。
・通信プロトコルのアプリケーションにおける処理方法の特徴を利用する。
・車内のネットワーク構成やネットワークアクセス方式の特徴を利用する。

(2) シミュレータによる評価方法の有効性検証

攻撃／被攻撃ノードを模擬したテスト環境を市販の車載通信プロトコルシミュレータ上に実現する方法を検討した。検討した実現方法に基づいて、攻撃の影響について確認することで(1)で検討した評価方法の有効性を確認した。

2.2e　実機を用いた評価の実施（テーマ②e）

自動車を構成する実機に対してサイバー攻撃を行うための技術を開発することは、サイバー攻撃から守る技術を開発するために重要である。ここで、攻撃対象として標準 ECU 単体、複数の標準 ECU から構成されるシステム、システムの組み合わせにより構成される車両本体、および車両本体とそれを取り巻くモビリティ社会を想定している。これらはセキュリティを検討する階層と同様に、インタフェースを含めた機能および計算機資源が異なるため、画一的な攻撃手法を確立することはできない。

本テーマでは一般的な組み込み機器に対する攻撃などを参照し、コンポーネントからシステムまで範囲を広げながら様々な攻撃を試みる。また、その成果は評価技術開発へ展開して評価基準作成および評価環境作成へ活用することを想定している。

(1) 標準 ECU に対する攻撃の実施と評価基準の導出

テーマ②b において開発した標準 ECU は、リプログラミングとデバッグの機能、その機能を制御するマイコンに搭載されたセキュリティ IP、マイコン上のソフトウェアとして AUTOSAR にて検討している BSW のセキュリティモジュール、さらに自身以外のコンポー

ネントとの連携などを目的とする車載 LAN のインタフェースをそれぞれ搭載している。まず､それら搭載機能の仕様や特徴から定性的にどのような攻撃が想定可能かを把握した。

平成 27 年度は上記で述べた機能の中でもリプログラミング時におけるツールとの認証に対する攻撃を実施した。平成 28 年度ではこれに加え、標準的に提供されるデバッグポートへの攻撃や保護機能に対する攻撃などを実施した。

平成 28 年度の攻撃側のプロファイル調査は平成 27 年度と同様の基準で実施し、新たな攻撃を実施することで評価基準の精度向上を目指す。

(2) システムに対する攻撃方法検討、攻撃の実施および評価基準の導出

攻撃対象となるシステムに対し、以下の手順で作業を実施した。

- 提供されたシステムに対し、仕様書、公開情報、および予備知識として与えられた情報などを収集し、対象システムの特徴を把握する。
- 対象システムの特徴から攻撃方法を検討する。検討を円滑に進めるために、既知の攻撃や、実施済みであるコンポーネントに対する攻撃から得たノウハウを活用する。
- 提供されたシステムに対し上記検討結果により得られた攻撃を適用する。
- 攻撃実施結果およびそれらに対する考察を行い文章化を行う。

さらにこれらの結果は、評価基準検討の一助として活用する。

2.2f　第三者認証に関する調査・検討（テーマ②f）

平成 27 年度に実施した調査の結果を踏まえ､自動車セキュリティへの第三者認証の妥当性の検討を行った。まず妥当性検討を行うための第三者認証研究会を立上げ、この研究会における議論を元に調査・検討の範囲を明確にし必要な調査を実施した。第三者認証研究会は、自動車業界関係者等に加え、他業界での知見を取り入れるために第三者評価・認証を実施している機関や研究機関等で構成した。

2.3　V2X 通信における署名検証の簡略化の研究（テーマ③）

平成 27 年度「V2X 等車外情報の活用にかかるセキュリティ技術の研究・開発」にて、署名検証の簡略化方式開発に係る机上検討として簡略化方式の調査とともに評価と分析を行い、あるべき簡略化方式の在り方を挙げた。

平成 28 年度においては、この評価結果に基づき打ち出された「署名検証の簡略化方式」について、V2X 通信への適用を前提とした仮想環境を整え、署名検証の簡略化の効果について通信評価を行った。

具体的実施内容は、仮想環境上での通信評価用機能の開発及び構築を実施し、評価用データを生成の上、それを用いて簡略化方式を評価した。その際、署名検証のリアルタイム性確保については、V2X 車載器が受信するメッセージ数として 1 秒あたり 1,000 程度を指

標とした。

2.3.1　メッセージ検証処理と簡略化方式

「簡略化方式の評価」の理解のために、V2X 通信のメッセージ検証処理、及び既存の簡略化方式と本研究による簡略化方式の概要を説明する。

2.3.2　簡略化方式の評価

(1) 評価項目・評価条件の検討

平成 27 年度「V2X 等車外情報の活用にかかるセキュリティ技術の研究・開発」の成果を基に簡略化方式の効果に関する通信評価を行うための評価項目、および、簡略化方式を評価するための評価条件を検討した。

(2) 通信評価機能の開発・構築、及び評価用データの生成

通信評価機能の開発・構築、及び評価用データの生成では、以下の内容を実施した。

- 各項目による評価行うための通信評価機能の要件を整理し、その要件に従い通信評価機能を開発した。
- 評価条件に従って通信評価機能の入力となる評価用データを生成する評価用プログラムを開発した。
- 評価条件に従い、評価用プログラムを用いて評価用データを生成した。

(3) 簡略化方式の通信評価

評価用データを用いて、仮想環境上での通信評価機能により簡略化方式の通信評価を行った。

2.3.3　評価結果の整理及び分析

評価結果を整理し、それを基に簡略化方式の効果について考察した。

2.4　V2X セキュリティに関する海外の仕様や技術動向に関する情報共有（テーマ④）

V2X 通信システムに関しては、適用されるセキュリティ仕様が国内と海外で異なっているため、海外のセキュリティ仕様や、技術・プロジェクトの動向について調査を行った。また、海外動向調査において得られる V2X 通信以外の自動車セキュリティに関する情報も合わせて整理した。

2.4.1　海外の動向調査

V2X セキュリティ技術に関する海外の動向調査として、国際会議やフォーラム等に参加して調査を行った。

技術動向の調査としては、平成 27 年度に調査により抽出された国際会議等のリストに基づき、現地調査の対象を選定した。

2.4.2　情報共有の仕組み運営

2.4.1 で調査した結果については、平成 27 年度に立ち上げた情報共有のメンバとの共有を図った。

第3章　事業内容

3.1　V2X 等車外通信の活用にかかるセキュリティ技術の研究・開発（テーマ①）

本テーマは、自動車業界における脅威分析のための共通プラットフォームとして利用が可能な脅威分析ツール（以下、脅威分析共通プラットフォーム）の開発を目指している。

3.1.1　基本構想

「脅威分析共通プラットフォーム」は以下の方針を元に構想としてまとめ、方法論の開発、ツールの基本仕様の開発を行うこととした。

① 脅威分析の統合支援ツールを提供
単一の手法、方法論だけでは無く、様々な手法、方法論を統合可能なツール環境の提供
② セキュア開発プロセスに基づいた開発工程の支援
機能（技術）サイバーセキュリティコンセプト（Functional（Technical） Cybersecurity Concept）開発の統合的支援
③ 3～5 年後に必要になる脅威分析技術の提供
- 多層防御（Defense in depth）戦略
  多層防御戦略のモデル化、それに基づく脅威分析と対抗策の同定とリスクアセスメントの提供
- 外部環境を含んだシステム全体の脅威分析
  V2V、V2I において侵入経路として考えられる、他システム（路側機、他自動車、他）等を含んだアーキテクチャモデルに基づいた脅威分析手法の開発
- 外部脅威データベースのデータの参照
  既に存在する脅威 DB（例：NVD、CWE）、Auto ISAC 等の脆弱性、脅威情報を参照できる外部 I/F の提供
- セキュリティ上の複数のリスクアセスメント方式の支援
  複数のアセスメント方式（例：CVSS、CC-CEM）が混在する環境でのアセスメント方式の提供

これらの基本構想の内容を以下に説明する。

(1) 脅威分析の統合支援ツールを提供

この方針は、単一の分析手法だけでは脅威分析をするのには不十分であることに起因している。脅威分析を実施するためには、分析の対象となるシステムに関する情報、複数の脅威分析手法による多面的な分析、脅威に対する対抗策の導出、同定されたリスクのアセスメント基準とその評価方法が必要になる。例えば、脅威分析に用いられているアタックツリーは非常に一般的な脅威分析手法であるが、アタックツリーだけでは何が攻撃の対象

（保護資産）なのか、どこから攻撃が行われるか（アーキテクチャ上で同定している侵入経路、システム固有の脆弱性）、同定された脅威からどのような対抗策（セキュリティ要求）が必要かの分析は出来ない。これらの課題を解決するためには、様々な目的により利用される異なる分析手法を統合し、成果物の作成を支援するツールが必要になる。

(2) セキュア開発プロセスに基づいた開発工程の支援

この方針の意図するところは、自動車システムを開発するに際して、どのような開発プロセスを想定したツールなのかにより、その実用性が影響を受ける点である。本ツールの目的は、自動車産業において利用されると予想されるセキュア開発プロセスを支援することである。図 3.1.1-1 は、自動車システムのサイバーセキュリティ規格である J3061[1]のコンセプトフェーズにおけるプロセスである。本プロセスは、自動車の機能安全規格である ISO 26262[2]との親和性も高く、本ツールは基本的にはこのようなプロセスの支援を想定している。

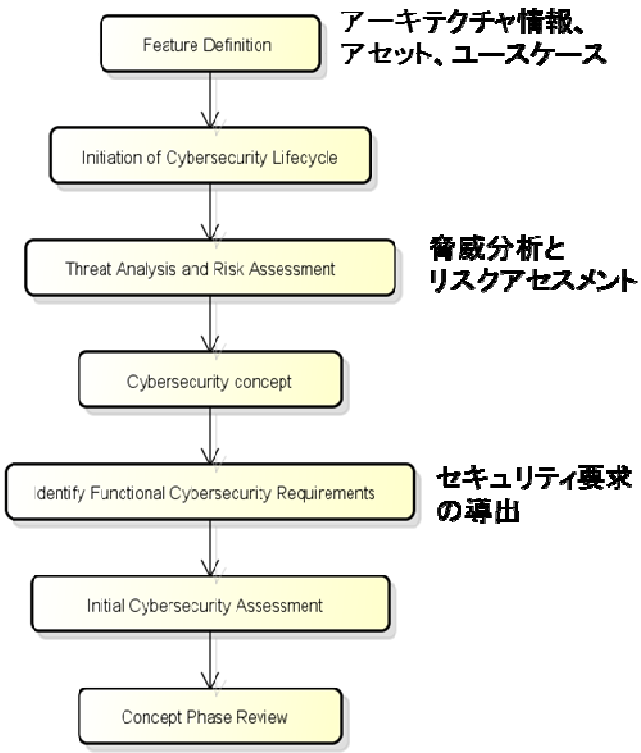


図 3.1.1-1　J3061 による機能サイバーセキュリティコンセプト・フェーズにおけるプロセス

(3) 3～5 年後に必要になる脅威分析技術の提供

3～5 年後には自動走行やコネクティッドカーの利用が予想され、必要な脅威分析技術は、下記の図 3.1.1-2 で示されるような、交通システムレベル、車両レベル、ECU レベルなど様々なレベルにおいて利用が可能なものと考えられる。

図 3.1.1-2　想定されるシステムのレベル

今後、コネクティッドカーが広く利用される場合、V2V（Vehicle to Vehicle）、V2I（Vehicle to Infrastructure）など様々な対象との間でデータ通信が行われる。このような環境においては、自動車本体への直接攻撃も可能であるが、自動車を含む交通システムを構成する様々な装置、システム経由の攻撃が想定される。すなわち、攻撃者は攻撃対象に対して、他のシステムを踏み台にした多段攻撃を行うことが通常の攻撃方法と考えられる。それに対して、防御側は多層防御を実施することになる。

このような交通システムのモデル化、交通システムモデルにおける多段攻撃と多層防御の分析とモデル化、モデル化した攻撃と防御に関連するリスクの評価が必要になると考えられる。

そこで、システムの個々のレベルに対しては、図 3.1.1-3 に示すシステムレベルの脅威分析と、車両・ECU レベルでは機能・技術サイバーセキュリティコンセプト開発という二つの軸に対して、開発方法論とその支援ツールを提供することを目的とした。

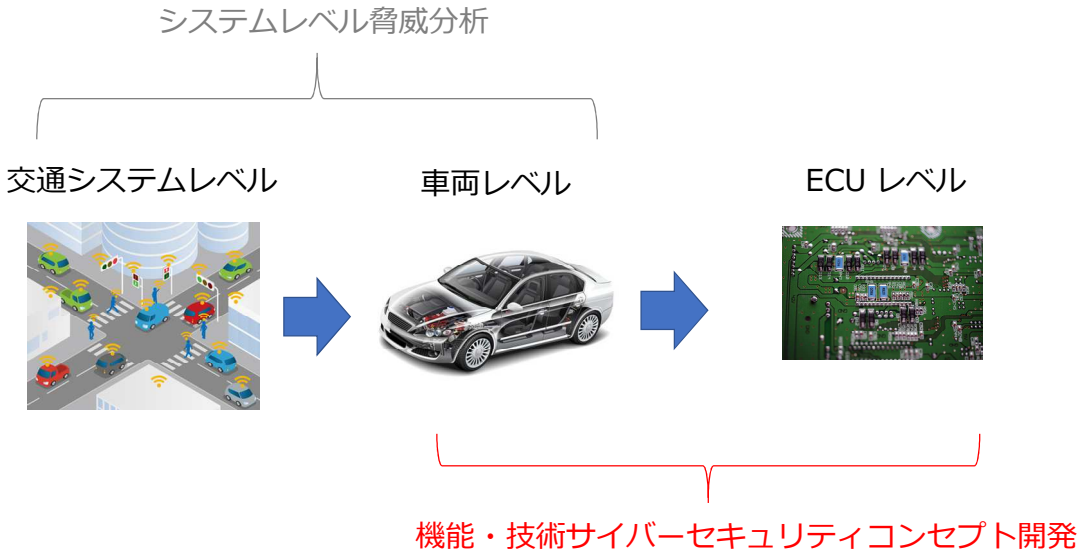


図 3.1.1-3　二つのレベルにおける脅威分析支援

外部脅威データベースのデータの参照では、既知の脅威・脆弱性情報を脅威分析の中で参照可能な外部 I/F を提供することを予定している。外部脅威データベースの例として、オープン SSL の脆弱性として著名なハートブリードは、脅威データベースである CWE

（Common Weakness Enumeration）[3]において CVE-2014-0160 として登録されている。

セキュリティ上の脅威・脆弱性に対しては、すでに情報システム関連においては、公共で利用可能な脅威・脆弱性情報のデータベース（例：NVD（National Vulnerability Database）[4]）が存在する。さらに脅威・脆弱性情報は、各産業固有のセキュリティ対応組織（CERT (Computer Emergency Response Team)）により提供されている。これらは国別の機関としては、日本では JPCERT[5]などがあり、産業界別にはプロセス制御関連では ICS-CERT（ICS は Industrial Control Systems の略）[6]などがある。自動車業界においては、Auto-ISAC（Information Sharing and Analysis Center）[7]が発足しており、今後、車載システムに関する様々な脅威・脆弱性情報が提供される。

そこで、本脅威分析共通プラットフォームは、外部の脅威や脆弱性情報をツールの中で利用することが可能なインタフェイスを準備することで、最新の情報を元に、効率的で共有可能な脅威分析の支援を行うことを目指している。

セキュリティ上の脅威が同定されると、そのリスクのアセスメントを実施することで、どれだけのリスクがあるかの評価が行われる。その評価基準（ここではセキュリティメトリックスと呼ぶ）には様々なものが提案されているが、現時点では自動車システムに関するセキュリティメトリックスは決まっていない。また、将来的には、様々なセキュリティメトリックスを併用して利用することも考えられる。このような状況において、ある特定のセキュリティメトリックスだけではなく、様々なセキュリティメトリックスが利用可能でなければならない。そこで、本脅威分析共通プラットフォームにおいては、様々なセキュリティメトリックスの利用とその計算方式を提供することを目指している。

以上のような構想をもとに開発された様々な手法とシステムレベルとの対応を示したものが以下の図 3.1.1-4 である。

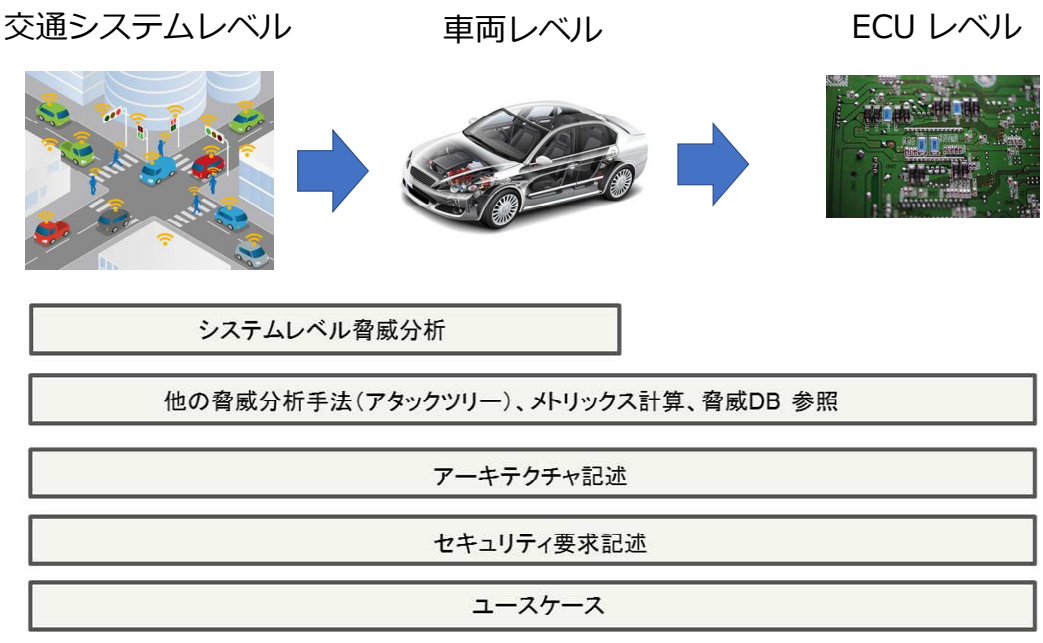


図 3.1.1-4　システムレベルと対応

3.1.2　支援ツールの概要

本節において、脅威分析共通プラットフォームのツールとしての構成、開発に利用するソフトウェアについて述べる。基本的な部分は SysML（The Systems Modeling Language）[8]のセキュリティ拡張として開発することを予定している。どのような記述言語とツールをベースに開発するかについては、アーキテクチャ記述言語、開発支援ツール・プラットフォーム、言語拡張方式の調査を行い、最終的には以下の理由で SysML を採用した。

【SysML を採用した理由】

・セキュリティ上の拡張が非常に容易（元々提供されている拡張機能を利用）
・プラグイン（アドイン）として開発することで、開発コストが抑えられる（API が提供されている）
・SysML を支援している商業用ツールが多い
・それらのツールは、安価に利用が可能
・モデルベース開発において利用されており、従来の開発プロセスとの親和性が高い

記述言語の候補は、SysML 以外に SAE International が開発したアーキテクチャ記述言語 AADL（Architecture Analysis and Design Language）[9]があった。しかし、調査の結果、拡張の難しさ、拡張にかかると予想される工数の多さ、利用コミュニティの違い（主に利用されているのが航空機業界）、知名度（一般的な利用はほぼ無い）といった理由により選択肢から外れた。

開発プラットフォームの中でもモデリング言語開発用の汎用プラットフォームとしては、ドメインスペシフィックモデリングツールである MetaEdit+や汎用的なモデリングプラットフォームである Papyrus Modeling Environment の調査・評価をした。前者に対しては、配布に必要なライセンス料、記述言語を最初から開発することによる開発コストの高さ、実際にどこまで仕様が実現できるかが不明な点から選択しなかった。後者に対しては、様々な拡張が可能なものであるが、日本における利用層の少なさ、開発コストの高さから選択しなかった。

セキュリティに関する記述言語に対する調査として、セキュリティ上の拡張を行う際の容易さ、意図された拡張の実現方式、支援ツールにおける実装の際の容易さについての調査を実施した。そこで、著名なものとして UML（Unified Modeling Language）のセキュリティ拡張記法である UMLsec [10]と SecureUML[11]の調査を行った。 UMLsec は汎用的なセキュリティ記述言語であり、SecureUML はロールベースアクセスコントロール実装のための記述言語である。これらは UML 上の拡張機能であるステレオタイプを利用してセキュリティ固有の概念を実現しており、拡張方式・支援ツールによる実装の容易さが格段に優れていることが判明した。SysML は UML のシステムズエンジニアリングのための拡張であり、UML の拡張機能をそのまま利用が可能である。このような理由により、SysML を採用することとした。

以下に SysML と脅威分析共通プラットフォームで提供する分析手法、記法との関連を示す。

【記法との関連】

①アーキテクチャ記述
　SysML のブロック図のセキュリティ拡張
②共通ユースケース
　市販の DB ソフトを利用予定
③脅威分析手法（システムレベル脅威分析）
　新たにプラグイン（アドイン）として開発
④セキュリティ要求
　SysML の要求図のセキュリティ拡張
⑤メトリックス計算
　新たにプラグインとして開発

アタックツリー記述に関しては、新規に開発するのでは無くすでに開発されたツールを利用することを予定している。

3.1.3　アーキテクチャ記述

自動車システムのアーキテクチャ記述のための汎用的な記述言語（図表現）として SysML におけるブロック図を採用する。

(1) アーキテクチャ記述に対する基本要件

セキュリティ分析のために必要なアーキテクチャ図としての基本要件には以下が必要と考えた。SA-基本要件 1）は、工数増加を抑え整合性保持を確保するための統一的な図表現を要求しているものであり、SA-基本要件 2）~5）は記述に対する要求である。

① SA-基本要件 1）
　通常のシステムアーキテクチャ記述を利用してセキュリティ分析が出来ること
② SA-基本要件 2）
　攻撃の対象（防御から見ると、保護の対象）である保護資産が記述できること
③ SA-基本要件 3）
　攻撃のための侵入経路（アタックサーフェイス）が記述できること
④ SA-基本要件 4）
　アーキテクチャのどの部分をセキュアにしたいか（セキュリティゾーン）が記述できること
⑤ SA-基本要件 5）
　アーキテクチャの構成要素のどの部分が信頼できるかどうか（信頼性境界）を記述できること

注：ここで SA は Security Architecture の略。

(2) アーキテクチャ構成要素

上記以外の基本要件について説明する。

① 保護資産

システムアーキテクチャの要素においてセキュリティ上の脅威から防御したい対象を記述することができる。ブロック（<<block>>）に対して、asset 属性を定義し、その属性値を定義することで、どのような保護資産であるかの記述が可能とする。

② アタックサーフェイス

対象システムに対する攻撃は、外部から行われる場合どこが侵入経路になるかを分析することが重要である。攻撃は弱い部分を攻めるのが常道であり、どの表面が攻撃に晒されるかを分析するところからこの名前が付けられている。マイクロソフト社のセキュア開発プロセス方法論 SDL（Security Development Lifecycle）では信頼性境界と呼ばれているものである。ブロック（<<block>>）に対して、新たに<<attack surface>> ステレオタイプを付けることで示す。

③ セキュリティゾーン

防御の対象となるシステム構成の範囲を明確にするために用いられる。本概念は J3061 における security perimeter と同等の概念である。

セキュリティゾーンと物理的なシステム構成は重なる場合もある。しかし、物理的なシステム構成が必ずしもセキュリティゾーンである必要は無い。ブロック（<<block>>）に対して、新たに<<security zone>> ステレオタイプ付けることで示す。

④ 信頼性境界（trust boundary）

今後、V2I/V2V など様々な対象と通信で結ばれる場合に、通信先の信頼の度合いによりセキュリティポリシーを変更する必要性が出てくると想定される。そのような場合に、信頼度の異なる対象を区別するために信頼性境界を導入する。さらに、高度な情報管理においては、機密性の保持を行うためにデータの流れを片方向にする機構を導入する場合がある。またそのための装置（例：data diode）が開発されている。これは、信頼度が高い方から低い方へデータを流さない、といった原理に基づいて利用されている。

注意する必要がある点は、SDL における信頼性境界はアタックサーフェイスを意味する点である。現時点では、信頼性境界は脅威モデルを構成する要素として入れるが、詳細な利用方法については継続して検討することが必要である。ブロック（<<block>>）に対して、新たに<<trust boundary>> ステレオタイプ付けることで示す。

アーキテクチャ記述に関しては、セキュリティ拡張のための基本要件の整理、SysML のブロック図を利用したアーキテクチャ記述に関するセキュリティ拡張、他分析手法との連携について設計を行った。

3.1.4 共通ユースケース

脅威分析を実施するためには、対象システムがどのように利用されるかのユースケースの明確化が必要である。昨年度に開発された共通ユースケース（CU1～CU45 として分類）を利用するために、ユースケースを格納するユースケースデータベースを構築する必要があり、ユースケース利用に関する基本要件の分析、実際にデータベース管理システム上に構築するために必要なユースケースデータベースの論理設計を実施した。

(1) ユースケース利用に関する基本要件

ユースケースを脅威分析に利用する際に重要な基本要件を以下のように分析した。

① UC-基本要件 1）
脅威分析において、どのユースケースを参照して分析しているかが明示できること。
② UC-基本要件 2）
脅威分析に関連するユースケースをユースケースデータベースから検索できること
③ UC-基本要件 3）　ユースケースデータベースの管理が出来ること
④ UC-基本要件 3-1）
必要に応じてユースケースをユースケースデータベースから削除できること
⑤ UC-基本要件 3-2）
必要に応じてユースケースデータベースのユースケースを変更できること
⑥ UC-基本要件 3-2-1）
新規にユースケースをユースケースデータベースに追加できること
⑦ UC-基本要件 3-2-2）
ユースケースデータベースのユースケースを改訂できること

注：ここで UC は Use Case の略。

(2) 論理設計

共通ユースケースは非常に単純な構造をしているので、基本的には図 3.1.4-1 のような ER 図として定義することが出来る。

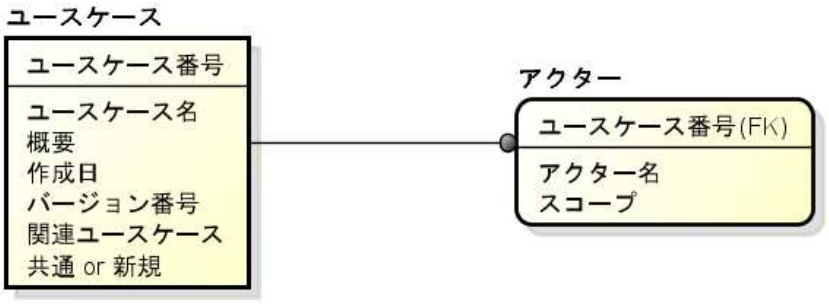


図 3.1.4-1　ユースケースデータベースの論理モデル

ユースケースは番号で参照され、その属性としては、ユースケース名、概要、作成日、バージョン番号、関連ユースケース（元になったユースケース)、共通ユースケースモデルとして提供されたか、新たに追加されたかどうかのフラグ、により構成されている。

ユースケースとは別にアクターの改訂も可能になっている。アクターとその利用範囲であるスコープの定義が可能である。本論理モデルにおける基本的なアクター（アクター名としての種類）は以下のものである。

【アクター名】

・車両
・RSU
・他車線の車両
・隊列のリーダーになりえる車両（PLV）
・追従車両と成り得る車両（PFV）
・隊列走行のバックオフィス（BOA）
・隊列のリーダー車両（LV）
・追従車両（FV）
・隊列外の車両（OV）
・緊急車両
・持込み機器
・ワイヤレスキー

それらに対して、アクターのスコープは以下のものである。

【アクターのスコープ】

・V2V
・V2I
・隊列走行
・緊急時
・降車、乗車

(3) 基本的な利用方法

ユースケースは脅威分析の様々な段階で利用される。例えば、「通信妨害」という脅威を考えた場合、「緊急車両通行時」に対する妨害が発生するかどうか、といったことを検討するために用いる場合は、共通ユースケースデータベースから、フリーワードの検索「緊急車両」を検索することで、以下のユースケースを参照することが可能である。

・CU-37：緊急車両を優先走行させる
・CU-38：緊急車両の走行支援

共通ユースケースを利用して脅威分析をする場合に、これだけでは不十分な場合も予想される。また、各社において既に利用されているユースケースを保有している場合もある。このような場合には、共通ユースケースをカスタマイズする必要がある。ここでは、簡単にどのようにユースケースをカスタマイズするかについて示す。共通ユースケースは以下の分類により作成されており、カスタマイズする場合は表 3.1.4-1 の分類に従い新たにユースケースを割り当てることで、共通ユースケースの利用が可能である。

表 3.1.4-1　ユースケースの分類

【ユースケースの分類】

- 通常運転
  - 共通事項
    - 交通法規の遵守
    - 道路環境への対応
  - 交差点以外の車線のある道路
    - 前方の検知
    - 車線をまたぐ検知
    - 隊列走行
  - 交差点（信号有）
    - 直進
    - 左折
    - 二輪車などの警告
  - 交差点（信号無）
  - その他
- 駐車
- 緊急車両通行時
- 停止時
- その他

共通ユースケースに関しては、ユースケースデータベースの機能に関する基本的な要求獲得を行い、必要な機能要件を整理した。検索機能としては、フリーワードによる検索機能を基本とすることとした。今後データベースシステムとして実装するための論理モデルを構築し、利用者によるユースケースデータベースのカスタマイズ方式を決定した。

共通ユースケース自体は、様々な研究プロジェクトにおいて開発されたユースケースを共通化して開発されたものであり、その分抽象度が高いものである。そのため、より詳細なユースケース記述が必要な場合には､元となったユースケースを参照する必要があるが、それらの利用に関しては著作権の対応が課題となる。

3.1.5 脅威分析手法

脅威分析手法に関しては、システムレベルの脅威分析手法と、脅威分析とセキュリティ要求を有機的に連携させる統合的な枠組みの開発を行った。

(1) 脅威分析手法に対する基本要件

以下が脅威分析手法に対する基本的要件である。

① TA-基本要件 1）

脅威分析が交通システムを含む全体のシステムレベル、車両レベル、ECU レベルなど様々なレベル・抽象度で実施できること（多段攻撃の記述が可能なこと）

② TA-基本要件 2）

基本要求 1 に示された各レベルにおいて同定された脅威に対して、利用可能なリスクアセスメント方式と連携ができること

③ TA-基本要件 3）

基本要件 1 に示された各レベルで実施された脅威分析において同定された脅威に対する対抗策（セキュリティ要求）と有機的に関連することができること（多層防御の記述が可能なこと）

④ TA-基本要件 4）

脅威分析として、機能セキュリティコンセプトの開発プロセス、成果物を支援できること

注：ここで TA は Threat Analysis の略。

開発されたシステムレベルの脅威分析手法においては、多段攻撃と多段防御の記述と分析が可能であり、当初目標としていた手法の開発をすることが出来た。

(2) システムレベル脅威分析手法

ここでは、システムレベル脅威分析手法を示す。図 3.1.5-1 は、ネットワークが「オフィス系ネットワーク」と「制御系ネットワーク」に分割されたシステムを表しており、それ

に対して、外部から攻撃「T1：標的型攻撃」が加えられた場合の分析を意味している。

オフィス系ネットワークに侵入するためには、ネットワーク内での攻撃「T2：乗っ取り」が実行され、成功すると「T3：情報収集」（例えば ID やパスワードの入手）が行われる。そして、乗っ取りが行われた PC を踏み台にして、「制御系ネットワーク」にいる制御システムに対する攻撃「T4：制御ソフトの書き換え」が実行される。このように、本脅威分析手法では、多段攻撃を自然にモデル化することが可能である。

さらに、各攻撃に対する防御策と検知策の記述も可能であり、この例では攻撃 T2 に対する対抗策として「M1：マルウェアの感染を防ぐ」があり、攻撃 T3 に対して対抗策「M2：不審なプロセスを動作させない」とその検知策「D1：不審な通信トラフィックの監視」が定義されており、自然な形で多層防御のモデル化が出来ている。

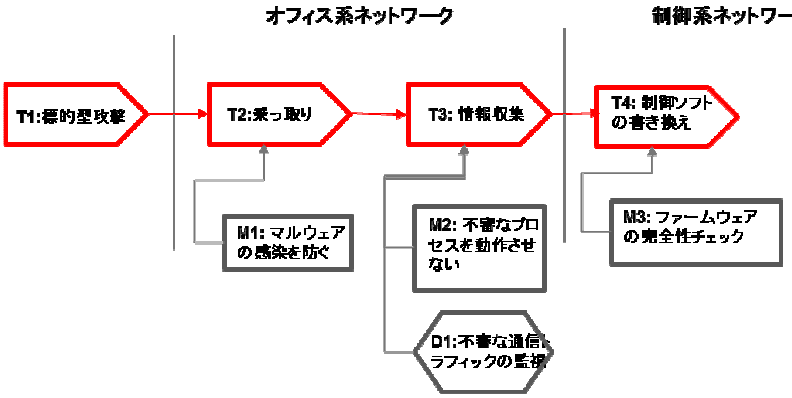


図 3.1.5-1　システムレベル脅威分析手法

さらに、これらの分析された攻撃と防御策は、それぞれアタックツリーとセキュリティ要求図を用いてより詳細に分析することが可能である（図 3.1.5-2）。

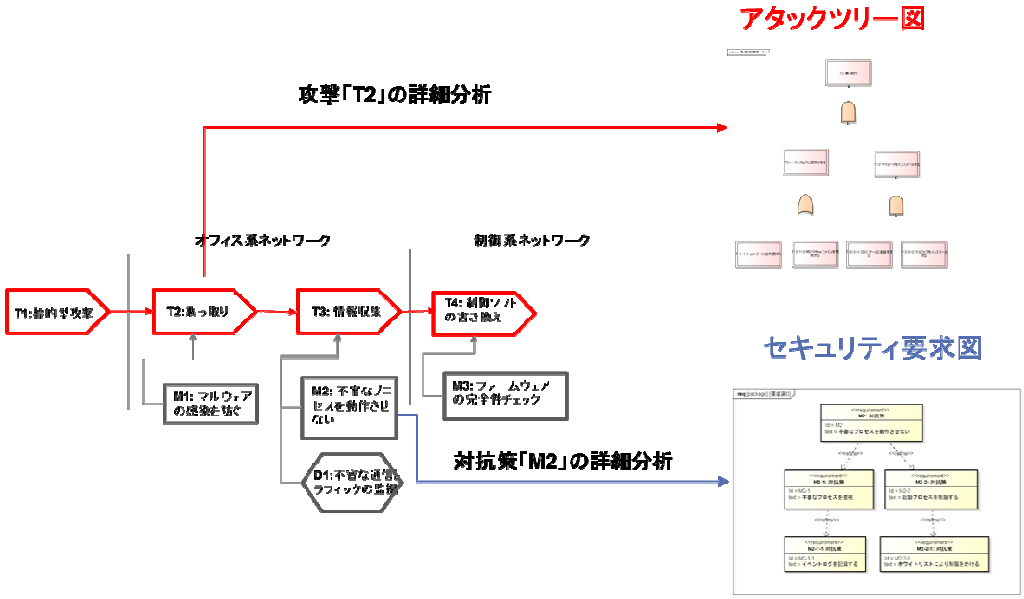
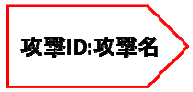
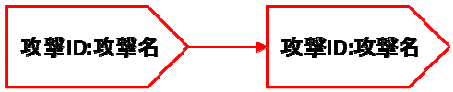
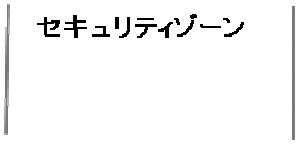
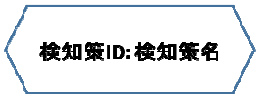
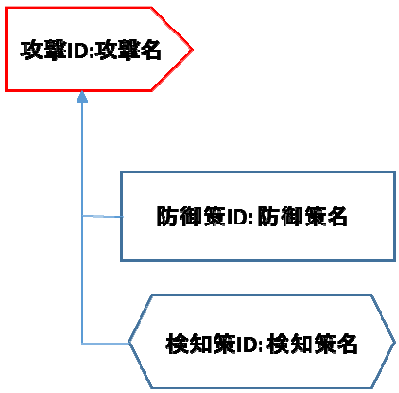


図 3.1.5-2　システムレベル脅威分析の詳細分析

本手法は、システム全体を概観した形での大域的な分析を元に、より詳細なシステムレベルでの分析を自然に組み合わせている点と、攻撃とセキュリティ要求の分析が同時に可能である、という特徴を持っている。

表 3.1.5-1 に、本脅威分析手法の記法を説明する。

表 3.1.5-1　システムレベル脅威分析記法の説明

記法	説明
	攻撃の記述の最少単位。 攻撃 ID（攻撃の ID）と、攻撃の記述（攻撃名）が可能。
	連続した攻撃の記述が可能。
	「3.1.2 アーキテクチャ記述」における、セキュリティゾーンを意味する。
	攻撃に対する防御策の最小単位。 防御策 ID（防御策の ID）と、防御策の記述（防御策名）が可能。
	攻撃に対する検知策の最小単位。 検知策 ID（検知策の ID）と、検知策の記述（検知策名）が可能。
	攻撃に対して、その防御策と検知策の記述することが出来る。防御策と検知策の双方を必ず記述する必要は無い。 この三者は、必ず一つのセキュリティゾーン内に記述され、他のセキュリティゾーンにまたがることは無い。

これらの記述方式が脅威分析共通プラットフォーム上でどのように他の記述方式と連携しているかは、図 3.1.5-3 に示す UML のクラス図で記述されたモデルにおいて規定される。

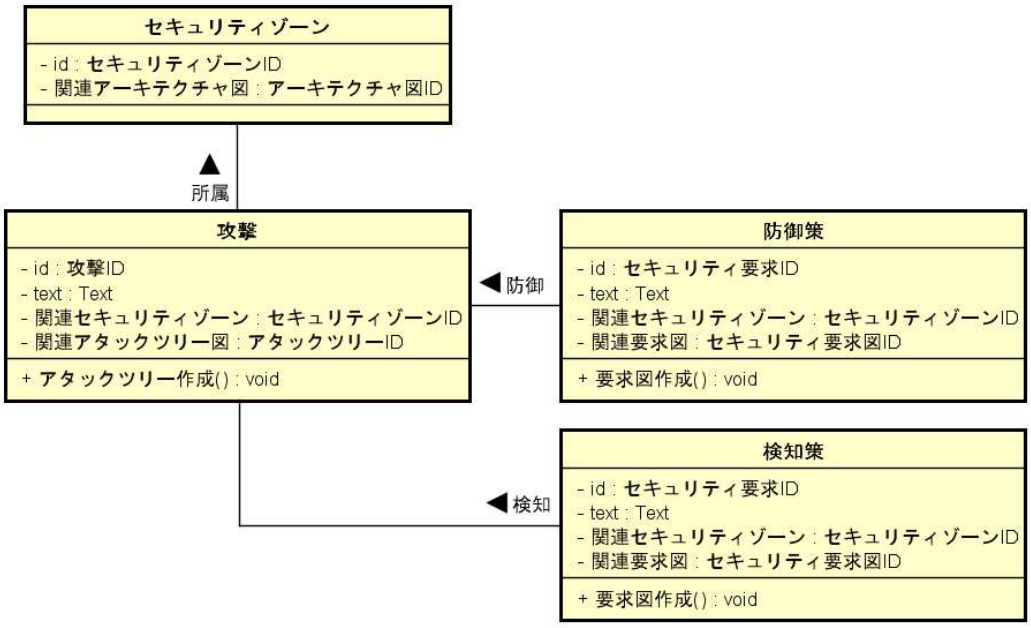


図 3.1.5-3　システムレベル脅威分析記法のモデル

例えば、セキュリティゾーンがアーキテクチャ図において定義されている場合には、関連アーキテクチャ図という属性の値としてそのアーキテクチャ図 ID により連携が行われる。攻撃に対しては、関連アタックツリー図により連携が行われ、防御策・対抗策については、関連要求図によりセキュリティ要求図との連携が成立する。さらにシステムレベル脅威分析の結果から､アタックツリーとセキュリティ要求図を生成する機能を示している。この機能により、システムレベル脅威分析作業においてアタックツリーとセキュリティ要求図を呼び出すことが出来る。

今回開発したシステムレベルの脅威分析手法と他の分析手法との連携方式は、新規性が非常に高いものである。さらに、交通システムレベルにおける多段攻撃と多層防御の分析を支援することを可能にするものである。本手法に関する今後の課題としては、このような手法とリスクアセスメントをどのように関連付けるかである。本課題については、3.1.6 節において説明を行う。

最後に、図 3.1.5-2 において示した、アタックツリーによる詳細分析の結果とセキュリティ要求の詳細化において、どのように連携するかを図 3.1.5-4 及び図 3.1.5-5 に示す。

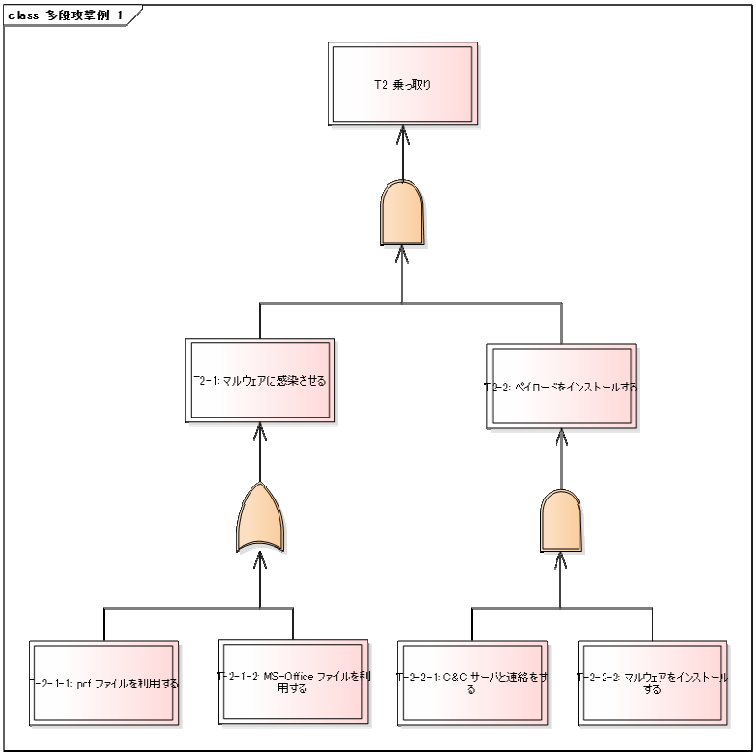


図 3.1.5-4　アタックツリーによる詳細分析

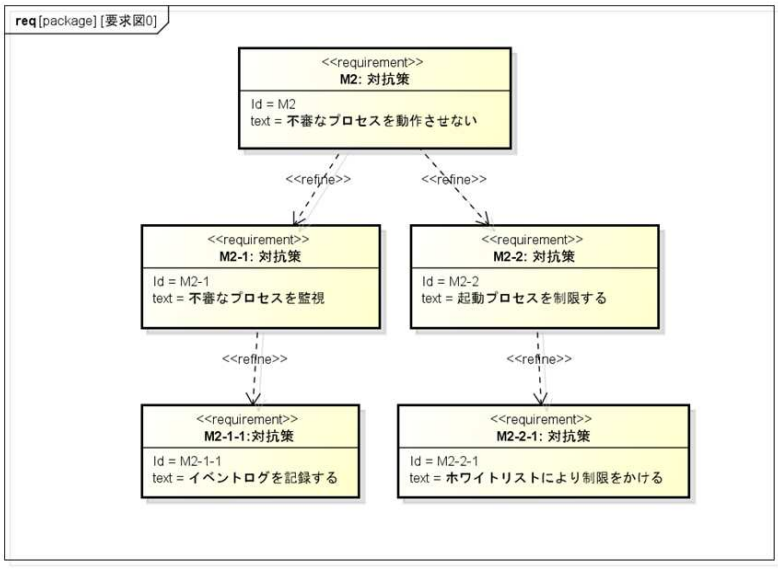


図 3.1.5-5　セキュリティ要求の詳細分析

3.1.6 脅威リスクアセスメント

セキュリティ上の脅威に対するリスクアセスメントは、ある特定の脅威が容易に実行可能かどうか、どれだけの損害をシステムに対して与えることができるかを評価するために非常に重要な要素である。脅威分析共通プラットフォームにおける脅威のリスクアセスメントとしての基本的な方針は、独自のものを開発するのではなく自動車業界で利用されるであろう複数のセキュリティ上のリスク評価標準を利用可能で、かつ個別にカスタマイズが可能な枠組みを提供することである。

(1) リスクアセスメントに対する基本要件

以下に、リスクアセスメントに関する基本要件を示す。

① SM-基本要件 1）
脅威分析において同定された脅威に対して､様々なリスクアセスメント基準(例:CVSS、CRSS、RSMA、CC-CEM）が利用可能であること
② SM-基本要件 2）
脅威のリスクアセスメントが、脅威単体でも脅威の詳細分析（例：アタックツリー）からも可能であること
③ SM-基本要件 3）
多段攻撃と多段防御を考慮した脅威のリスクアセスメントができること
④ SM-基本要件 3-1）
脅威に対してその対抗策（セキュリティ要求）により、どれだけリスクが低減されたかをアセスメントできること

注：ここで SM は Security Metrics の略。

脅威のリスクアセスメント方式は業界（そしてその業界における規格、ガイドライン）により異なる。自動車技術会が策定した JASO TP 15002:2015[12]では、情報システムで利用されている米国インフラストラクチャ諮問委員会（National Infrastructure Advisory Council）で開発された CVSS（Common Vulnerability Scoring System）[13]を元に作成された CRSS（CVSS based Risk Scoring System）や、ISO/IEC 27005[14]を元に開発された RSMA（Risk Scoring Methodology for Automotive systems）が用いられている。

それに対して、V2X のセキュリティに関しては、ISO/IEC 15408（Common Criteria）[15]ベースの CC-CEM[16]が用いられることが予想される。このように自動車システムを構成するデバイスに対して様々なリスク評価基準が利用される可能性があり、脅威分析共通プラットフォームではこれらが利用可能な枠組みをツールとして提供することを目指している。

(2) アタックツリーによるアセスメント方式

V2X セキュリティの研究プロジェクトであった EVITA[17]においては、上述の CC-CEM ベースにアタックツリーを用いて計算する方式を用いている。脅威分析共通プラットフォームにおいては、アタックツリーにおける脅威のリスクアセスメント計算の設計を CC-CEM を例に図 3.1.6-1 に示す。

ここで示す「攻撃」はシステムレベル脅威分析手法で定義したモデルと同様だが、セキュリティのリスク計算の値を security integrity level という属性への格納を追加している。セキュリティメトリックス計算では様々な計算方式を支援できるように、リスク計算全体を統括する「セキュリティメトリックス計算」が定義され、下位にあたる特定のセキュリティメトリックスとその計算方式を管理する。ここでは CC-CEM の計算方式のモデルを示している。

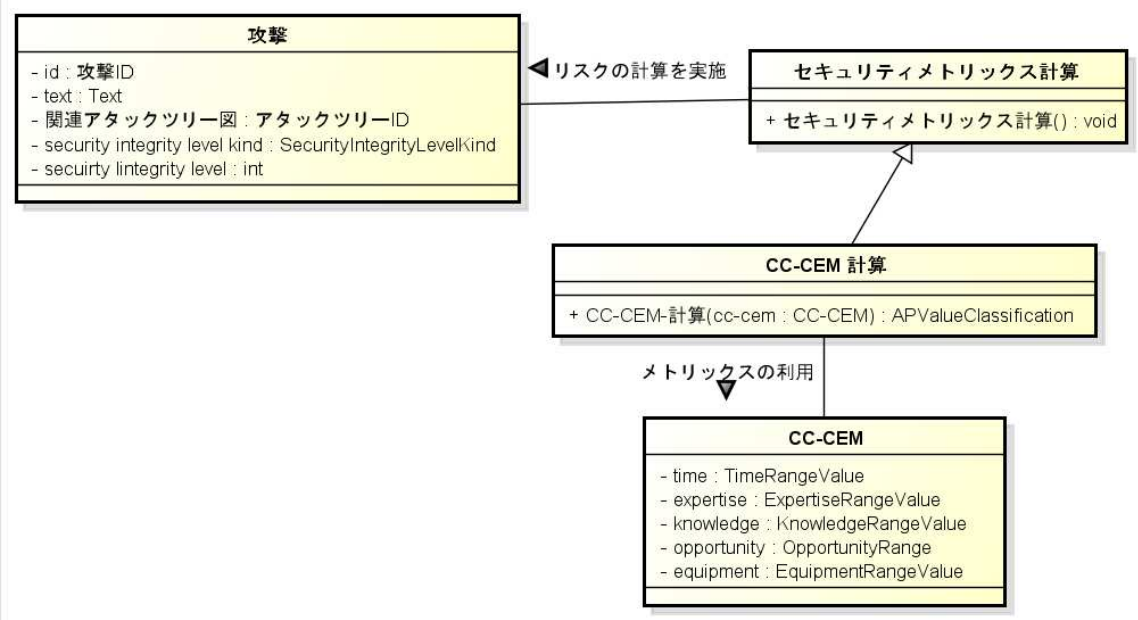


図 3.1.6-1　メトリックス計算の設計

CC-CEM 計算の下に定義されているのが CC-CEM において定義されているリスクの計算パラメータであり、5 種類の評価軸（time，expertise，knowledge，opportunity，equipment）で構成されている。「CC-CEM-計算」操作はこの 5 つのパラメータを値として取り（cc-cem : CC-CEM）、その結果攻撃の潜在的なリスクを示すアタック潜在力（Attack Potential）とそのレベル APValueClassification を返すものである。図 3.1.6-2 に攻撃潜在力に関するレベル分けのデータ構造と、攻撃時間のパラメータである TimeRangeValue の定義を示す。

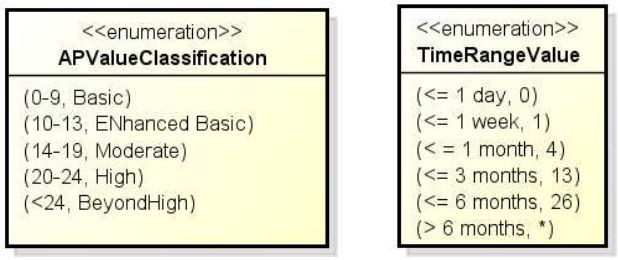


図 3.1.6-2　メトリックス計算のデータ設計例

これらのデータは通常はマトリックス（表）として表されているものであるが、例えば、攻撃潜在力の分類 APValueClassification においては、数値の範囲（例：0-9）と範囲内の場合の返り値 Basic を（0-9，Basic）といった表現で記述している。

CC-CEM によるセキュリティメトリックスに対するリスクアセスメント計算をアタックツリーにより実施する方式は EVITA[17]で行われたものであり、脅威分析共通プラットフォームにおいても採用する方式である。アタックツリーをリスクアセスメントに利用する方式は、必ずしも一般的でないが（例えば、前述の JASO TP15002:2015[12]において採用されている CRSS は採用していない）、攻撃をより詳細に分析し、そこからリスクのアセスメントをするので、より正確なリスクアセスメントが可能である。CRSS のようにアタックツリーを採用していない場合においても、アタックツリーをリスクアセスメントに利用可能なように工夫することは可能である。以下にリスク評価基準 DREAD に対して利用する場合を説明する。

DREAD はマイクロソフト社の研究者により考案されたリスクの評価基準である。マイクロソフト社自身は 2008 年以降利用していないとのことであるが、OpenStack（マイクロソフト社のクラウドプラットフォーム）他において利用されていると言われている。DREAD は以下の評価軸の頭文字を取ったものである。

- Da（damage or damage potential）　損害もしくは潜在的損害
- R（reliability or reproducibility）　（攻撃の）信頼性もしくは再現性
- E（exploitability）　攻撃される可能性
- A（affected users）　影響があるユーザ
- D（discoverability）　脅威の発見容易性

DREAD への批判の一つとして評価基準の独立性の欠如が指摘されている。しかしマイクロソフト社の別の基準である STRIDE よりは、個々のファクターの独立性は高いといえる。

計算のためのスコアは、1~10 もしくは Low，Medium，High，Critical という 4 レベルで行われる。ここで 1 は確実性が少なく、10 は確実性が高いとしている。

リスク値の計算方式としては、以下のように求められる。

リスク値 = (Da + R + E + A + D) / 5

アタックツリーにおける、AND-ゲートと OR-ゲートの計算方式は、EVITA の計算方式をそのまま利用する（AND = min、OR = max）。ここでは便宜上スコアを 1~10 とし、以下の 4 段階の値の範囲で攻撃の容易さを計算することにする。

- 1.0 ≦ x ≦ 2.5: 1
- 2.5 ＜ x ≦ 5.0: 2
- 5.0 ＜ x ≦ 7.5: 3
- 7.5 ＜ x ≦ 10.0 : 4

この場合、AND ゲートにおける計算は図 3.1.6-3 のようになる。

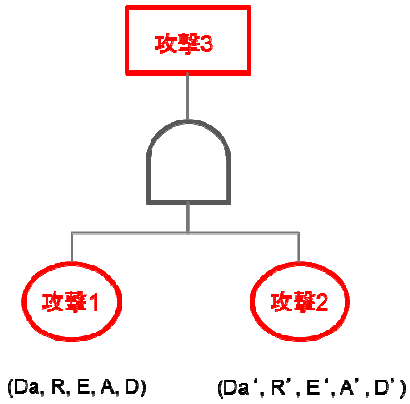


図 3.1.6-3　DREAD のアタックツリーにおける計算

すなわち、攻撃 3 の攻撃潜在力は以下のように計算される。

- min{ AP1, AP2}
- AP1 = (Da+R+E+A+D)/5
- AP2 = (Da'+R'+E'+A'+D')/5

OR ゲートに対しては、上記の min を max に変えることで計算が可能である。

脅威分析共通プラットフォームは、様々なセキュリティメトリックスと計算方式の提供することを目指しており、DREAD はその一例となるものである。

(3) リスクアセスメントに関する課題

ここまでの検討過程において、リスクアセスメント方式として新たな課題が多く見つかった。ここでは、それらの課題について説明をする。

① 多段攻撃、多層防御におけるアセスメント方式

多段攻撃と多層防御を考慮したリスクのアセスメント方式（SM-基本要件 3）の開発における課題を説明する。図 3.1.6-4 で示す多段攻撃と多層防御のためのリスクアセスメントとは、図 3.1.1-3 で示された多段攻撃・多層防御のモデル化に関するものである。ここで、

リスクのアセスメントとしては、図 3.1.6-4 の①、②、③の三段階で考慮することが可能である。

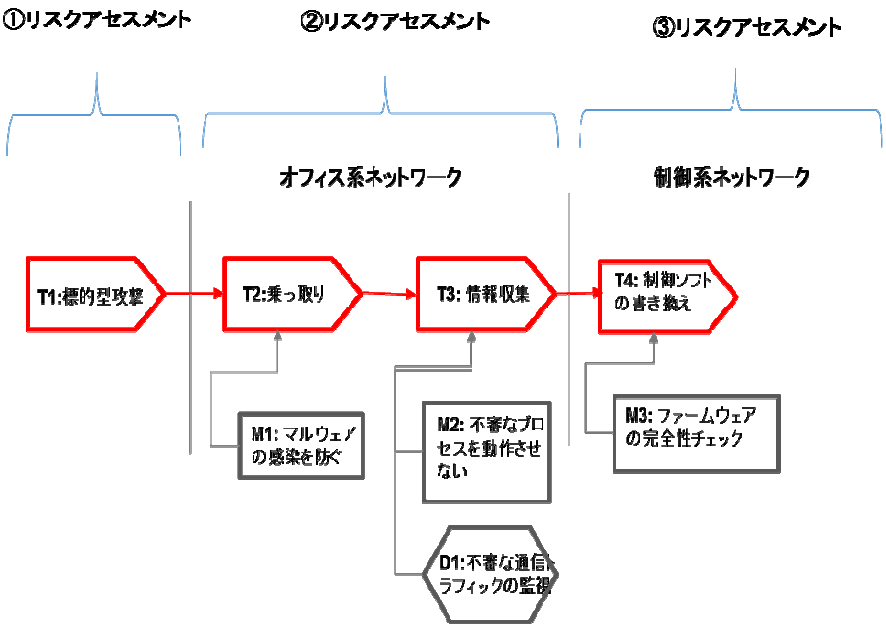


図 3.1.6-4　多段攻撃・多層防御のリスクアセスメント

リスクアセスメントの考え方としては、図の①は②と③を統合した形でのアセスメントとして考えられる。それに対して多層防御側から考えると、②によりセキュリティ上のリスクが低減され、さらに③によりリスクが低減されるという以下の式が成立するのが理想である。

アセスメント結果（③）≦アセスメント結果（②）

実際には、一旦、システム内に侵入されると検知するのが難しく、リスクは大きくなる可能性が高い。

このような形でセキュリティ上の脅威をアセスメントする方式はまだ確立していない。

② ROI と ROA によるアセスメント方式

セキュリティ上の脅威のリスクアセスメントに関して、ここまではシステム開発レベルにおいて詳細なアセスメントを行う際の方法について述べてきたが、製品の企画レベルおよび経営レベルにおいて重要なことは、どれだけのセキュリティ上の投資を行えばどれだけセキュリティ上の損失を防ぐことができるか、という点である。このようなアセスメントは、図 3.1.6-4 の①の段階で実施できることが望ましい。このようなリスクアセスメント方式としては、ゲーム理論（ディフェンスツリー）と経済指標を用いたものが知られている。

経済指標としては、以下に示す ROI（Return of Investment）と ROA（Return on Attack）を用いたものが基本的に利用されている。

（参照）攻撃と防御の経済指標

【経済指標】

- ROI (Return of Investment)
  - ✧ $\mathbf{ROI = \frac{(ALE * RM) - CSI}{CSI}}$ [18]
    - ALE (Annualized Loss Expectancy)
    - ALE = AV * EF * ARO ([18][19])
      - AV (Asset Value)
      - EF (Exposure Factor)
      - ARO (Annualized Rate of Occurrence)
    - RM (Risk Mitigated) , where RM ∈ [0,1]
    - CSI (Cost of Security Investment)
- ROA (Return on Attack) ([20])
  - ✧ $\mathbf{ROA = \frac{GI}{Cost\ before\ S + loss\ caused\ by\ S}}$
    - Security measure S
    - GI (Expected Gain)

このようなリスクアセスメント方式を、本システムレベル脅威分析手法への導入を検討する必要がある。

③ 攻撃と対抗策の双方を考慮したアセスメント方式

従来のセキュリティ上の脅威のリスクアセスメントは攻撃の容易性、難易度、その影響を基準にしていた。しかし、上記の経済指標のように防御と攻撃の双方の評価を元に、相対的な脅威のアセスメントをする方式の方がリスクがどれだけ軽減されたかが明らかに出来るはずである。このようなアセスメントのために攻撃と防御の双方を一つの記法の中で記述する方式が提案されている。

図 3.1.6-5 で示したものは、アタックツリーに検知と対抗策を同時に記述する方式である。

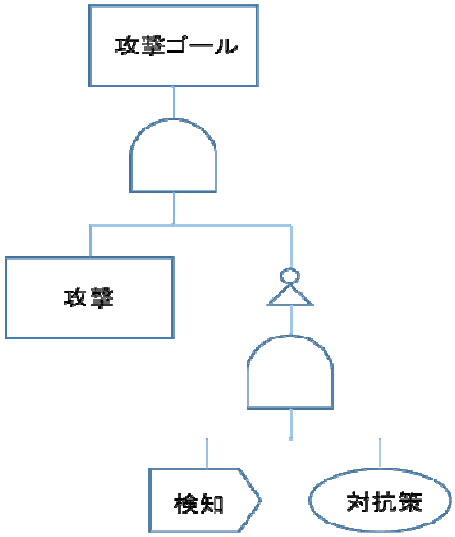


図 3.1.6-5　攻撃と防御の双方の記述（ディフェンスツリー）

計算方式としては以下が提案されている。

$$P_{攻撃ゴール} = P_{攻撃}（1－P_{検知}＋P_{検知}（1－P_{対抗策}））$$

本計算式では、検知が行われなかったか（$P_{攻撃}（1－P_{検知}）$）、検知が行われたが対抗策が無効であった場合（$P_{攻撃}P_{検知}（1－P_{対抗策}）$）に解釈される。

このような方式を複雑な評価基準（CC-CEM など）にどう適用するかは解決されていない課題である。

④ 複数の対抗策を考慮したアセスメント方式

実際に脅威分析をすると、一つの脅威に対して複数のセキュリティ上の対抗策が考えられる場合がある。そのような場合、どちらの対抗策の方が効果的かまたコスト上有利か、といった点が問題となる。開発現場においてはこのような事例が往々にして起き、その際のリスクアセスメントが重要な課題となっている。これを上記のディフェンスツリーに当てはめると、複数の対抗策があった場合の記述方式と計算方式が必要になる。このような研究はこれまでに知られておらず、研究が必要な分野であると言える。

本研究においては、リスクアセスメントに関しては評価基準・評価の計算方式についての調査を行い、その計算方式をアタックツリー・セキュリティ要求との関連で計算する方式の設計を行った。本報告書ではその設計の一部を CC-CEM を例に示した。

セキュリティのリスクアセスメントに関してはまだまだ研究すべき要素があり、それを課題として示した。それによりツールでの実現可能な部分と課題を明確にすることが出来た。

3.1.7　セキュリティ要求記述

セキュリティ要求（security requirement）はセキュリティ上の脅威に対抗する対抗策（countermeasure）、セキュリティコントロール（security control）などの総称としてここでは考えている。セキュリティ要求の記述は SysML の要求図をセキュリティ拡張することにより行われる。本拡張方法は SysML の要求図に対してセキュリティ上の様々な属性を定義することによる保存的拡張であり、その点では SysML における要求図に対する従来通りの利用も可能であるという利点がある。

ここではまずセキュリティ要求記述に対する基本要件を示す。次に UML のクラス図によるセキュリティ要求記述のモデル化を示し、最後に実際の要求の記述方式を示す。

(1) セキュリティ要求に対する基本要件

セキュリティ要求記述に対する基本的な要件を以下に示す。

① SR-基本要件 1）
記述の抽象度によるセキュリティ要求の分類、詳細化が出来ること。必要に応じてセキュリティ要求間の関係を独自に定義できること。
② SR-基本要件 2）
セキュリティ要求の種類の分類が出来ること。
③ SR-基本要件 3）
セキュリティ要求に対するリスク評価値、その評価に利用されたセキュリティメトリックスが明示出来ること。
④ SR-基本要件 4）
セキュリティ要求と他の分析、評価作業の成果物間のデータとしてのリンクが出来ること。

注：ここで SR は Security Requirement の略。

セキュリティ要求は記述内容の抽象度により詳細化されていることを明示的に定義し表示できるようにする。詳細化は要求間のリンク関係により示される。これは、基本的には要求工学における GORE（Goal Oriented Requirements Engineering）の考え方と同様であり、さらに ISO 26262 においても安全要求の階層化に利用されている。さらに、セキュリティ要求間に必要に応じて任意の関係を定義し、モデル要素として要求を記述する図表現において表現できるようにする。

J3061 においては、機能サイバーセキュリティ要求（functional cybersecurity requirements）、技術サイバーセキュリティ要求（technical cybersecurity requirements）により、開発プロセスの各段階のセキュリティ要求を分けている。これらの区別は、元々自動車の機能安全規格である ISO 26262 の Part 3 と Part 4 における機能安全要求（functional safety

requirements）と技術安全要求（technical safety requirements）の区別と相似の関係にある。ISO 26262 においてはハザーダスイベント（危険事象/hazardous event）に対抗する安全要求の最も抽象度が高いものを安全ゴール（safety goal）と呼ばれる。同様に J3061 においてはサイバーセキュリティゴール（cybersecurity goal）と呼ばれる。これらの区別は各要求にその属性として指定することを可能にすることで、明示的に示すことが出来る。これらのセキュリティ要求の種類については、セキュリティ拡張における要求の種類（requirementKind）（図 3.1.7-2）において示されている（すなわち J3061 のサイバーセキュリティゴールであることを示すには、要求の種類として cyberSecurityGoal を値として取れば良い）。

セキュリティ要求は関連する脅威のリスク評価値を継承するという考え方がある。その際に、評価値とその評価の元になったセキュリティメトリックスを明示的に示すことができる。

(2) セキュリティ拡張

ここでは、セキュリティ拡張のための基本的な設計を実施した。セキュリティ拡張は主に、SysML の基礎である UML（Unified Modeling Language）の拡張機能であるステレオタイプを用いている。

図 3.1.7-1 に示すものが、セキュリティ要求をモデル化したものである。

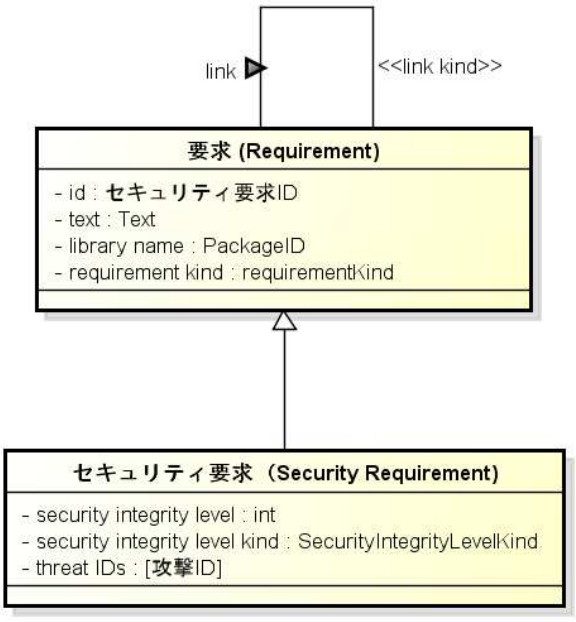


図 3.1.7-1　セキュリティ要求のモデル図

このクラス図の各要素に関する説明を表 3.1.7-1 に示す。

表 3.1.7-1　セキュリティ要求のクラス定義.

クラス	属性	属性の意味
要求（Requirement）	id	要求の ID を示す（id は、SysML の要求図のものをそのまま利用するので、その意味論に従う必要がある）
	text	要求の記述が入る（text は SysML の要求図のものをそのまま利用するので、その意味論に従う必要がある）
	library name	要求の集まりを指定する場合に用いる。library name はユニークである必要がある。 ただし、ここで library name として UML/SysML におけるパッケージを利用することを考えており、同一のセキュリティ要求も、別のパッケージの中で管理することが可能である。 この利用方法としては、同一のセキュリティ要求を、別のリスクアセスメント方式でリスクを計算することなどを想定している。
	requirement kind	要求の種類を示す。型として、requirementKind を持つ。
セキュリティ要求（Security Requirement）	security integrity level	記述されたセキュリティ要求のリスクアセスメントされた結果を値として持つ。
	security integrity level kind	どのようなセキュリティインテグリティレベルを利用したかを、値として持つ（複数のセキュリティインテグリティレベルを利用可能なことを想定しているため）。型として securityIntegrityLevelKind を持つ。
	threat IDs	対抗する脅威を表す。対抗する脅威の ID（複数個可能なことを[攻撃 ID]で示している）を値として持つ。 この id は、表 3.1.5-1 における攻撃の id に対応する。

基本的には SysML の要求図の基本的な箇所（id や text）はそのまま利用し、新たに導入したものは library name（セキュリティ要求をライブラリ化して整理、再利用する際に用いる）、requirement kind（セキュリティ要求の種類を明示する。図 3.1.7-2 に型定義を示す）などである。それに対して、セキュリティ要求では、security integrity level（対応する脅威

のリスクアセスメント値が入る)、security integrity level kind（利用されたセキュリティインテグリティレベルの名称が入る)、threat IDs（対抗する脅威の ID を示す。これにより脅威の情報とそれに対するセキュリティ要求が関連付けられる）が定義されている。

さらに、要求間のリンクは<<link kind>>ステレオタイプで定義される。

図 3.1.7-2 に型定義（枚挙型として定義されている）の例として requirementKind（要求の種類）と SecurityintegrityLevelKind（セキュリティインテグリティレベルの種類）の型の定義例を示している。本定義は利用者が拡張することが可能となっている。セキュリティ要求の種類としては、J3061 に準拠した分類（cyberSecurityGoal、functionalcyberSecurityRequirement、technicalcyberSecurityRequirement）とより一般的な分類（securityGoal、securityRequirement、functionalSecurityRequirement、technicalSecurityRequirement）の両者が定義されていることに注意が必要である。

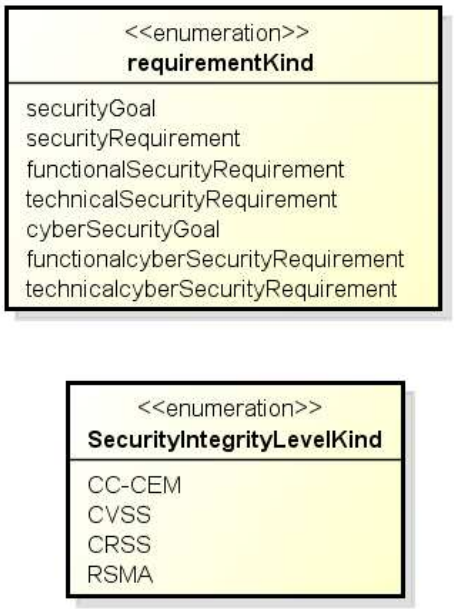


図 3.1.7-2　セキュリティ要求に関連する型定義

(3) 具体例

以下の図 3.1.7-3 に示す例は USB のファームウェア書き換え攻撃に対するセキュリティ要求を示したものである。ここではセキュリティ目標（<<securityGoal>>）として「USB のファームウェの書き換えを防ぐ」を設定し、より詳細なセキュリティ要求（<<securityRequirement>>）として「署名認証をする」、「接続 USB のホワイトリストを作る」、「ファームウェアの書き換えを禁止する」という詳細なセキュリティ要求を導出したものである。ここでは、セキュリティ要求間の詳細度を示すために<<rifine>> リンクを用いた。

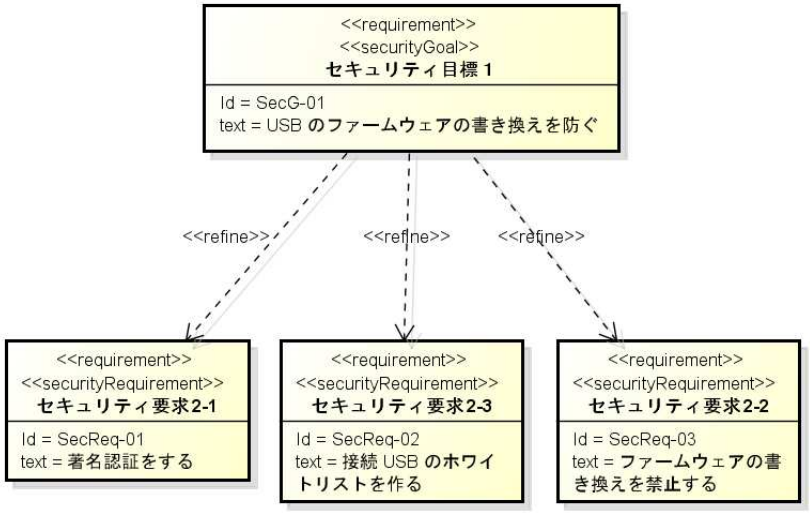


図 3.1.7-3　セキュリティ要求記述例

以上述べたように、セキュリティ要求に関して SysML の要求図におけるセキュリティ拡張方式の考案および他の手法との連携の設計を行った。

3.1.8　まとめ

自動車がコネクトされ自動運転が導入される際に避けて通れないセキュリティの問題を解決する方法の一つとして、セキュア開発プロセスの構築とそのプロセスを支援するツールの開発は非常に重要である。本テーマにおいては、セキュア開発プロセスの中でも特に重要な脅威分析フェーズに対して、統合的に支援する手法とそのための開発環境を「脅威分析共通プラットフォーム」として開発することを目指し、基本要件の導出、基本要件に基づく手法の開発、様々な手法のツールとしての統合方法、各手法とそれらの連携をツールとして実現するための設計を実施した。このような開発支援ツールは世界的に見ても存在せず、日本の自動車産業におけるセキュア開発を強力に支援する道具になることが期待されるものである。

今回開発した手法は、最新の研究成果を俯瞰しつつその良い点と欠けている点を明確にし、実用的に適用できる範囲で手法として取り入れたものである。採用した手法、それらを統合する方法については、既に生産現場で実施されている自動車の機能安全規格 ISO 26262 の考え方やサイバーセキュリティ規格 J3061 に示されたフレームワークを元に、統合のための新しいアイデアを導入した。

・手法全体の統合・連携方式の考案
　－自動車システムのセキュリティ開発における脅威分析プロセスの支援
　－スタンドアロンツールでは無く、外部の脅威情報との I/F の提供
　－セキュリティ要求、アーキテクチャ記述、攻撃の分析（アタックツリー）、システムーレベルの脅威分析、リスクアセスメント方式の有機的な連携方式の考案
　－SysML のセキュリティ拡張（セキュリティ要求、アーキテクチャ記述）

・システムレベルでの多段攻撃と多層防御を明示的にモデル化・分析が可能な脅威分析手法の考案
ーセキュリティのリスクアセスメント方式についての課題の明確化。

＜参考文献＞

[1] SAE: J3061, Cybersecurity Guidebook for Cyber-Physical Vehicle Systems, 2016

[2] ISO: ISO 26262: 2011, Road vehicles - Functional safety

[3] CWE: https://cwe.mitre.org/

[4] NVD: https://nvd.nist.gov/

[5] JPCERT: https://www.jpcert.or.jp/

[6] ICS-CERT: https://ics-cert.us-cert.gov/

[7] Auto-ISAC: https://www.automotiveisac.com/

[8] SysML: www.omgsysml.org/

[9] AADL: http://www.aadl.info/aadl/currentsite/

[10]J. Jurjens: Secure System Development with UML, Springer 2005

[11] T. Lodderstedt, D. Basin, J. Doser: SecureUML: A UML-Based Modeling Language for Model-Driven Security, UML '02, pp 426—441

[12]JASO: TP-15002 「自動車の情報セキュリティ分析ガイド」, 2015

[13] CVSS: https://www.ipa.go.jp/security/vuln/CVSS.html

[14] ISO/IEC 27005: 2011, Information technology — Security techniques — Information security risk management

[15] ISO/IEC 15408, Information technology -- Security techniques -- Evaluation criteria for IT security

[16] ISO/IEC 18045, Common Methodology for Information Technology Security Evaluation

[17] EVITA: Deliverable D2.3: Security requirements for automotive on-board networks based on dark-side scenarios, 2008

[18] Stefano Bistarelli, Fabio Fioravanti, Pamela Peretti: Defense trees for economic evaluation of security investments. ARES 2006: 416-423

[19] Stefano Bistarelli, Marco Dall‘Aglio, Pamela Peretti: Strategic Games on Defense Trees. Formal Aspects in Security and Trust 2006: 1-15

[20] Marco Cremonini, Patrizia Martini: Evaluating Information Security Investments from Attackers Perspective: the Return-On-Attack (ROA). WEIS 2005

3.2　車両への攻撃に対する対策の評価手法・認証の調査・研究

車両のセキュリティに関する評価技術・評価基準の検討を目的として、本事業で設定した階層「車両全体」、「車内システム」、「コンポーネント」に対して実機およびシミュレーションを用いた攻撃手法の検討・再現・評価を実施した。また、車両全体の評価環境として、平成 27 年度に策定した標準アーキテクチャに準拠した車両模擬システムを構築した。

さらに、セキュリティの評価・認証について有識者による研究会を開催し知見を得た。

3.2a　セキュリティ評価用車両模擬システム構築

自動車が、様々な通信によって車両外部と接続されるようになってきており、それとともにセキュリティが課題となっている。特に、セキュリティ攻撃の主な口となる車両外部との通信を経由して行われる攻撃に対して、どのように評価を行うのか、また、攻撃された結果として、車両がどのような動作を行うのかを観測できるような評価環境を構築することが非常に重要である。このような評価環境は、実装しているセキュリティ対策が有効に機能していることの確認や、その評価技術の研究を行う上で重要な役割を果たすものと考えられる。これまでに、IT 分野では大規模ネットワークを模擬した Starbed 4（http://starbed.nict.go.jp/）があり、重要インフラ分野では技術研究組合制御システムセキュリティセンター（http://www.css-center.or.jp/index.html）に模擬プラントシステムがテストベッドとして構築され、評価・研究に活用されている。

こうした例を参考として、自動車のセキュリティ向けの評価環境を考えると、まず自動車の場合に攻撃の口となる車外との通信、その情報を受け必要な情報を車内に伝達するゲートウェイ､そして車内ネットワークとそれに接続される ECU といった階層構造からなる構成要素を持つことが必要と言える。

車両外部との通信経路としては、V2X や、WiFi/Bluetooth、携帯電話網利用の通信（TCU : Tele-Communication Unit）、スマホ等の持ち込み機器（Nomadic Device）、GPS（Global Positioning System）、TV/FM、TPMS（Tire Pressure Monitoring System）など様々なものがある。これらの通信はそれぞれ異なる通信プロトコルを持ち、車両内部には異なる経路、あるいは異なる接続ポートに接続される。

これら車外からの情報を受けるゲートウェイにおいては、Firewall やメッセージフィルタリングなどの機能が実装されることが想定される。また、車内ネットワークでは信号の暗号化や、接続される機器の認証などのセキュリティ対策が実装される可能性がある。

これとは別に、機能安全の仕組みとして組み入れられた冗長性が、車両の制御に対するセキュリティ防衛策として有効に機能する可能性もあるため、外から自動車の挙動を見ているだけでは攻撃されているのかどうか分からないといったケースも起こり得る。

こうした自動車特有の事情を考慮した上で、本テーマでは、これら車両外部からの攻撃に対して、どの様に評価を行い、その結果として、車両がどういう挙動を示すのかをビジュアルに表現しつつ、車両内部のネットワークの状態をモニタ出来るシステムとして車両模擬システムの開発を行うこととした。

3.2a.1　セキュリティ評価用車両模擬システムの概要

セキュリティ評価用車両模擬システムに求められる要件は、以下のように設定した。

・車両模擬システムとして、基本的なアーキテクチャは本事業において想定したもの（図 3.2a.1-1）に準拠すること。即ち、車両外部との通信のためのポートを持つゲートウェイと、ゲートウェイを介して接続される複数に分割された内部ネットワークを持つこと。
・車両動作を模擬するものとして、走る・曲がる・止まるの機能を実装していること。
・走る・曲がる・止まるの機能のうち、どれかは機能安全の仕組みを実装していること。
・システムを構成する HW として、セキュア IP を実装しているものを含むこと。
・セキュリティ対策技術は様々であり、今後も進歩していくことが想定されるため、システムに実装される SW は書き換え可能であること。あるいはマイコン/メモリの差し替えで対応可能なこと。
・本システムの目的は、攻撃されても破られないシステムを作ることではなく、セキュリティ評価技術の研究・向上が主であることから、セキュリティ対策は、基本的に多層防御とし、各階層ごとにセキュリティ強度が可変出来ること、また場合によっては、故意にセキュリティホールを作ることが出来るようにしておくこと。
・研究・教育への活用も見据え、内部ネットワークの仕様のうち必要な情報については、評価実施者に限定するとしても、原則、公開可能であること。

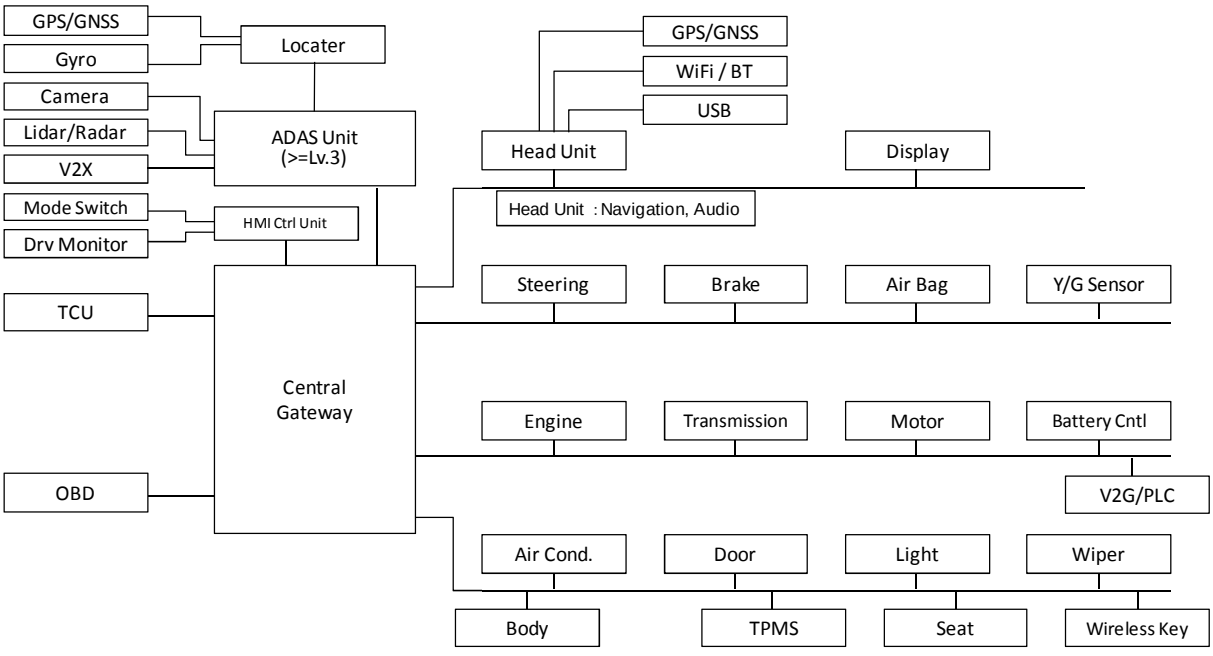


図 3.2a.1-1　本事業で想定した基本アーキテクチャ

(1) 車両模擬システムの構成

これらの要件を満たすものとして、本テーマで構築する車両模擬システムの基本構成を図 3.2a.1-2 に示す。ここでは、模擬する車両として EV を想定し、TCU や V2X、WiFi/Bluetooth 等の通信により得られる車両外部からの情報と、操舵系やモータ制御、ブレーキ制御などの ECU が接続される車載ネットワークが、セントラルゲートウェイを介して接続される構成とする。

従って、セントラルゲートウェイには、車両内部の LAN 接続として最低 3 系統（そのうち、少なくとも操舵系ユニットは機能安全の仕組みを実装する前提のため 2 重化が必要）、車両外部との通信機器として TCU、WiFi/BT、V2X 機器が直接、あるいは Head Unit／ADAS Unit を介して接続可能な複数のポート、および、OBD-II ユニットを接続するポートを具備することが必要である。

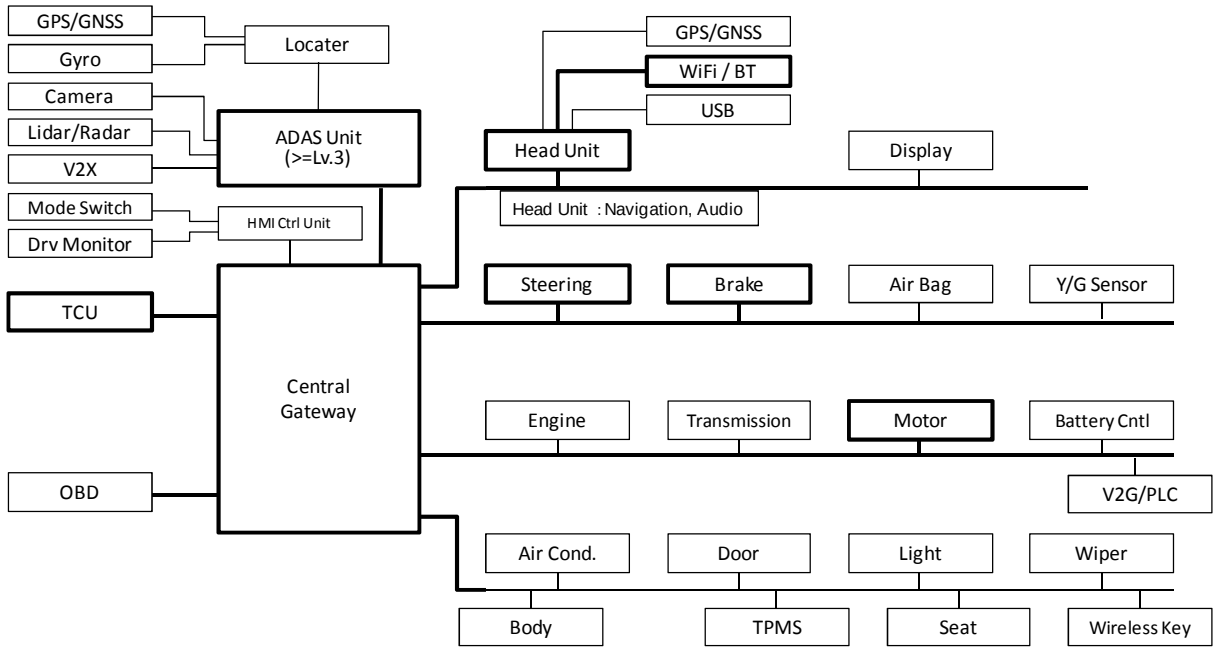


図 3.2a.1-2　本テーマで構築する車両模擬システム

ここで示した基本構成のコンセプトを実証するものとして、まず、この基本構成のワンパスを構築する。ワンパスは、車両外部からの通信を模擬するものとしてのダミー基地局やスマホ等から繋げることが出来る TCU を模擬した通信ユニット、通信ユニットが接続されるゲートウェイ、ゲートウェイの車両内部のネットワークの一つに接続されるダミーECU から構成するものとし、このうち、ゲートウェイとダミーECU 間の通信には、セキュリティ対策技術として TPM を用いた暗号化通信を搭載するものとする。なお、ダミーECU は、操舵系 ECU や、ブレーキ制御 ECU などの代わりとして用いている。

また、TCU を模擬した通信ユニットへの攻撃としては Jamming などの電波を利用した攻撃も考えられることから、TCU への RF 信号入力の経路を構成するものとして、RF 発生器を使用した評価についても検討が必要である。

(2) 車両模擬システムを含む評価システム（テストベッド）

車両模擬システムを用いたセキュリティ評価を行うには、上述した車両模擬システムに加え、評価ツールや観測装置の組み込みが必要となる。これらを組み込んだものがセキュリティ評価用車両模擬システム（図 3.2a.1-3、以降「テストベッド」と称す）となる。

テストベッドに組み込む評価ツールとしては、独自に開発することも考えられるが、既に市販されているツールも存在している。特に、Codenomicon の Difensics はファジングツールとして知られており、「認証研究会」（3.2f 節を参照）においても議論されている通りツールの使いこなしもセキュリティ評価技術の研究を進める上で重要であることから、Difensics の利用を想定する。また、この他に同等または類似の評価ツールがある場合は、その比較を行う。

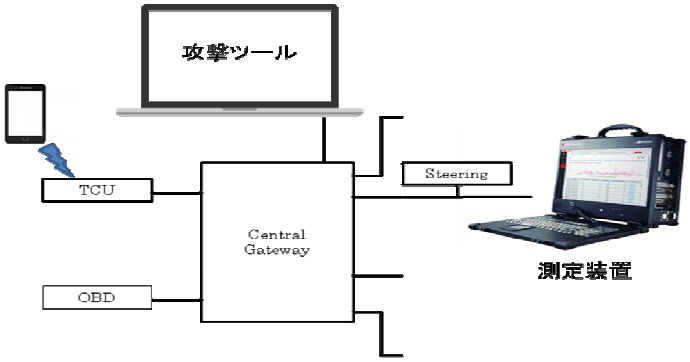


図 3.2a.1-3　セキュリティ評価用車両模擬システム

テストベッドの構築を検討するにあたり、まず、IT 業界での知見などをベースとして、ゲートウェイに求められる機能等について整理したうえで、車両模擬システムの具体化について検討を行った。

3.2a.2　車内・車外通信とセキュリティ評価基盤

自動車内部で各種制御を行う車載ネットワークは、我々が日常利用する IP（インターネットプロトコル）ネットワークとは異なる独自の通信プロトコルによって規定される。低速型の LIN（Local Interconnect Network）、中高速型の CAN（Control Area Network）、マルチメディア機器向けの MOST（Media Oriented Systems Transport）、高速で動作し、故障保証までの機能を有する FlexRay などがある。これらの車載ネットワークは、その用途により通信プロトコル以外にも必要とされるデータ転送速度に差異があるため、通常セントラルゲートウェイを起点として複数の異なる LAN（Local Area Network）で分離されている。なかでも CAN は、ボディ系 (エアコン、ドアなど内装品)、パワートレイン系（エンジン制御など）、情報系 (カーナビやオーディオなど)など、今日もっとも広く利用される車載ネットワークプロトコルである。

CAN では、ライン上に形成されるバス型トポロジーに、複数の ECU（Electrical Control Unit）と呼ばれる分散型のコンピュータが接続され、車両の細部に対して様々な制御を行

う。しかしながら、IP ネットワークにおける経路、優先、廃棄など高度なネットワーク処理を行うルータは、CAN には存在しない。CAN の通信パケット（フレーム）は、ECU からブロードキャストとして送信され、フレーム内の ID（Identifier）が示す情報によって、簡易的な経路や優先制御さらには廃棄制御をエンドノードである ECU 自身が処理する。IP ネットワークとのこうした違いから、CAN のセキュリティについては、以前から多くの問題が指摘されている。不正 ECU への交換、悪意を持った OBD（On Board Diagnostics）ツールの接続などは、攻撃者にとって容易かつ絶好の侵入経路となる。そのため、車載ネットワークに対する攻撃と防御手法について、また想定される新たな脅威を未然に防止するための評価手法について、継続的に議論され、車両内ネットワークのシミュレーション環境も報告されている（「車載ネットワークのセキュリティ監視システム」2015 年 7 月 SEI テクニカルレビュー・第 187 号）。

すでに自動車は車内から車外へと、その通信範囲を大幅に拡張した。自分自身の位置情報を取得することから、自動車が発信する情報をまとめた急ブレーキマップ（ ex. ホンダ SAFRTY MAP http://www.honda.co.jp/safetymap/ ）など、車外との通信により、自動車はこれまでに成し得なかった価値を享受し、また社会へ価値を提供するようにもなった。今後は、車載ネットワークに接続される ECU も車外との通信を利用したファームウェアやソフトウェアの更新が行われるようになり、メンテナンスの容易化やリコール問題の早期解決など、新しい価値を創出するものと考えられる。同時に自動車は、従来議論された自動車内部のネットワークのセキュリティ問題に加えて、車外から受ける脅威と、誤って車外へ発信してしまう脅威までを考えなければならない。

本節では、ネットワークセキュリティについて、先例となる IP ネットワークにおける様々な攻撃とその対策手法、また具体事例から、自動車が車外との通信することによって引き起こされる脅威を想定し、今後車両セキュリティに関わる多くの研究者が各種攻撃パターンや効果的な対策の検討などの評価を行うことを目的とした「テストベッド」について検討した結果をまとめている。

ここで検討するテストベッドの特徴となるセキュリティ機能は、総合型、且つ継続型と想定した。3.2a.3 で説明する様々なセキュリティ機能は、セントラルゲートウェイに実装され、トラフィックを動的に監視することにより、車外からの攻撃を動的に検知し、即応性が要求される場合は直ちに廃棄処理を行い、さらにログ情報として蓄積する。加えて、タイムスロット単位でトラフィック情報を収集し、経時的な変化から異常を検知する。統計情報の解析については、WAN 側回線を介して上位に要求するほか、簡易的なデバッグを目的とした仮想的なサンドボックスをセントラルゲートウェイ内部に設ける。

車載ネットワークへの攻撃のうち、車外との通信によって想定される脅威を表 3.2a.2-1 に示す。ここでは脅威を、車載ネットワークへの攻撃、標的型の攻撃、第三者への攻撃加担、車載ストレージへの攻撃に分類し、各々についての詳細を以降に記す。

表 3.2a.2-1　車外との通信によって想定される脅威

想定される脅威		侵入経路		
脅威・攻撃の分類	評価すべき攻撃	不正 ECU	OBD	車外通信
1．車載ネットワークへの攻撃	CAN バスのバッファオーバフロー	✔	✔	✔
	なりすまし、盗聴	✔	✔	✔
	DoS/DDoS/DRDoS の標的			✔
	ブルートフォース、ファジング			✔
2．標的型の攻撃	繰り返し型攻撃			
	侵入、拡大、調査、搾取			✔
	偵察、スキャン、攻撃、メンテナンス			✔
3．第三者への攻撃加担	DoS/DDoS の踏み台			✔
4．車載ストレージへの攻撃	各種履歴情報の改ざん、搾取	✔	✔	✔
	車両の制御情報の改ざん、搾取	✔	✔	✔
	車外・車内プロトコル変換情報改ざん			✔
	その他個人情報の改ざん、搾取			✔

OBD など、車両のデバッグインタフェースを用いて侵入するといった CAN の脆弱性を突く攻撃手法や、車内ネットワークのセキュリティに関する評価環境は、従来よく研究されている分野のひとつであるが、ここでは表 3.2a.2-1 において網掛けで示した、特に車外との通信によって想定される新たな脅威について着目した。代表的な攻撃と脅威について、以下のように整理した。

① CAN など車載ネットワークへの攻撃と脅威

現在多くの自動車は、CAN プロトコルによって車載 ECU 間の通信を行う。CAN は、マルチマスタ方式のバス型トポロジであり、その脆弱性について様々な報告や指摘が挙げられている。例えばバス上にトラフィックが集中した場合、バッファオーバフロー状態となり、全ての ECU は機能停止状態に陥る。バッファオーバフローとは、広義において、当初は予見し得ないトラフィックの発生によって引き起こされる資源の消耗と破壊であり、CAN バスへの攻撃手法についてシミュレーション結果も報告されている。(「繋がる組込みシステムの脅威とその対策」独立行政法人情報処理推進機構 2014 年 7 月)

従来の評価において、主な攻撃の侵入経路は OBD ポートと不正 ECU であったが、今後の車外との広帯域通信が開始されることによって、例えば Ethernet パケットによる連続型・標的型の攻撃が行われることが懸念される。したがって、評価用攻撃パターン生成と脆弱性調査など、新たな評価技術と評価基盤の整備が早急に必要である。特に今日の IP ネットワークにおけるサイバー攻撃は非常に高度化、複雑化していることが特徴で、標的となっていることになかなか気付かない場合がある。ECU や CAN バスが、知らずのうちに DoS/DDoS 攻撃の標的となる可能性は十分に考えられるため、経時的な監視を含めた評価基盤、評価環境構築を検討する必要がある。

② 特定 ECU など標的型の攻撃と脅威

昨今の IP ネットワークにおけるサイバー攻撃は、具体的なターゲットを絞ったものが多く、「標的型攻撃」と呼ばれている。ここでは、一度の攻撃だけで脅威となることは稀であるが、繰り返し攻撃を重ねることで、情報搾取など攻撃者の意図する最終目標を達成することが大きな特徴となっている。以下に 3 つの具体例を示す。

【例 1】

・第 1 段階　「Reconnaissance（偵察）」
　インターネット等を利用し、攻撃対象の情報を収集する。
・第 2 段階　「Scanning（スキャン）」
　攻撃対象者が提供するサービス、またどのような脆弱性が存在するかなどをスキャンする。スキャンにより攻撃の成功率が上昇する。
・第 3 段階　「Exploitation（攻撃）」
　Metasploit 攻撃ツールの利用、Exploit コード（攻撃プログラム、多種多様なプログラム言語で構成）などである。
・第 4 段階　「Maintenance（メンテナンス）」
　機密情報の継続的取得のため、攻撃者によるバックドアを仕掛ける

（出典）「事例から学ぶ情報セキュリティ」標的型サイバー攻撃のシナリオ、技術評論社 2015 年 1 月

【例 2】

・第 1 段階　「侵入」

攻撃者は、社内ネットワークのパソコンをウィルスに感染させて、橋頭堡（きょうとうほ）として利用する。

・第 2 段階　「拡大」
　感染パソコンが、社内ネットワーク内にある別のパソコンを見つけてウィルスを伝染させる。攻撃に必要な機能を追加するために、1 台のパソコンに複数のウィルスを感染させることもある。
・第 3 段階　「調査」
　感染パソコンからアクセスできるサーバを探し出し、機密情報など重要なデータを取り出せるかを確認する。

・第 4 段階　「取得」
感染パソコンがサーバにアクセスして、重要なデータを収集し、攻撃者が用意した外部のサーバに送信する。こうして企業の重要なデータが外部に漏洩してしまう。だが逆に言えば、この連鎖をどこかで断ち切りさえすれば、重大な被害が出る前に攻撃を防ぐことができる。

（出典）「サイバー攻撃の 4 ステップ」2014 年 8 月日経 NETWORK

【例 3】
・第 1 段階　「偵察」
インターネットなどから組織や人物を調査し、対象組織に関する情報を取得する
・第 2 段階　「武器化」
エクスプロイトやマルウェアを作成する。
・第 3 段階　「デリバリ」
なりすましメール(マルウェアを添付) を送付するなりすましメール(マルウェア設置サイトに誘導) を送付し、ユーザにクリックするように誘導する。
・第 4 段階　「エクスプロイト」
ユーザにマルウェア添付ファイルを実行させるユーザをマルウェア設置サイトに誘導し、脆弱性を使用したエクスプロイトコードを実行させる。
・第 5 段階　「インストール」
エクスプロイトの成功により、標的がマルウェアに感染する。
・第 6 段階　「C&C」
マルウェアと C&C サーバを通信させて、感染 PC を遠隔操作する。新たなマルウェアやツールのダウンロード等により、感染拡大や内部情報の探索を試みる。
・第 7 段階　「目的の実行」
探し出した内部情報を、加工(圧縮や暗号化等) した後、情報を持ち出す。

（出典）サイバーキルチェーンモデル Lockheed Martin

以上 3 つの例で示したように、標的型・連続型のサイバー攻撃では、偵察や侵入などウィルスの感染が第一歩であるが、真に脅威となるのはその後に続く攻撃者あるいはウィルスによって感染した機器による活動である。予測される侵入経路に関して、従来の OBD ポートや不正 ECU の接続に加え､車外ネットワークからの侵入経路は攻撃者にとって繰り返し攻撃を仕掛けることができるため、極めて大きな脅威となる。今後自動車のセキュリティが経時的な監視によってこれを対策することは必須であり、テストベッドにおいて各種攻撃パターンの生成から対策手法の検討や実行ができることは、極めて重要な技術課題である。

③ 第三者への攻撃を加担する脅威

これも昨今 IP ネットワークにおいて急増する深刻な脅威のひとつである。攻撃者の標的となった被害者がウィルス等に感染したことを認識できず、いつのまにか DDoS の踏み台として他の被害者を攻撃し攻撃者への加担を行ってしまう場合がある。この攻撃についても、今後車外との通信が拡大する車載ネットワークについて、容易に想定することができる。例えば、ファームウェアの更新や ECU の状況把握を目的として、車外から車載ネットワークさらには ECU に対して様々な要求通信がなされるようになる。それに対して、各 ECU は応答通信することになるが、この場合の要求通信が攻撃者が仕掛けた偽サーバからの発信でありさらに要求通信に記された発信元が標的被害者となるアドレスであれば、多くの車からの大量の応答パケットが一斉に標的に向かって通信を開始することとなる。IP ネットワークにおけるこのような DDoS 攻撃は、特に DRDoS 攻撃（Distributed Reflection Denial of Service）と呼ばれ、DNS（Domain Name System）や NTP（Network Time Protocol）の脆弱性を突く攻撃手法である。車載ネットワークのセキュリティの研究開発を効果的に進めるためには、本節で紹介した高度で複雑な攻撃パターンとなるテストや対策手法の開発と、評価が可能となるようなテストベッドを検討することが重要である。

④ 車載ストレージへの攻撃

車外との通信によって自動車が扱うべき情報量は、今後急速に拡大することが予想される。例えば次節で示すセントラルゲートウェイで取得されたトラフィック経時情報は、自動車本体､すなわちローカルで保存すべき情報と､クラウドなどで管理される情報に分割、最適化され、整理を進めるべきデータと考えられる。このほかにもドライブレコーダなど様々な用途において、ローカルでデータを扱う車載ストレージは、当然ながら攻撃者からのターゲットとなる。機能の停止や情報の改ざん、さらには搾取などを狙った様々な攻撃が考えられる。従って、テストベッドにおいても、車載ストレージデバイスとして外部からアクセスできる独立したインタフェースを物理的、仮想的に実装するなど、車載ストレージに対する攻撃パターンや対策手法について、研究開発が可能な評価基盤を提供する必要がある。

3.2a.3　評価基盤を提供するテストベッド

本節では､3.2a.2 で示した昨今の IP ネットワークセキュリティにおける脅威の事例から、IP ネットワークと接続される自動車に対する攻撃を想定し、今後研究者が有効な防衛手法を開発、評価するためのテストベッドについて具体仕様案を示す。本テストベッドは、脅威から守る様々なセキュリティ機能をセントラルゲートウェイに実装する他､LAN 側となる ECU に実装する防御手法や、車外 WAN 側での防御手法を接続して監視、評価することが可能な総合型、継続型の評価環境となるよう検討した。

テストベッドの仕様と内部を構成するコンポーネントを表 3.2a.3-1 に示す。

表 3.2a.3-1　テストベッドの仕様とコンポーネント

テストベッド仕様			
セントラル ゲートウェイ 本体	内部の実装 ユニット	トラヒック監視・制御 コプロセッサ	・多層監視：複数のレイヤの組合せで構成するアクセス制御 ・経時監視：タイムスロット単位でトラヒックを監視 ・統計処理：ログ情報の管理と上位（WAN側）との通信 ・解析処理：動的・静的解析、仮想的サンドボックス環境 ほか
		車載ストレージユニット	・用途を特定しないストレージ（メモリ）資源として ユーザに開放
		WAN-LAN間のアドレス変換 代理サーバ	・ネットワークアドレス変換テーブルの管理 ・LAN側の代理サーバ機能によるLAN側ネットワーク保護
	外部との インターフェース	Ether	・WAN側との通信用インターフェース
		CAN	・LAN側との通信用インターフェース（Rawバス）
		SPI	・LAN側との通信用インターフェース（Securedバス）
		JTAG	・セントラルゲートウェイメンテナンスインターフェース
セントラルゲートウェイ WAN側の接続		ハードウェア制御・処理 パケットジェネレータ	・各種攻撃パターンの生成と帯域限界試験 ワイヤースピードでのパケットを生成とキャプチャ
		ソフトウェア制御・処理 パケット生成ツール	・PCへのパケット生成ツールインストール
セントラルゲートウェイ LAN側の接続		Securedバス セキュリティ機能あり	・セントラルゲートウェイによる セキュリティ処理を実行したバス
		Rawバス セキュリティ機能をバイパス	・セントラルゲートウェイによる セキュリティ処理をバイパスしたバス

本項で内部ブロック図など詳細を示す本テストベッドは、各種の攻撃から車内ネットワークを守るインテリジェンス諸機能をセントラルゲートウェイに実装する。ただし、個別 ECU 内部での対策や、上位 WAN 側での対策は、それぞれ CAN や Ether 通信回線を用いて接続することで、総合的なセキュリティ評価が可能である。また、セントラルゲートウェイでは IP ネットワークセキュリティの基本となるアドレスベースでの多層監視と経時監視に加えて、高位レイヤやネームベースでの監視を可能とする。

(1) 全体構成

図 3.2a.3-1 の全体構成で示す通り、本テストベッドは、セキュリティ機能を実装するセントラルゲートウェイを中心として､左側は車外 WAN 側インタフェース､右側は車内 LAN 側インタフェースを有す。

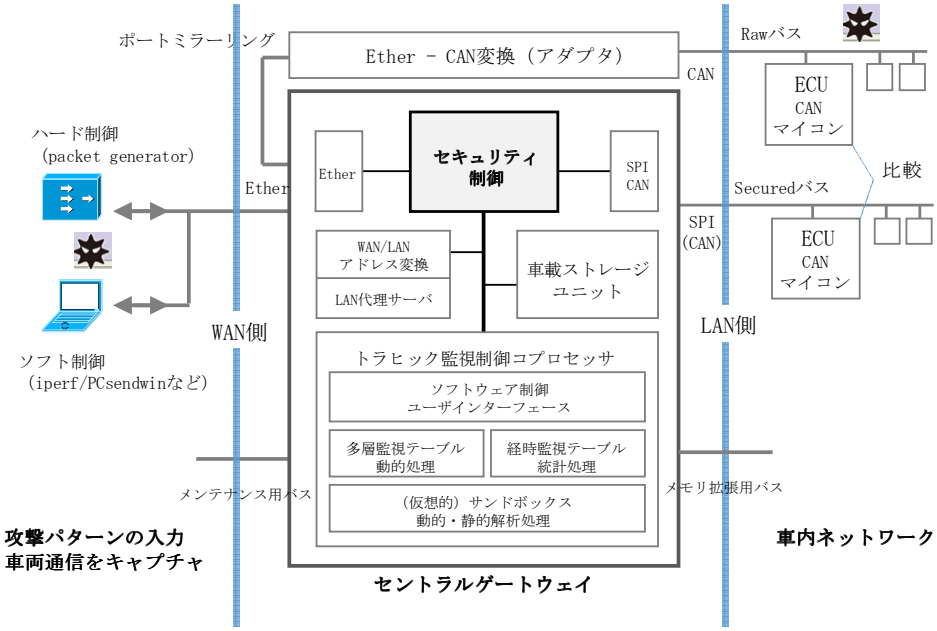


図 3.2a.3-1　テストベッドの全体構成

以下、個々のコンポーネントについて説明する。

① 車外 WAN 側インタフェース

図 3.2a.3-1 において、左側の WAN 側インタフェースでは、外部の IP ネットワークに接続するため Ethernet 通信回線を使用する。攻撃パターンを含む Ethernet パケットを入力することで、WAN 側より各種の入力と評価を可能とする。そのため WAN 側では、次に示すソフトウェア制御、ハードウェア制御、双方の接続が可能とする。

(ⅰ) ソフトウェア接続と制御

PC 上でネットワーク測定ツール iperf（https://iperf.fr/）やパケットジェネレータ ipsendwin などのソフトウェアを利用することで、情報搾取を目的とするなど攻撃用のテスト IP パケットを生成することが可能である。例えば標的とする車両や特定する ECU の情報を IP（L3），MAC（L2）で定義し、アプリケーション層（L7）に攻撃パターンを実装し、WAN 側よりセントラルゲートウェイへ送信する。このようにソフトウェア制御の場合はテスト用パケットの生成と評価は PC 上で行えるが、攻撃パケットの生成と通信の速度性能を確保することが難しく、真に送信されたかは保証されない。従って、複数の連続する攻撃パターンを周期的に送信し観測するなどの評価については、次に記すハードウェア制御が必要となる。

(ⅱ) ハードウェア接続と制御

DDoS の実験など、大量のパケットを帯域限界まで生成して送信することで攻撃するなど、負荷を与えることが重要なテストにおいてハードウェア制御が必要となる。ここでは、一般に「パケットジェネレータ」と呼ばれる特殊な装置を用いることが必要である。

またこの WAN 側インタフェースでは、車内 LAN 側インタフェースから採取された車両の情報（例えば ECU における定期的な車両情報）を、セントラルゲートウェイを経由して外部と通信し監視する目的においても評価が可能である。さらに、遠隔操作などマルチホップやインターネット経由を想定し、ネットワークルータ、インターネットへの接続も可能とする。

② セントラルゲートウェイ

本テストベッドでは、車外からの侵入、脅威の監視からその対策までセントラルゲートウェイに実装する。次節の個別構成で詳細を記すが、多層監視、経時監視、パケットの廃棄処理からログの作成と登録など、複数の手段を実装し防御を行う。車内 CAN バスネットワークは車外 IP ネットワークと分離されており、例えば ECU がインターネットを経由して自動車の保全サービスと通信する場合などにおいては、セントラルゲートウェイが全ての ECU の代理サーバとなり、車内ネットワークへの直接的侵入から守る。

また車外 Ethernet から入力される各種攻撃パターンは、ポートをミラーリングし攻撃パターンの複写を行うことを特徴としている。この攻撃を Ethernet - CAN プロトコル変換し、CAN へ「Raw バス」として開放する。すなわち、本ゲートウェイにおいて、セキュリティ制御が施された「Secured バス」と対策されない「Raw バス」との比較実験し、評価することを可能としている。

Ethernet-CAN のプロトコル変換については、市販の変換アダプタを利用することが可能である。

③ 車内 LAN 側インタフェース

LAN 側は、CAN プロトコルで動作する。通信帯域は 500Kbps - 1Mbps 程度とする。本インタフェースに接続する CAN マイコンは、車載ネットワークにおける各種 ECU 接続を想定するワンパスであり、攻撃の標的と、悪意を持ったソフトウェアによる感染を ECU において処理する実験評価も可能としている。通常 CAN マイコンは、CAN のほか 1Mbps 程度で動作する SPI（System Packet Interface）インタフェースを具備するため、FPGA などで試作するセントラルゲートウェイとは、SPI で接続することで、より拡張性のある評価が可能である。

(2) 個別構成

①　セントラルゲートウェイ実装ユニット

本項では、想定される車外通信による脅威から車載ネットワークを守るため、セントラルゲートウェイで実装すべき機能を個別に説明する。前項(1)の全体構成で示した図 3.2a.3-1 のセントラルゲートウェイに実装される各ユニットについて、機能を説明する。

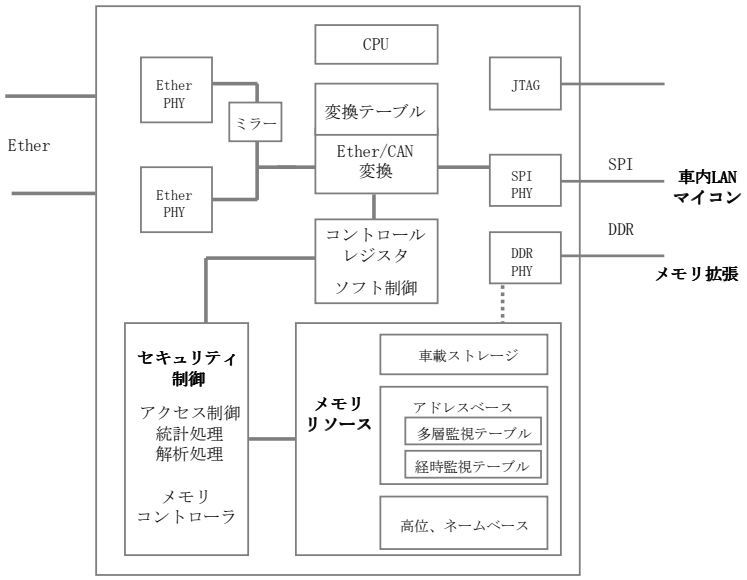


図 3.2a.3-2　FPGA へのセントラルゲートウェイ実装ユニット

図 3.2a.3-2 に示すとおり、本テストベッドにおいて、セントラルゲートウェイでのセキュリティユニットは、FPGA（または SoC）に実装するものとする。セントラルゲートウェイ実装ユニットの外部インタフェースは、1Gbps 程度の帯域性能を有する Ethernet 通信回線による車外 WAN 側インタフェース、CAN もしくは SPI プロトコルによる車内 LAN 側インタフェース、そして FPGA 内部のメモリリソース拡張用の DDR インタフェースである（メンテナンス用 JTAG は記さず）。以下、図 3.2a.3-2 に示した内部の各ユニットの機能について説明する。

(ⅰ) 車外ネットワークインタフェース

最大 1Gbps の Ethernet 通信回線とし、前節で示したソフトウェア制御、ハードウェア制御のほか、ネットワークルータに接続し、マルチホップ評価も可能とする。また、ゲートウェイ内で処理されたセキュリティ機能の効果を評価するため、Ethernet 入力をミラーリングした Ethernet 出力ポートを兼ね備える。

(ⅱ) 車内ネットワークインタフェース

最大 1Mbps の通信速度をもつ SPI インタフェースで、後述の CAN マイコンを用いたワンパスとの通信を行う。CAN は実効データ転送速度 500kbps 程度であるため、SPI で代用することが可能であり、これにより CAN マイコンとの接続通信を拡張するほか実装や評価が安価で容易となる。

(ⅲ) メモリリソースとコントローラ

FPGA（または SoC）上のメモリリソースとカスタム設計によるメモリコントローラによって以下の機能を実現する。本テストベッドにおいて、ゲートウェイに搭載すべきセキュリティ機能は、従来 IP ネットワークにおいて主流となるアドレスベースの監視と制御に加えて、高位レイヤとネームベースの監視と制御を有することを特徴のひとつとしている。

a) アドレスベースの監視と制御

IP ネットワークにおいては、ファイアウォールに実装されるアクセスコントロールリスト（ACL）によるセキュリティ機能に相当する。車外との通信に対しては多層監視を行う。監視と検知において、到着パケットの廃棄など処理について即応性が要求されるものは、本制御により動的に処理されるものとする。また動的処理にかかる制御ポリシーは、コントロールレジスタによりソフト的に書換えが可能とする。例えば L2、L3、L4 の所定の組合せによって定義されたフロー情報と一致したパケットが WAN 側インタフェースに到着した際に、これを攻撃パケットと判断し LAN 側に送信せず廃棄する場合がある。フロー情報は、次節で示す監視・制御内の多層監視用テーブルにブラックリストとして登録され、廃棄する、許可する、蓄積するといった動的に扱う処理は外部からソフトウェアによって変更が可能とする。多層監視では、本テストベッドのユーザとなるセキュリティに関わる研究開発者が以下の実験と評価を行うことが可能である。

・L2 監視（IP ネットワーク MAC アドレスフィルタリングに相当）
・L3、L4 監視（IP ネットワークファイアウォールのアクセス制御に相当）
・さらなる高位監視

次に、セントラルゲートウェイ実装ユニットにおいて、FPGA さらに FPGA に接続するメモリによって実現する LUT(ルックアップテーブル)について説明する。LUT は、多層監視用 LUT と経時監視用 LUT（双方を両立する仕様も可能）を実装するものとする。

b) 多層監視用 LUT

図 3.2a.3-3 で示す多層監視用 LUT では、Ethernet パケットにおいて L2、L3 さらに L4-L7 までの多層の組合せで構成される情報をひとつのフロー情報としてまとめ、フロー単位に管理し、各々のフローに対して個別のルールを与える。Ethernet パケット到着時、パケットヘッダーに記された L2 - L4、さらに目的によっては L7 情報までを本多層監視用 LUT に登録されたフロー情報と比較参照することで動的な多層監視を可能とする。多層フロー情報の参照において一致したフローについて、LUT にあらかじめ登録されたルールの引き出しを行う。ルールは各フロー単位に対して独立に付与されており、前節と同様に許可/拒否などユーザはソフトウェアによって随時書換えが可能である。

IP ネットワークにおいては、ファイアウォールに実装されるアクセス制御とよばれるセキュリティがあるが、本テストベッドが提供する LUT では、次の 2 つの方法のいずれもをソフトウェアによる制御が可能とする。

・ブラックリスト方式
　拒否するリスト。これに一致するパケットは即座に廃棄処理し、到着時刻等をログ管理する。
・ホワイトリスト方式
　許可するリスト。これに一致するパケットを LAN 側へ送信する。ゲートウェイを中継し、LAN 側の ECU への送信を要求する通信については、ホワイトリスト方式が多用されるものと考えられる。

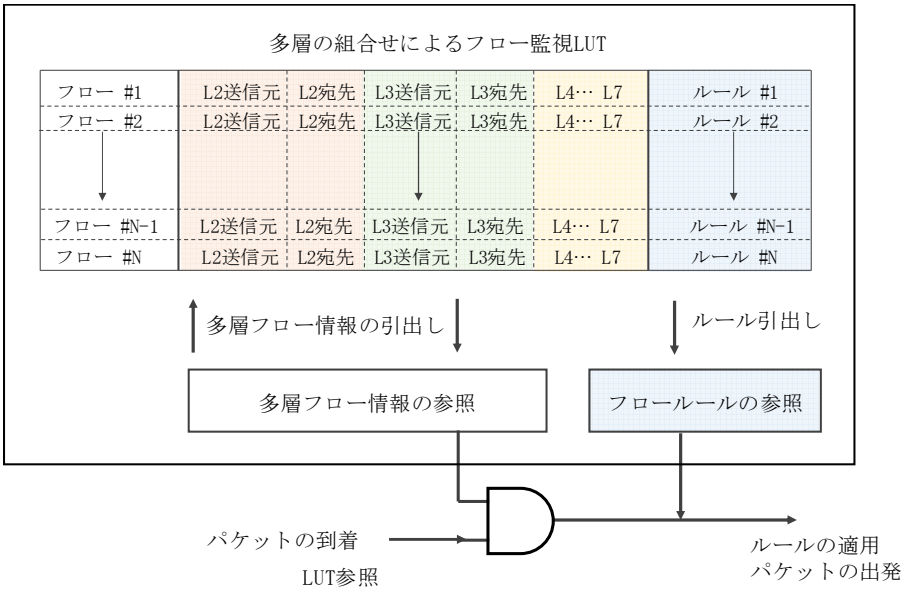


図 3.2a.3-3　多層監視用 LUT

c) 経時監視用 LUT

図 3.2a.3-4 に示す経時監視用 LUT では、ユーザによって定義される時間（タイムスロット）単位で到着/出発するパケットトラフィックの情報を監視しつつ、セキュリティ制御を行う。例えばパケット到着時に、パケットヘッダー情報を経時監視用 LUT に登録された多層フロー情報と参照し、一致したフローについて該当するタイムスロットでのトラフィック情報の更新を行う。これにより、タイムスロット単位で平均的なパケット到着数を知ることが可能になる。あるタイムスロットにおいて、徐々にあるいは急激に到着数が増加する傾向を知ることができれば、DDoS 攻撃による標的となっている可能性があるため、深刻な被害が発生する前にその発信者からのトラフィックを遮断する。タイムスロット、フロー、ルールに関しては、前節と同様にソフトウェアによってユーザからの書換えが可能とする。さらに複数のタイムスロットによって蓄積された情報を、WAN 側インタフェースへ転送する。これはクラウドなどさらに上位における管理や監視の実験・評価を目的とするものであり、昨今 IP ネットワークにおいて注目される SDN（Software Defined Network）の概念を評価の一つとして導入するものである。

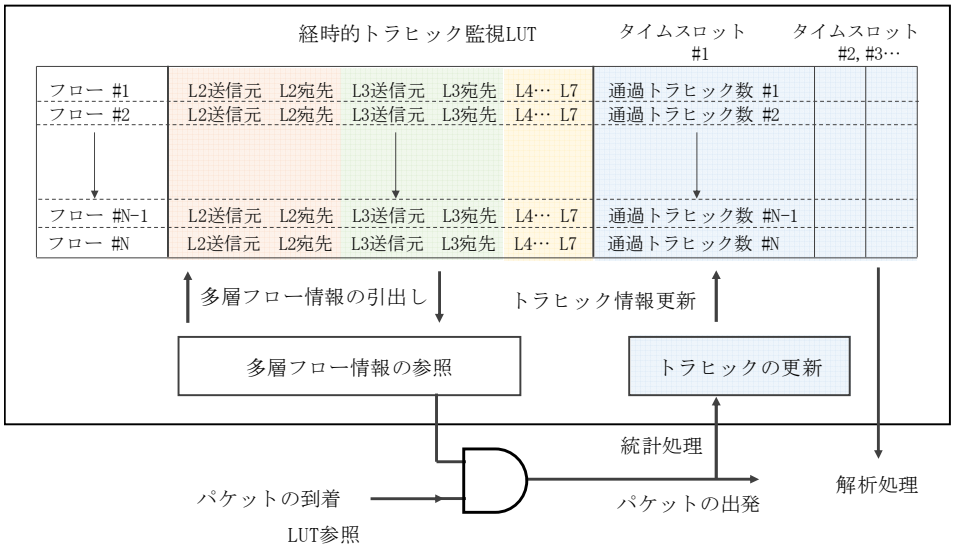


図 3.2a.3-4　経時監視用 LUT

d) ネームベースの監視と制御

近年 IP ネットワークにおいて、アドレスをベースとした経路制御やセキュリティの限界から、ネームベースでの制御や処理について研究開発が進められている。

（参照）NDN（Named Data Networking）プロジェクト：https://named-data.net/

IP ネットワークにおいてプロトコルの基本となる IP アドレスや MAC アドレスには、元来意味が存在しない。そのため前節で示したファイアウォールでのアクセスコントロールリストのように、許可/拒否という意味をアドレスまたは複数のアドレスを多層に組み合わせたフローに対して逐次与えることでセキュリティの管理を行っている。しかしながら、意味を持たないアドレス情報と許可/拒否を接続する情報に対しての搾取や改ざんを想定した場合、アドレス情報をベースとしたセキュリティはもはや全く機能しなくなる。

ネームベースとは、このような背景を鑑みて IP ネットワークにおいて新しく提案された技術であり、名前・コンテンツそのものを用いて、例えば高度なセキュリティの制御を行うものである。従来のセキュリティ技術のひとつである L2 - 4 以降の高位レイヤを用いた DPI（Deep Packet Inspection）では、パケットペイロードに潜むウィルスの検知等で用いられ高い効力を有するセキュリティ技術であるが、ネームベースでは本来通信すべきでない相手との通信を監視するなど、さらに幅広い拡張機能の実装と評価が可能である。

e) 車載ストレージ

セントラルゲートウェイに実装される車載ストレージには様々な応用が考えられるが、ここでは特に用途の限定はしない。本テストベッドを用いて評価を行う研究者が、ログの管理など様々な目的で自由に利用できる空間としてメモリ資源を提供する。

以上示した多目的用途によって、メモリリソースが枯渇するケースも考えられる。そのため FPGA では、DDR インタフェースにより外部メモリの拡張利用を可能とする。

(ⅳ) Ethernet–CAN 変換

Ethernet パケットを車内 LAN 向けのフレームに変換するユニットを図 3.2a.3-5 に示す。Ethernet パケットの L3 以降のレイヤは CAN に存在しない。そのため例えば TCP によるステートフル通信は車内 LAN で処理できないため、セントラルゲートウェイで完了しなければならない。さらに IP アドレスや MAC アドレスが CAN 接続の ECU を特定する通信の場合、これらを CAN フレーム内の ID に変換することで、通信を確立する評価環境が必要である。

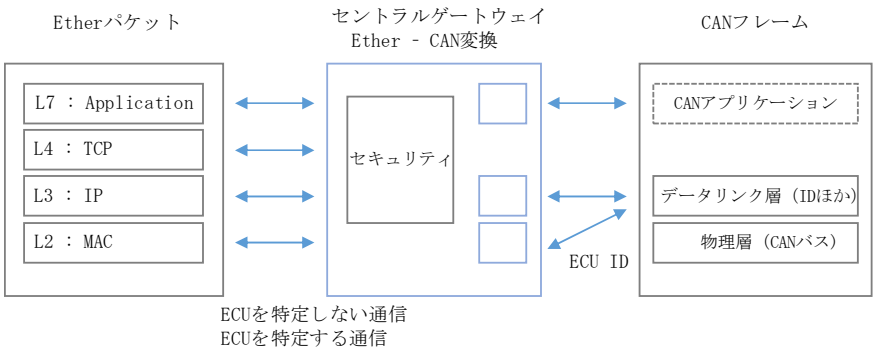


図 3.2a.3-5　Ethernet-CAN フレーム変換

また、セントラルゲートウェイは IP ネットワークにおける NAT（NAPT）の機能を実現する変換テーブルを実装しており、CAN 接続の ECU からセントラルゲートウェイを介して車外へ通信する場合は代理サーバとして機能する。ECU から送信される CAN フレームの ID をもとに IP アドレスと MAC アドレスを導出して車外へ通信を行い、車外からの応答パケットはセントラルゲートウェイに割り当てられた IP アドレスで受ける。従って、応答パケットに攻撃者からのメッセージが含まれる場合でも、セントラルゲートウェイで遮断し、車内 LAN への侵入を防止することが可能となる。

図 3.2a.3-6 は Ethernet パケット（L2 フレームのみ記載）から CAN フレームへの変換の具体例を示している。宛先 MAC アドレスと送信元 MAC アドレスは CAN 内部で特定される ECU を示すべきものであるため、この例では CAN の ID として変換し CAN フレームを生成する。同時に MAC と ID 間の接続情報を変換テーブル内に登録する。

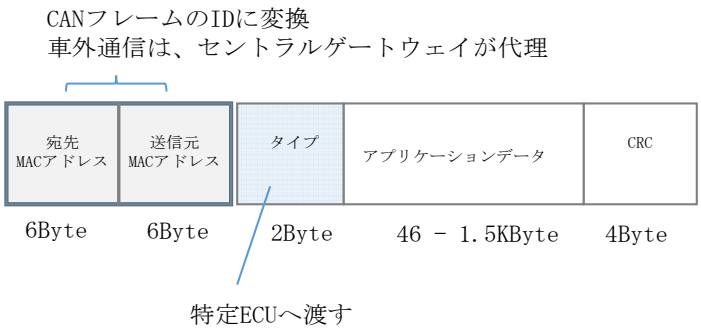


図 3.2a.3-6　フレーム変換

ここで示した Ethernet フレームは、WAN 側に接続するソフトウェア制御、ハードウェア制御のどちらからも生成し、セントラルゲートウェイに向かって送信することが可能であるが、様々な攻撃パターンの生成目的においてはソフトウェア制御、DDoS 攻撃など大量のパケットを連続するなど負荷が重要な評価目的においてはハードウェア制御が向いている。

② ワンパスユニット

図 3.2a.3-7 に示すワンパスユニットは、セントラルゲートウェイ実装ユニットと SPI インタフェース（あるいは CAN インタフェース）で接続し、車内 LAN 側の ECU ワンパスモデルを提供する。ECU においても、フレームの ID 照合などセキュリティ機能を実装し評価することが可能である。本機能は CAN マイコンを用いるもので、CPU 及びメモリを内蔵しソフトウェアによる書換えが可能である。またメンテナンス目的に JTAG インタフェースを有する。

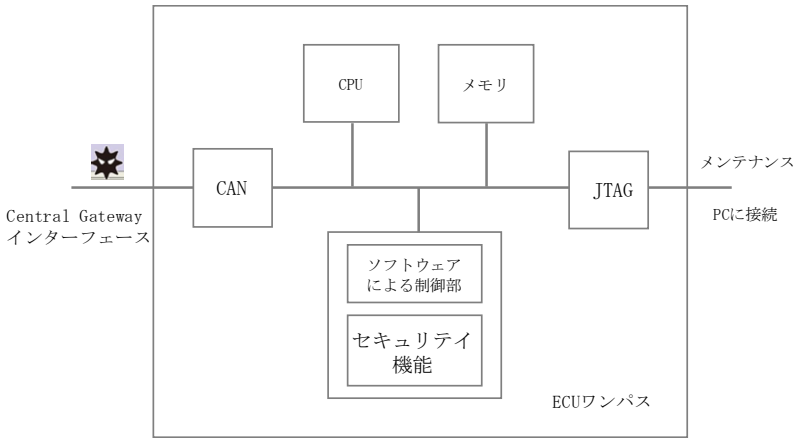


図 3.2a.3-7　ワンパスユニット

3.2a.4　テストベッドを用いた評価

ここでは、テストベッドを用いた攻撃パターンや評価について検討した。

(1) テストベッドで評価が可能な攻撃と検知

表 3.2a.4-1 は、本テストベッドによって評価が可能な攻撃と評価の例を示している。

表 3.2a.4-1　テストベッドを用いた評価

項目	内容	侵入経路 OBD　撃 の検知方法	侵入経路 車外 Ethernet	撃の検知方法
デバッグインタフェースから ECU 制御部への不正アクセス	LAN 側からの侵入と操作を評価する。	✔		OBD からの侵入。ECU、CAN マイコンで検知する。
車外ネットワークから ECU 制御部への不正アクセス	WAN 側からの侵入と操作を評価する。		✔	Ethernet パケットによる侵入。セントラルゲートウェイで監視し、処理に係る制御ポリシーを与える。さらにログを登録する。
デバッグインタフェースからのバッファオーバフロー攻撃	LAN 側から侵入するフレーム流量を評価する。	✔		OBD からの侵入。ECU、CAN マイコンで検知する。
車外ネットワークからのバッファオーバフロー攻撃	WAN 側から侵入するパケット流量を評価する。		✔	Ethernet パケットによる侵入。セントラルゲートウェイで監視し、処理に係る制御ポリシーを与える。さらにログを登録する。
標的型の繰り返し攻撃	偽サーバへの誘導監視や DDoS など標的型攻撃を評価する。		✔	経時監視フローテーブルにトラフィック情報を登録し、動的監視と廃棄処理を行う。
バックドア攻撃	予見が困難な実装デバイスからの不正アクセスを評価する	✔	✔	セントラルゲートウェイで監視。ネームベースなどを利用し、トラフィックフローが通常と異なる場合を検知する。
デバッグインタフェースからのファジング評価	ブルートフォース攻撃パターンの生成	✔		
外部インタフェースからのファジング評価	ブルートフォース攻撃パターンの生成		✔	

(2) 各攻撃パターンにおいて評価すべき内容

① 車載ネットワークへの不正アクセス

本項で示す攻撃とは、標的型の連続攻撃と異なり一度の攻撃で攻撃者の目的を完了するものとし攻撃に対する即応性を評価する。例えばハードウェアとして正常に動作する ECU に対して、攻撃者は不正なソフトウェアをリプログラミングする。偽ソフトウェアを書き込まれた ECU は、機能の停止や不正動作、さらには情報の漏えい・搾取などに陥る。早期復帰などの対処は事後対策として重要な措置であるが、ここでは不正パケットを即座に破棄する対策手法を講じるものとする。従来、OBD ポート接続や不正 ECU 接続によって事前に評価すべき内容は報告されているが、新たに車外との通信による攻撃が追加される。セントラルゲートウェイにおいては、多層監視により LUT に登録されたリストとの照合によって不正アクセスを検知し、即座にパケットの廃棄処理を行うことが可能である。

また近年、マルウェア数の増加から拒否を登録するブラックリストではなく、許可を登録するホワイトリスト型のアクセス制御方式が注目されている。本テストベッドにおいても、許可を与えるフローを事前に LUT に登録し、さらにメンテナンスを行いつつ、到着パケットの動的監視とホワイトリストとの照合により、より高いレベルでのセキュリティをテスト・評価することが可能である。

② 標的型による繰り返し攻撃

近年のサイバー攻撃において見られる情報搾取や DDoS 攻撃による標的は、そのほとんどが繰り返し型の攻撃によって攻撃者の意図を達成するものである。昨今の DDoS 攻撃は高度で複雑であり、一度や二度のトラフィック解析では後に自身が標的となることの検知は非常に困難である。

そのため高度な DDoS 攻撃を防御するためには、CAN プロトコルに変換される前に検知しなければならない。通常の検知方法である SPI（Stateful Packet Inspection）とは、車載ネットワークに限らずトラフィックの監視とセキュリティ制御を行うためには、「どこに」「何を」「どのように」通信しようとしているのかを把握し識別することである。ここで「どこに」とは L2、L3 に該当し、「何を」「どのように」は L4 を意味する。ファイアウォールにおけるアクセス制御による防衛方式は通常 L4 までを監視する。しかし昨今の高度な攻撃に対して識別機能の強化が必要となり、DPI と呼ばれる SPI に代わる技術が有望視されている。ここではさらに高位の L5 - L7 レイヤを監視する。同様にアドレスではないネームベースにおいても監視を行う。OSI 参照モデルでは L5 より高位を、TCP/IP 参照モデルではアプリケーション層を参照してトラフィックの制御を行う。繰り返し型攻撃は早い段階で検知することが重要であり、本テストベッドは様々な攻撃と対策手法について評価が可能な環境を提供する。

③ CAN バスバッファオーバフロー

現在、CAN バスに接続する最大ノード数は、16 個である。これは CAN がマルチマスタのイベント駆動型の通信方式を採っているためであり、平均トラフィックの 30% ~ 40%の範囲でネットワーク制御を行わないと通信障害が発生する可能性が高い。従来から指摘される CAN の脆弱性のひとつであり、攻撃者からのデバッグインタフェースを用いた攻撃のほか、車外ネットワークから大量のトラフィックを侵入させる攻撃が想定される。本テストベッドでは、トラフィック流量の経時監視から、例えば閾値となる脅威レベルを予め設定し、流量の急激な変化を逸早く検知することでバッファオーバフロー攻撃を予期し、ログ登録とパケットの解析を行うことが可能である。攻撃と判断した際には、さらにネットワークの切断や送信元を特定するといった評価も可能である。

3.2a.2 から 3.2a.4 において示したように、本テストベッドは「点」ではなく「線、面」で守る総合型・継続型の評価環境をユーザに提供する。

- 物理レイヤからトランスポートレイヤ、さらにはアプリケーションレイヤまで多層においてトラフィックの監視とセキュリティ制御をユーザに提供する。
- パケット到着時のみではなく、タイムスロットとしてプログラムする時間間隔においてトラフィックの監視とセキュリティ制御をユーザに提供する。
- 本テストベッドは、単体で完結するものではなく、様々な拡張性を有している。例えば、ECU におけるセキュリティ評価基盤や Ethernet における評価基盤など他の研究資産と接続するインタフェースを具備しており、さらなる拡張により統合的評価が可能である。

3.2a.5　評価環境（車両模擬システム）の構築

ここまで、テストベッドの要件およびテストベッドを用いて行うことの出来る評価について検討してきた。ここでは、テストベッドにおける評価対象となる部分であるセキュリティ評価用車両模擬システムの構築について述べる。セキュリティ評価用車両模擬システムは、その主要部分となるセントラルゲートウェイと、そのゲートウェイと通信を行うユニットで構成される。

(1) セントラルゲートウェイの仕様

セキュリティ評価用の車両模擬システムに使用するセントラルゲートウェイ（CGW）は、ルネサス製“RH850F1L”を搭載しているが、セキュアコア搭載の“RH850F1H”も搭載できるように､以下のような入出力ポートと主要周辺デバイス構成､および回路構成とした。作成したセントラルゲートウェイボードの外観を図 3.2a.5-1 に示す。

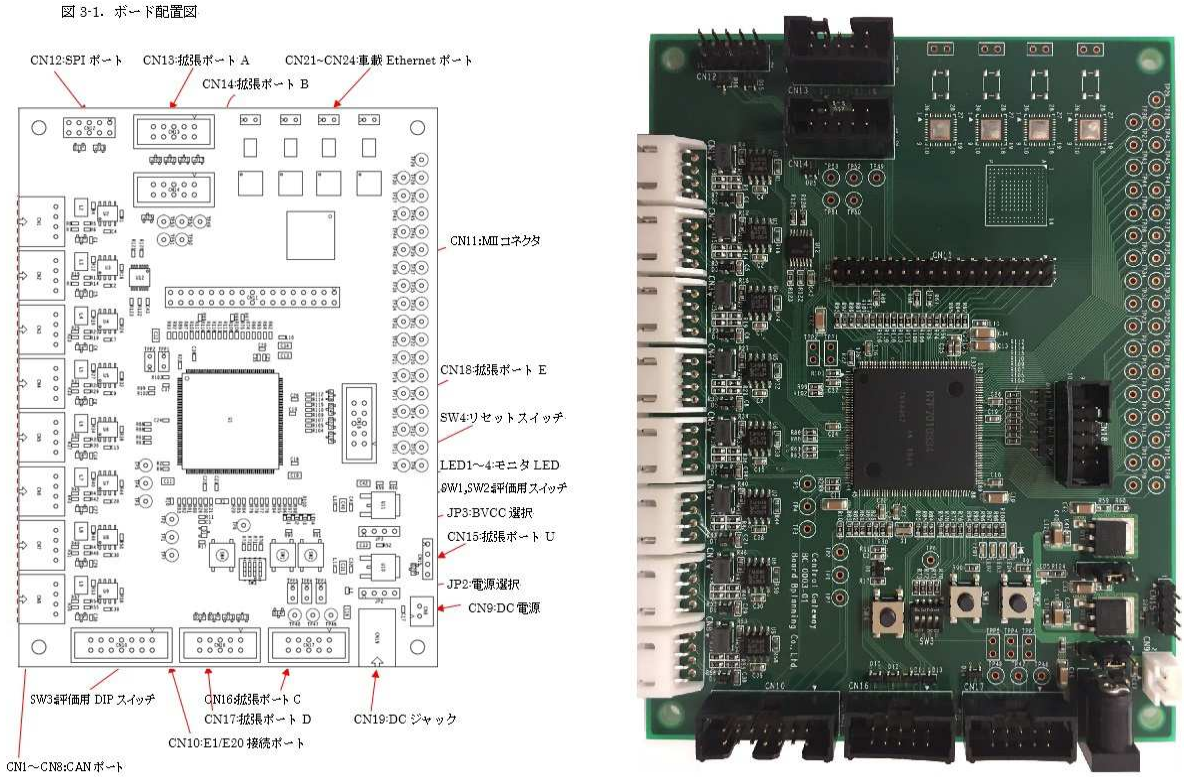


図 3.2a.5-1　セントラルゲートウェイ外観図

本セントラルゲートウェイボードは、以下の入出力ポートを実装している。

① 通信評価対象ポート

・CAN FD	6 ポート (RH850/F1L 搭載時は無し)
・CAN	2 ポート (RH850/F1L 搭載時は 6 ポート)
・車載用 Ethernet	4 ポート

② 評価補助用ポート

・SPI ポート	セキュリティ機能拡張用
・拡張ポート U	ホスト PC 接続用
・拡張ポート A, B	3.3V IO 信号拡張用
・拡張ポート C, D, E	5V IO 信号拡張用
・E1/E20 接続ポート	プログラム書込み，プログラム・デバッグ用

(2) 車両模擬システムの構成

今回構築した車両模擬システム（図 3.2a.5-2）は、セントラルゲートウェイとそれに接続される車両外部からの入力としての TCU、疑似舵角データを発生する舵角データ発生ユニット、およびセントラルゲートウェイを介して舵角データを受信する操舵系ユニットで構成する。また、セントラルゲートウェイと操舵系ユニットのそれぞれに、通信を暗号化

するための TPM（Trusted Platform Module）評価ボードを接続している。

・CGW 評価ボード：ルネサス製マイコン RH850F1L 搭載
・操舵系ユニット：車載マイコン評価ボード：ルネサス製マイコン RH850F1L 搭載
・TPM 評価ボード：STM 社製 TPM2.0 対応 TPM 評価ボード
・TCU：模擬的な車外通信ユニット、スマホ等と接続可能
・舵角データ発生ユニット：車両模擬システムに舵角データを供給するユニット

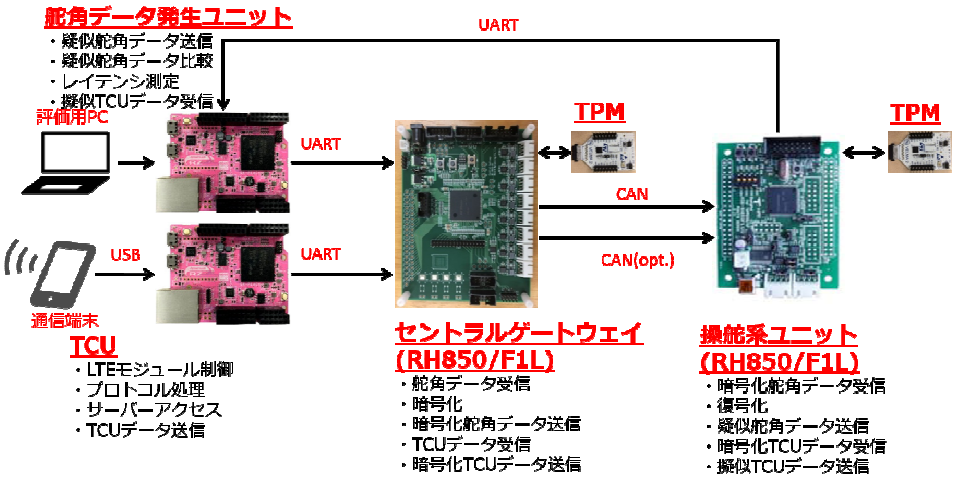


図 3.2a.5-2　車両模擬システム構成図

また、上記車両模擬システムに搭載したセキュリティの車両システム向けソフトウェアは以下の通りである。

・セキュリティミドルウェア：
　－車両通信(CAN 等)向けセキュリティ機能の実装
　－TPM 用ミドルウェアの車載マイコン向け対応
・アプリケーション：
　－セキュリティを考慮した各機器向けアプリケーション

以下に、舵角データ発生ユニット（図 3.2a.5-3）と TCU ユニット（図 3.2a.5-4）のシステム構成図を示す。

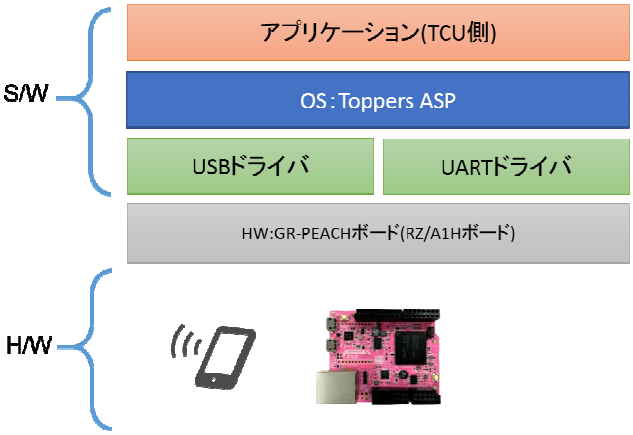


図 3.2a.5-3　舵角データ発生ユニットシステム構成図

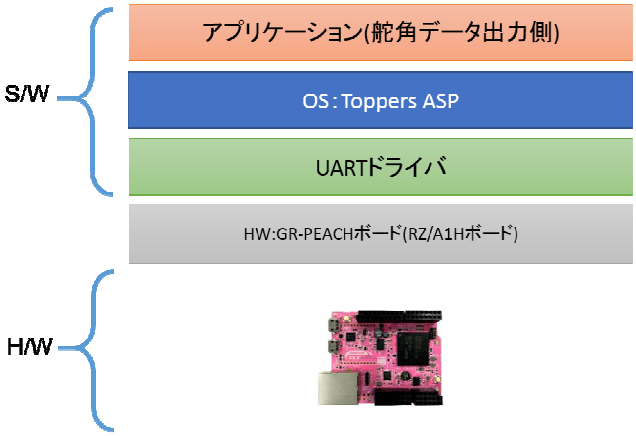


図 3.2a.5-4　TCU ユニットシステム構成図

3.2a.6　評価環境（車両模擬システム）を用いた攻撃評価の試行

3.2a.5 で構築した車両模擬システムに対して、攻撃評価として脅威注入（Fuzzing）テストを試行した。

(1) Fuzzing テスト環境

セントラルゲートウェイボード（CGW ボード：BC-0003-01）上に実装された暗号化 CAN 通信ソフトウェアに対し、Fuzzing テストを試みた。Fuzzing テストでは、CGW ボード上の CAN ポートより Fuzzing データを入力し、暗号化 CAN 通信ソフトウェアの挙動を確認する。Fuzzing 評価に用いたテスト環境を図 3.2a.6-1 に示す。

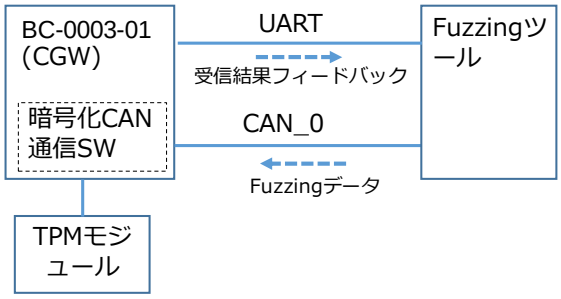


図 3.2a.6-1　Fuzzing テスト環境

今回、Fuzzing データは暗号化された CAN bus より入力した。暗号化 CAN bus における CAN フレームのフォーマットを、図 3.2a.6-2 に示す。

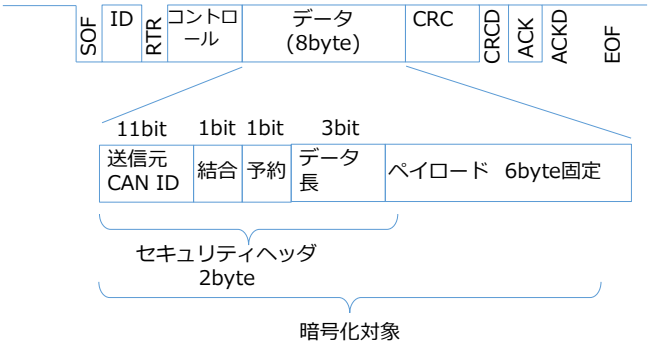


図 3.2a.6-2　暗号化 CANbus 通信フレームフォーマット

暗号化 CAN bus 通信では、通常の CAN フレームのフォーマットのデータ部分に暗号化されたセキュリティヘッダとデータペイロードが搭載される。セキュリティヘッダは、暗号化 CAN 通信のために送信側（暗号化側）で付加され、受信側（復号化側）の暗号化 CAN 通信ソフトウェアはセキュリティヘッダの情報を元に暗号化前の平文を復元する。今回の Fuzzing テストでは、暗号化 CAN 通信ソフトウェアのセキュリティヘッダ領域を Fuzzing の対象とした。

暗号化 CAN 通信では、セキュリティヘッダとペイロード部分を 1 ブロックとして暗号化されるため、セキュリティヘッダやセキュリティヘッダ内の特定の領域に対し Fuzzing テストを実施するためには、Fuzzing ツールは平文のテストベクタを暗号化 CAN bus の暗号鍵で暗号化しテスト対象に入力する必要がある。

しかし、今回 CGW ボードに実装した暗号化 CAN bus 通信の仕様は独自に設定したものであり、市販の Fuzzing ツールは今回実装した暗号鍵の配信プロセスや暗号化プロセスをサポートしていない。そこで、暗号化されたデータ部全体を Fuzzing 対象としてテストすることとした。

データ部分全体を対象とした Fuzzing データは、暗号化 CAN 通信ソフトウェアにより復号化され、復号化された結果得られるセキュリティヘッダに対してフレームの正当性等のチェック処理が実施され、正当と判断されたフレームに対しては平文の復元処理が実施さ

れる。今回の試験では、これらの正当性チェック及び平文の復元処理を対象とし、Fuzzing データの入力に対する応答を確認する。Fuzzing テストの流れを図 3.2a.6-3 に示す。

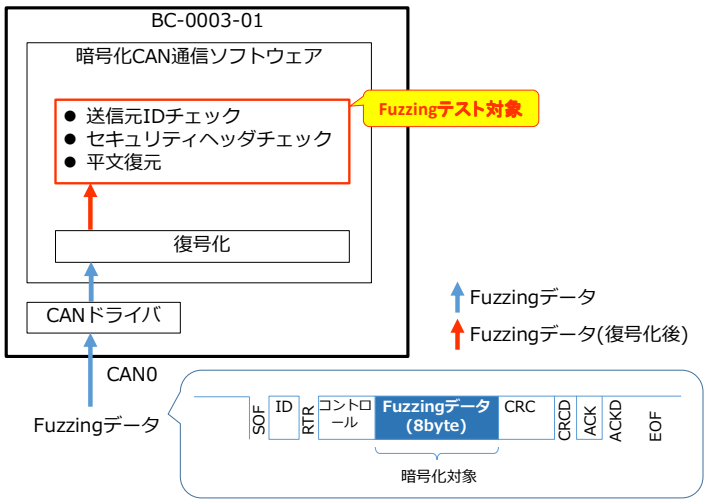


図 3.2a.6-3　Fuzzing データの流れと Fuzzing テスト対象

① 暗号化 CAN 通信ソフトウェアの受信処理概要

ここでは、Fuzzing テストの対象となる暗号化 CAN 通信ソフトウェアの受信処理の概要について述べる。

暗号化 CAN 通信は、従来平文で通信されていた最大 8byte の CAN データを暗号化して送受信することを目的としており、従来の平文データを図 3.2a.6-4 のように暗号化 CAN 通信のフレームに再構成し通信を行う。

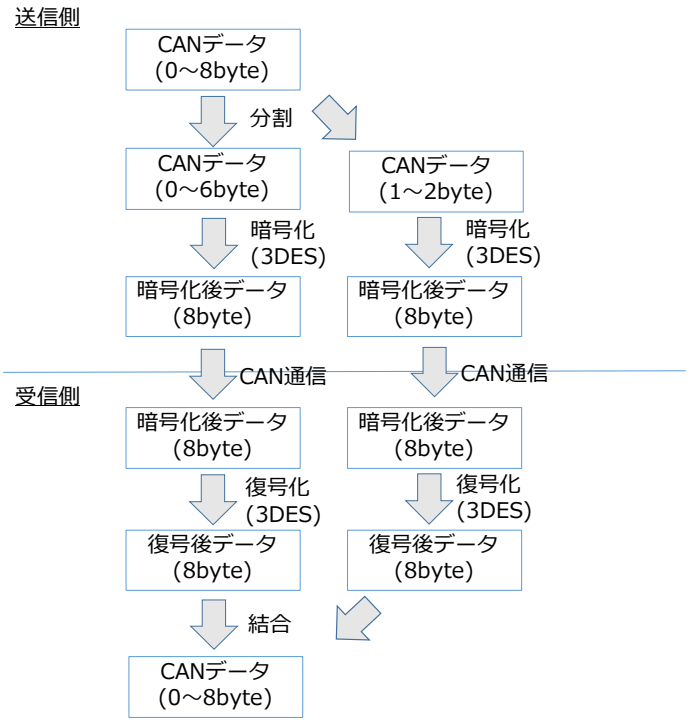


図 3.2a.6-4　暗号化 CAN 通信の暗号化/復号化過程

暗号化して CAN で送受信する際、平文の CAN データは 2 パケットに分割されて送信されるケースがある。これは暗号アルゴリズム 3DES（Triple Data Encryption Algorithm）での暗号化に際して 3DES のブロック長である 8byte のデータ長が最低限必要なためである。8byte 未満のデータを暗号化する際は、不足分にパディングを付加し暗号化する。この際、パディングを除く有効データ長を復号側に伝達する必要があり、少なくともデータ長のビット数分データが長くなる。そのため、平文で 8byte のデータは暗号化後のデータとして 8byte で送信できなくなることから、平文で 8byte の CAN データは、暗号化時 2 フレームに分割されて送信される。

本システムでは、暗号化前の有効データ長の情報および 2 フレームに分割された暗号フレームを結合するための情報は 2byte にまとめられ、各 CAN フレームにセキュリティヘッダとして付加される。結果として、1 つの暗号化後の CAN フレームで送信可能な平分データサイズは最大 6byte になる。

暗号化 CAN フレーム受信時、暗号化 CAN 通信ソフトウェアは以下の手順に従い暗号化 CAN データを受信し平文を復元する。

・8byte の CAN 暗号化データを 3DES で復号化し、セキュリティヘッダとデータを取り出す。
・送信元の CAN ID について、送信元 CAN ID のフィルタテーブルの登録 ID と比較し、登録されていない場合は、受信フレームを破棄。
・セキュリティヘッダの値についてエラーチェックを実施し、不正な設定値を含む受信フレームについて破棄する。

Fuzzing テストでは、8byte の Fuzzing データがデータ部分に挿入された CAN フレームを受信し、Fuzzing データは復号化された上で上記受信処理に入力される。今回の Fuzzing テスト環境における、入力されたデータパターンに対する正常・異常判定の条件を表 3.2a.6-1 に示す。

表 3.2a.6-1　暗号化 CAN 通信ソフトウェアにおけるフレームの正当性判定条件

送信元 CAN_ID	結合 Bit	データ長	結合待ち状態	受信判定
0x100,0x200	0	－	－	正常受信
	1	6	－	正常受信
		1～2	同一送信元から、結合 bit=1、データ長=6 のフレームを受信済み	正常受信
		1～2	同一送信元から、結合 bit=1、データ長=6 のフレームを受信していない	受信フレーム破棄
0x100,0x200 以外	－	－	－	受信フレーム破棄

“－”は Don’t Care を示す。

暗号化 CAN 通信ソフトウェアは、受信処理において結合待ちのフレームの有無に対する内部状態の管理や、送信元 CAN_ID ごとでの結合待ちフレームの領域管理を行う。今回の Fuzzing テストでは、これらの処理に対し、実装不備によりシステムをハングアップさせるような不具合（状態管理のストールや、領域アクセス違反によるメモリアクセスエラーの発生）が存在しないことを確認する。

今回の評価では、Fuzzing ツールは暗号化 CAN bus 通信の暗号鍵を知り得ないため、Fuzzing 入力データの正常値／異常値はコントロールできない。その為、Fuzzing ツールからみたテスト結果の振る舞いは、入力データがランダムに正常判定及び異常判定される結果となる。そこで、CAN bus からの Fuzzing データの入力後、正常受信もしくは異常受信の判定結果が返されるかどうかをチェックし、そのどちらかが返されない場合はシステムが異常状態に陥っていると判断し、テスト結果=Fail とする。

② Fuzzing テスト向け実装

Fuzzing テストでは、CAN ポート（CAN0）に異常データも含めたデータを入力し、その際の処理の正当性を確認する。暗号化 CAN 通信ソフトウェアは暗号化 CAN データ受信時にフレーム内のセキュリティヘッダの正当性を確認しエラーフレームを破棄する。Fuzzing テストではこのエラー処理判定についても確認されるため、判定結果のフィードバックが必要となる。この判定結果は、CGW ボードに搭載したスイッチの切り替えにより、UART ポートから以下のログとして出力するように設定可能である。

正常受信時：　“CAN RECV DATA : OK”を出力
異常受信時：　“CAN RECV DATA : NG”を出力

Fuzzing テスト向けの評価環境を図 3.2a.6-5 に、CGW ボードおよび周辺 ECU 等を接続した車両模擬システムを図 3.2a.6-6 に、評価環境全体を図 3.2a.6-7 に示す。

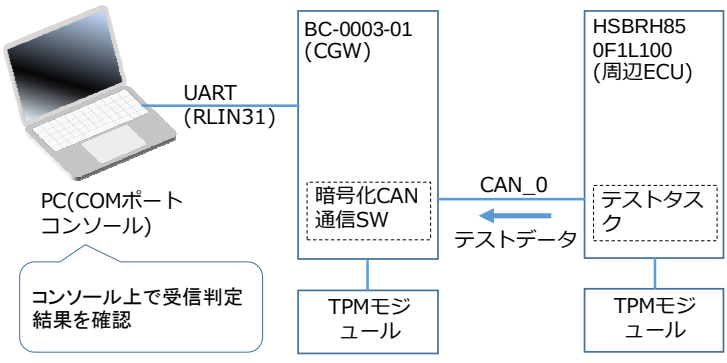


図 3.2a.6-5　Fuzzing テスト向け評価環境

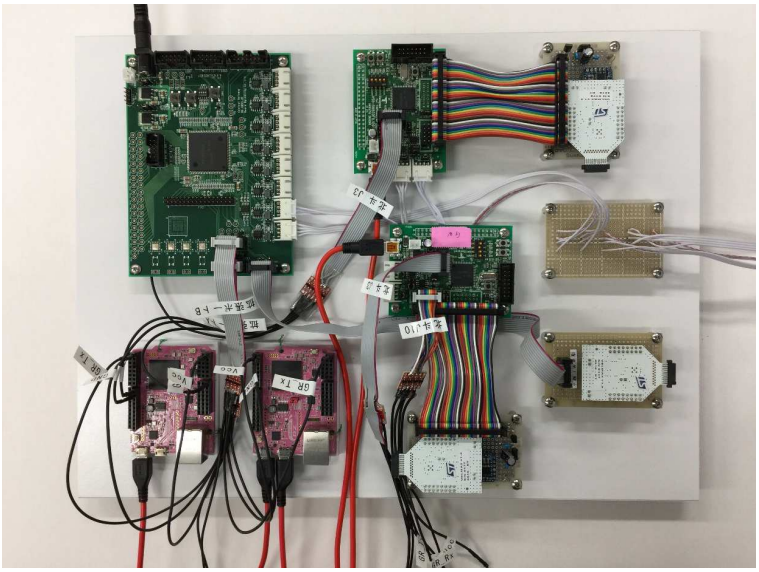


図 3.2a.6-6　Fuzzing テスト向け実装システム外観

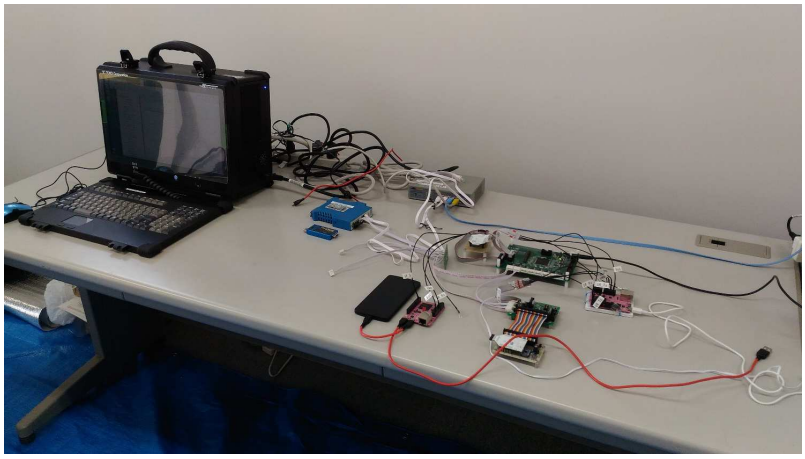


図 3.2a.6-7　Fuzzing テスト向け評価環境全体

(2) Defensics による Fuzzing テストの実行

ここでは、Synopsys 社の評価ツールである Defensics を利用して、暗号化 CAN bus 通信システムで利用している暗号化プロトコルに対する脆弱性検出テストを実施した。この暗号化プロトコルは本テーマ向けに独自に開発したものであるため、Defensics を含む Fuzzing ツールでは未実装のプロトコルとなっている。そこで、今回のテストでは、入力するデータフォーマットは図 3.2a.6-3 に準拠し、ペイロードのみを Fuzzing データとした。

今回評価したテスト対象システム（下記）に対して、今回用いたツールでは脆弱性は検出されないという結果となった。また、今回のテスト実施時に仮実装したメッセージ認証の有効性も確認できた。しかし、一部不正な CAN ID にも関わらず通常と同じゲートウェイ処理を実施したケースが見受けられたため、CAN ID 処理についてはより詳細な分析・検討が必要と考えられる。

① テスト対象システム

Fuzzing テストの対象は、図 3.2a.6-4 に示す CAN bus 通信の経路となる ECU、セントラルゲートウェイ（暗号化プロトコルを実装している場合には、Sec GW と称す）、操舵系ユニット（EPS）から構成され、セントラルゲートウェイ – 操舵系ユニット間で暗号化通信が行われる。

また、今回の評価対象では、通信の方向性も EPS → Sec GW → ECU の一方向のみであり、CAN raw frame を図 3.2a.6-8 に示す方向で送出するのみとなる。ファジングテストでは、Sec GW に不正な ECU が接続されたという想定の下、図 3.2a.6-9 に示すように Defensics を接続し、Sec GW の脆弱性評価を行うという方法をとった。

Sec GW は、CAN bus からの入力信号を暗号化通信とみなし復号化を試みる。復号化の結果、送信元 CAN ID でフィルタを実施し通信 OK/NG の判定を行う。（簡単な通信データの Integrity チェック）通信 OK/NG の結果は、シリアル経由で Defensics に送信する。

図 3.2a.6-8　評価対象となる CAN bus 通信システム

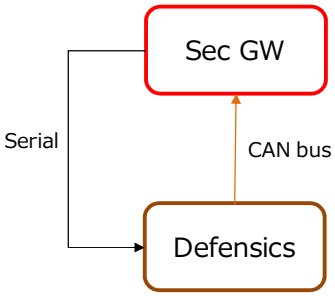


図 3.2a.6-9　セントラルゲートウェイと Defensics の接続方法

② テストプロトコル

暗号化 CAN bus 通信プロトコルは CAN bus raw frame と上位層の中間に位置するが、上位層のプロトコルがインプリされていない。そこで、暗号化 CAN bus 通信プロトコルの暗号化 CAN データを上位層とみなし、独自プロトコル can-proprietary-spec として定義する。Sec GW に対しては、CAN bus raw frame と can-proprietary の 2 種を用いて Fuzzing を実施する（図 3.2a.6-10、図 3.2a.6-11）。

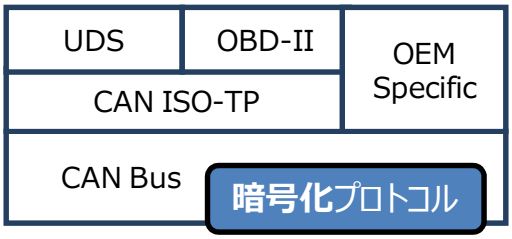


図 3.2a.6-10　暗号化 CAN bus 通信プロトコル

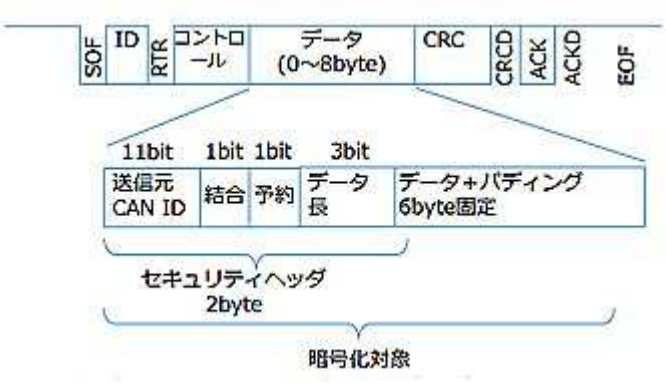


図 3.2a.6-11　暗号化 CAN データフォーマット

③ テスト結果

Sec GW に対する今回の評価では、表 3.2a.6-2 に示すとおり、60%以上のテストケースで Pass となった。

表 3.2a.6-2　テスト結果

	個数	コメント
テストケース総数	5,982	
Pass テストケース数	3,740	NG 返送があったケース
Pass ではないテストケース数	2,242	
CAN ID の不正入力テストケース数	2,706	CAN ID が不正なケース

テスト結果で Pass とならなかったケースについては、ほとんどが CAN ID が不正なケース（暗号化 CAN bus 通信プロトコルの CAN ID は 300h 固定）であり、問題ではない。

しかし、CAN ID 不正なケース数が Pass ではないテストケース数を上回るため、不正な CAN ID に対しても Sec GW が復号化処理を施したと考えられる。CAN ID の処理については、更に詳細な検討が必要である。

また、OK（復号化処理成功）返送がゼロであったことは、CAN データの Integrity が保証されていることを示すものと考えられる。今回は、ファジングテスト用には簡単な Integrity チェックを一時的に施したのみであるが、メッセージ認証メカニズムの有用性が検証出来た。

3.2b　コンポーネント・車内システムにおける評価技術の検討

3.2b.1　コンポーネント（リプログラミング）に対する評価方法・評価技術の検討

車業界のコンポーネント評価基準策定に向け、ECU に対する攻撃の発生可能性を定量化する必要がある。平成 27 年度では、リプログラミングモジュール認証への攻撃に着目し、発生可能性の定量化に向けた評価対象を開発した。また、この評価対象を使った実機評価（攻撃）の結果からは、認証機能の中で使われる乱数の重要性を再認識する結果を得た。

平成 27 年度は、ソフトウェアでの暗号アルゴリズム実装により実現した乱数発生器等のセキュリティ機能であったが、平成 28 年度はハードウェアの実装に置き換えていく。

ソフトウェア実装における人為的なミスから発生する脆弱性を排除したり、開発効率化の狙いから、暗号アルゴリズムをハード実装化する動きが見られるようになってきた。こういった動きを捉え、平成 28 年度は、マイコン内蔵のセキュリティハードウェア（以降、セキュリティ IP と称する）を活用した評価対象を開発した。

(1) 開発対象の概要

本テーマで開発する評価対象（コンポーネント）の概要について記述する。

① つながる車のコンポーネントとしての標準 ECU の概要

つながる車、あるいは自動走行車（Connected and Automated Vehicle）のコンポーネントとしての標準 ECU は、自動走行機能を構成するという性質を鑑み、設計段階から安全面および信頼面について慎重に配慮しなければならない。これを IPA が提唱しているソフトウェア品質 6 特性（機能性、信頼性、使用性、効率性、保守性、移植性）の観点から分析すると、上記 6 特性の中でも機能性に含まれるセキュリティ（Security）に着目する必要がある。その理由を以下に記す。

- 標準 ECU が外部の情報ネットワークに Gateway（Firewall）を介して接続される場合、情報ネットワークを経由した脅威の対象になることが想定されるため。
- 特に ECU プログラム自体を改変するような脅威が発生し攻撃が成功した場合、ECU の不正動作や機能の無効化が想定されるため。

上記理由のいずれも ECU や自動走行機能に対して重大な影響を与えるものである。設計者はこれらの影響を抑止するためにセキュリティ特性に着目し、コンポーネントのセキュリティモデルを構築する必要が有る。

本テーマにおいて開発する標準 ECU は、セキュリティ特性に対する評価が有効となるような機能を実装する。

※参照: 組込みソフトウェア開発における品質向上の勧め（コーディング編）
https://www.ipa.go.jp/files/000005106.pdf
※参照: 高信頼化ソフトウェアのための開発手法ガイドブック
https://www.ipa.go.jp/files/000005144.pdf

② リプログラミング環境全体

つながる車のコンポーネントとしての標準 ECU は、ECU を構成するプログラムのダウンロード（リプログラミング）を前提条件として設計する。これは標準 ECU における様々な機能上の問題を、ECU プログラム更新で解決するために重要な手段である。

リプログラミング手段に関しては、ISO14229-1 Road vehicles – Unified diagnostic service（UDS）によって診断サービスの一機能として規定されている。リプログラミング環境はこの規定を実現するために必要な環境である。構成する要素を図 3.2b.1-1 に記す。

・リプログラミング手順 – 共通規格（ISO14229-1）をベースにし、開発者の独自仕様を加えた診断サービス機能手順。
・リプログラミング手順を実施できる外部機器 – 一連のリプログラミング手順を何らかの手段で実行できる外部機器または PC ソフトウェア。
・リプログラミング可能な標準 ECU – 上記リプログラミング手順を機能として実現するソフトウェアモジュールを実装した ECU。

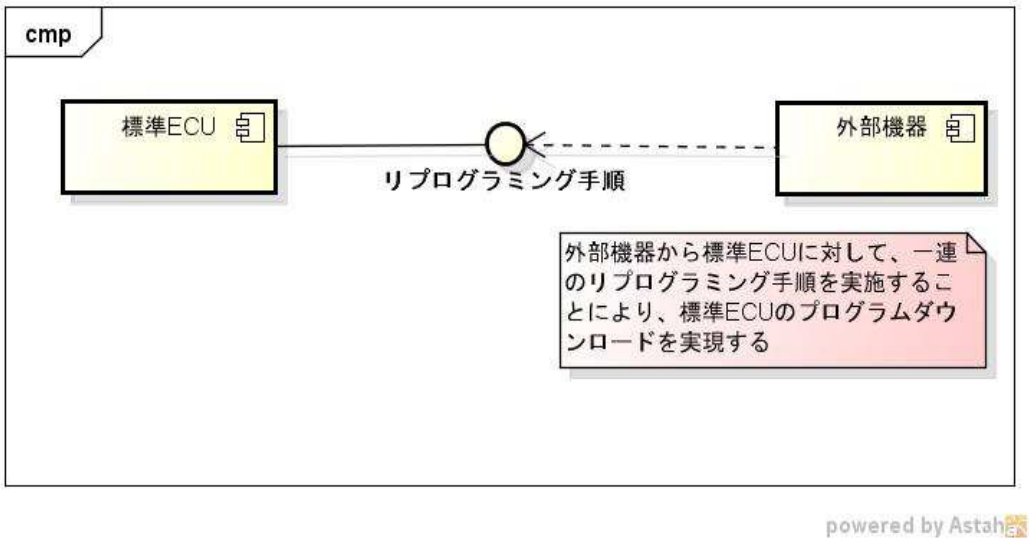


図 3.2b.1-1　リプログラミング環境構成

上記要素で構成するリプログラミング環境では、リプログラミング手順を保護するためのセキュリティ機能の堅牢性が重要となる。これはリプログラミング手順にセキュリティ上の脆弱性が存在する場合、不正な外部機器によるリプログラミングが実現する、すなわちセキュリティ攻撃が成功する確率が高まるためである。

本テーマでは、これらリプログラミング手順の信頼性を評価することが可能となる評価

コンポーネントを開発する。平成 27 年度に基本機能で構成される評価コンポーネントを開発した際、この評価コンポーネントのセキュリティ機能は、ソフトウェアでの暗号アルゴリズム実装により実現している。

平成 28 年度では、セキュリティ IP を使用したセキュリティ機能を、平成 27 年度版評価コンポーネントに追加実装した。ソフトウェアはセキュリティ IP へのインタフェース処理のみ実装し、暗号アルゴリズムの実装は行わない。

③ デバッグ環境全体

評価コンポーネントの開発期間および評価期間では図 3.2b.1-2 の動作環境を使用した。この環境はマイコン評価用およびリプログラミング手順検証用として実績がある環境構成である。

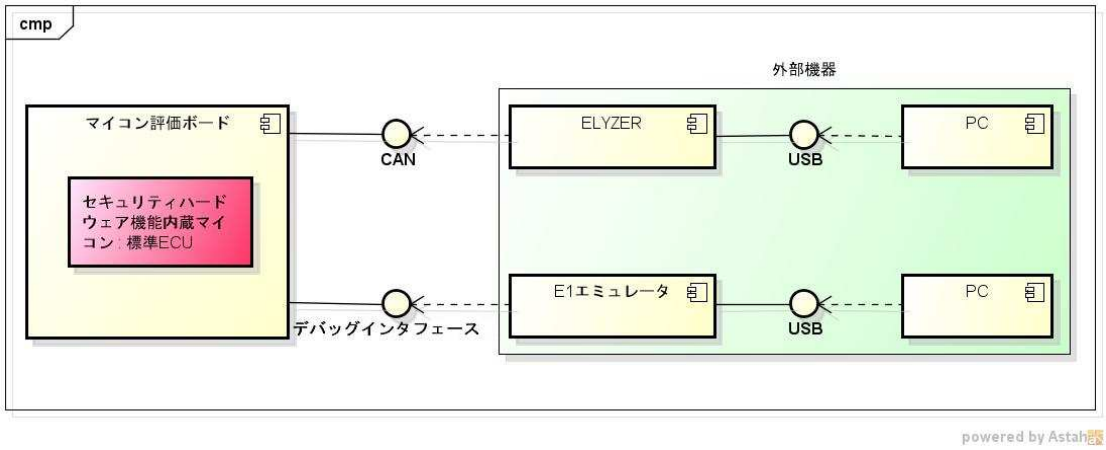


図 3.2b.1-2　デバッグ環境構成図

また両期間中においては、リプログラミング手順を実現する外部機器として ELYZER（イーソル株式会社製）を使用する。これはリプログラミング手順およびセキュリティ手順の動作を確認することを目的としている。更に ELYZER は CAN 通信アナライザ兼ログ取得ツールとしても使用可能である。

(2) 要求仕様

本テーマで開発する評価対象（コンポーネント）への要求仕様について記述する。

① 評価コンポーネント（標準 ECU）

評価コンポーネント（標準 ECU）に必要な機能を以下に記す。

・セキュリティ評価の対象となるような認証機能
・リプログラミング手順を実現するためのリプログラミング機能
・外部評価者が容易に評価コンポーネントの状態を認識できるような表示機能
・セキュリティ IP が提供する各種機能を使用するためのインタフェース機能

上記機能は、実際の ECU に近い形式で実装する。その為、基本的な ECU の機能やインタフェースを実現できる車載マイコン環境を用意する。パソコンおよびエミュレータ環境上での仮想動作による機能実現は行わない。

本評価コンポーネントの要件を表 3.2b.1-1 に記す。

表 3.2b.1-1　評価コンポーネント要件一覧

項　目	説　明
マイコン	一般的な 32bit 車載マイコン。セキュリティ IP を内蔵していること。 （例：ルネサス エレクトロニクス株式会社製 RH850 F1L/premium）。
通信規格	一般的な ECU 間通信、診断、リプログラミング、デバッグ等で用いられる CAN 通信。
評価ボード	使用する車載マイコンに対応した機能評価ボード。 CAN 通信インタフェースおよびデバッグポートを備えること。 動作状態を目視できる表示器（LED、LCD 等）を備えること。

※注: CAN ネットワーク仕様に関しては ISO15765-2 Road vehicles - Diagnostics on Controller Area Networks （CAN） - Part 2: Network layer services に準拠する。

② リプログラミング仕様

本評価コンポーネントで実現するリプログラミング仕様は、ISO14229-1 Road vehicles – Unified diagnostic service（UDS） Second edition 2013-03-15 の記述に準拠する。ISO14229-1 では一般的な診断サービスおよび各種仕様が規定されている。本評価では、必要最小限のリプログラミング仕様を ISO14229-1 Chapter 15 Non-volatile server memory programming process より抽出し設計と実装を行うものとする。

また評価コンポーネントには、リプログラミング禁止状態、許可状態を設定する。リプログラミング禁止状態から許可状態への遷移時は、セキュリティアクセス手順の正常完了を必須とする。

リプログラミング仕様要件を以下に記す。

・電源投入またはリセット後は、「リプログラミング禁止状態」に遷移すること。
・Diagnostic Session Control（DSC）診断サービスを用いてセキュリティアクセス手順を実行できる状態に遷移すること。
・「リプログラミング許可状態」に遷移するには、セキュリティアクセス手順を用いて、認証成功すること。
・セキュリティアクセス手順を用いずに「リプログラミング許可状態」に遷移しないこと。
・リプログラミング許可状態において、プログラムダウンロード手順を用いた FLASH メモリ書き換え機能を実現すること。
・リプログラミング禁止状態において、プログラムダウンロード手順を用いた FLASH メモリ書き換え機能を拒否すること。

リプログラミング仕様における状態遷移を図 3.2b.1-3 に、ISO14229-1 準拠のリプログラミング手順をシーケンスとして図 3.2b-1.4 に記す。

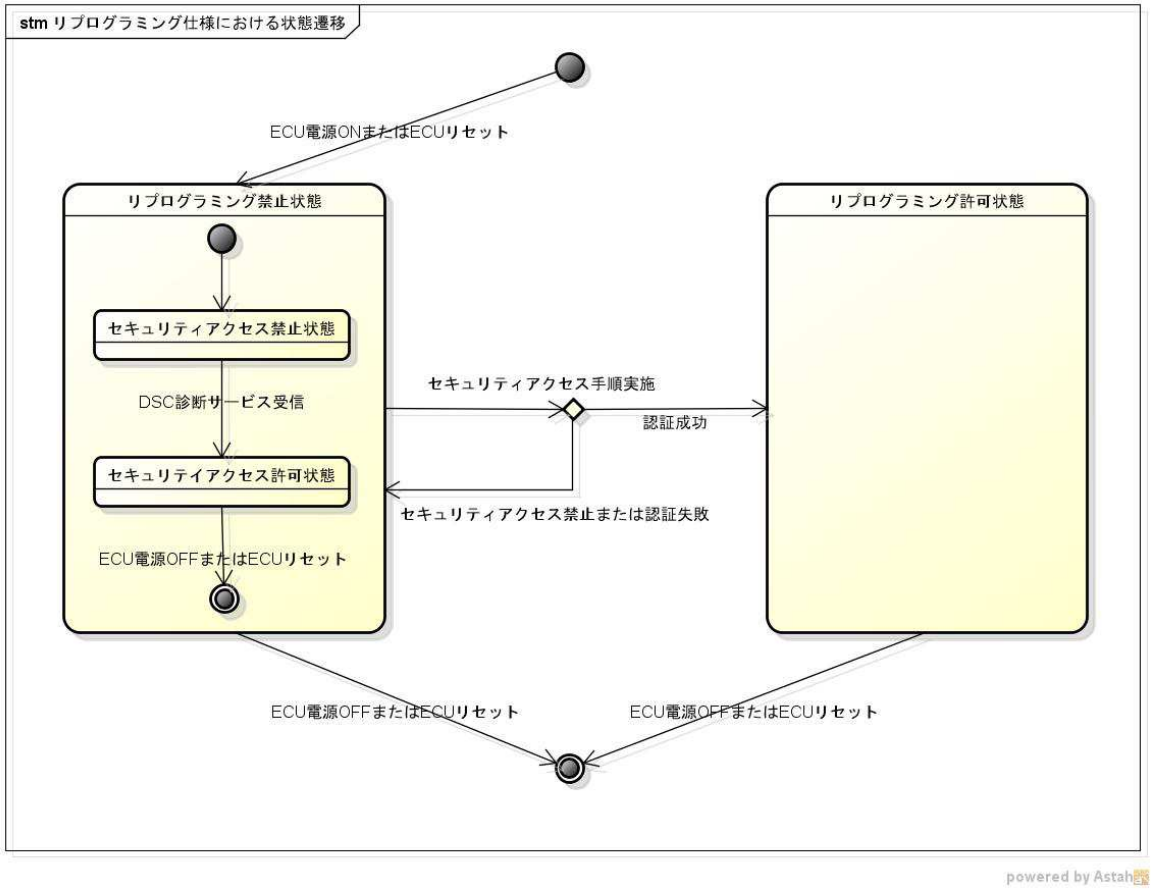


図 3.2b.1-3　リプログラミング仕様における状態遷移

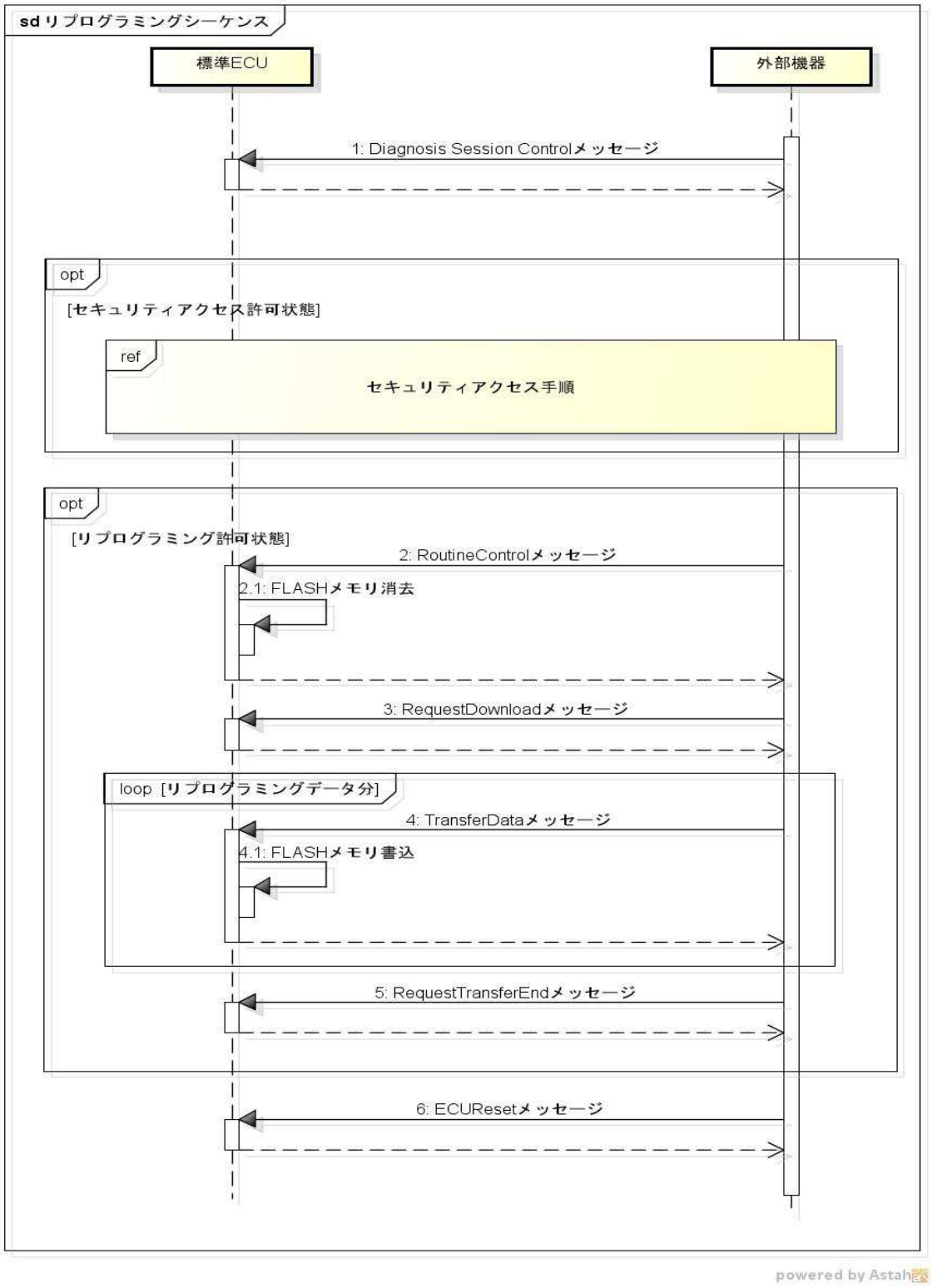


図 3.2b.1-4　リプログラミング手順

※注：シーケンス中の「～メッセージ」は ISO14229-1 にて規定された診断メッセージを示す。

※注：セキュリティアクセスシーケンスの詳細については、③セキュリティアクセス仕様にて規定する。

※注：2.RoutineControl メッセージから 5.RequestTransferEnd メッセージまでの手順がプログラムダウンロード手順となる。

③ セキュリティアクセス仕様

セキュリティアクセス仕様は、ISO14229-1 Road vehicles – Unified diagnostic service（UDS） Second edition 2013-03-15 の 15 Non-volatile server memory programming process および Annex I （normative） Security access state chart にある規定に準拠する。ただしこれらの規定にはツールからの ECU 認証形式のみ規定されており、その他の認証形式や手順詳細は規定されていない。よって本評価向けに認証形式と手順の詳細を定義する必要がある。

セキュリティアクセス仕様の機能要件を以下に記す。

・診断サービスメッセージを使用した相互認証（ECU からツール、ツールから ECU）を実現すること。
・暗号化アルゴリズムは AES128 を CBC モードで使用する。
・暗号化アルゴリズムで使用する鍵は、類推されにくい鍵を選定すること。
（例：「123456…」や「abcdef…」のような、一般的に脆弱とされる定型的な数値や文字列を使用しないこと。）
・擬似乱数生成器アルゴリズムとして、標準的な擬似乱数生成手順をソフトウェア実装すること。
・ソフトウェア実装した擬似乱数生成器初期化の有無を切り替えられるようにすること。
・ソフトウェア実装した擬似乱数生成器から生成される擬似乱数のエントロピーを切り替えられるようにすること。
・セキュリティ IP が提供する機能を用いた擬似乱数生成手順を実装すること。
・ECU リセット時には、認証状態を維持しないこと。

本評価においては、上記セキュリティアクセス手順失敗時のペナルティは設定しない。

ペナルティの例を以下に記す。

・セキュリティアクセス手順の連続実行における応答の遅延対応
・試行回数による応答変更や拒否対応

セキュリティアクセス手順をシーケンスとして図 3.2b.1-5 に記す。

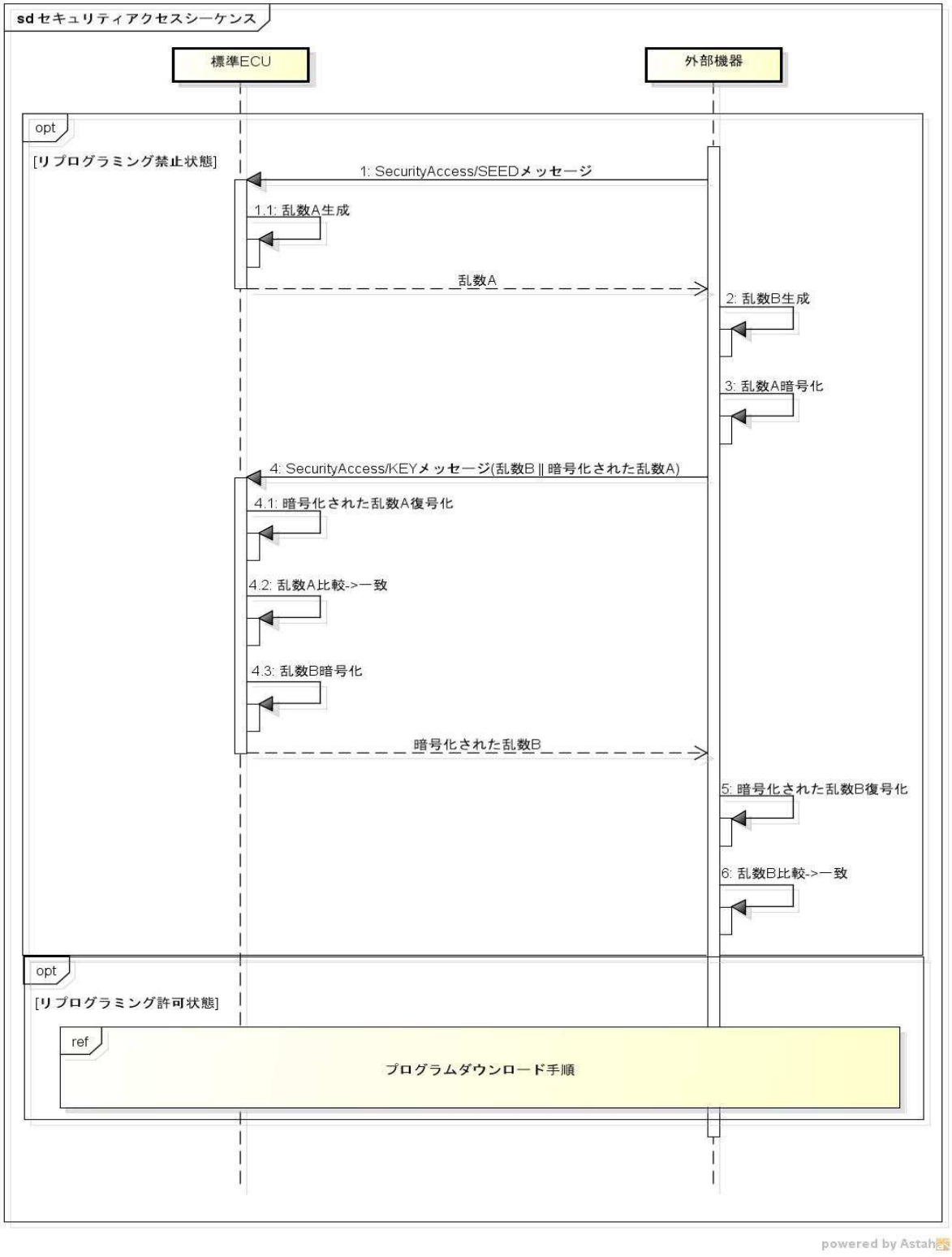


図 3.2b.1-5　セキュリティアクセス手順

④ デバッグインタフェース仕様

評価コンポーネントに対するデバッグインタフェース接続は、評価コンポーネントの開発およびデバッグ時に必須である。だが無条件で接続を許可する場合、評価コンポーネントへの物理的な攻撃（リバースエンジニアリングによる秘匿情報の暴露）が成功する危険性が大きくなる。この危険性を小さくするために、デバッグインタフェース接続では、正規のプログラミング環境またはデバッグ環境であるかを認証することが必須となる。

デバッグインタフェース仕様に関する要件を以下に記す。
・正規のプログラミング環境またはデバッグ環境であることを認証すること。
・不正なプログラミング環境やデバッグ環境からの接続を拒否すること。

⑤ 脆弱性水準

本評価に対してのセキュリティ評価を行う観点から、評価コンポーネントには異なる脆弱性水準を設定する必要がある。

ここで、平成 27 年度に定義した標準 ECU に必要な機能要件の一つを下記に示す。

FCS_RNG.1.2　TSF は、[割付: *定義された品質尺度*] を満たす [選択: *ビット, オクテットビット, 数* [割付: *数値の形式*]] の乱数を提供しなければならない。

この機能要件に割り付けられる値が脆弱性水準となり、セキュリティアクセス仕様に対して設定した水準を表 3.2b.1-2 に記す。水準①から④は平成 27 年度で実現した水準である。平成 28 年度では水準⑤を追加した。

表 3.2b.1-2　セキュリティアクセス脆弱性水準一覧

水準	乱数生成器 初期化有無	生成乱数空間長 （bit 長）	ECU で使用する 擬似乱数生成器	暗号化で 使用する鍵
①	有り	128bit	ソフトウェア実装	文字列 A の HASH 値
②	有り	4bit （先頭 bit から 124bit までは 0 とする。）	ソフトウェア実装	文字列 A の HASH 値
③	無し	128bit	ソフトウェア実装	文字列 A の HASH 値
④	無し	4bit （先頭 bit から 124bit までは 0 とする。）	ソフトウェア実装	文字列 A の HASH 値
⑤	有り	128bit	セキュリティ IP が提供する機能	文字列 B の HASH 値

※注：セキュリティ IP が提供する機能を使用した乱数生成では、乱数生成器の初期化が必須となるため、⑤と対になる「初期化無し」の水準は設定しない。

セキュリティ強度の観点から見ると⑤>①>②>③≒④の順位となる。

生成される乱数の空間長が十分に長い場合、生成乱数を用いたセキュリティ強度は高まる。ただし強度を高める条件として「完全乱数に近い擬似乱数、つまり類推されにくい乱数を生成する」ことが条件となる。これは乱数生成パターン解析からのメッセージデータ構築とメッセージ再送攻撃を抑止もしくは遅延するために必須の事項となる。

セキュリティ IP が提供する機能による乱数生成（上記水準の⑤）よりも、ソフトウェア実装による乱数生成（上記水準の①）の強度が弱い理由は、乱数生成アルゴリズムのソフトウェア実装に起因する。

平成 27 年度にソフトウェアで実装した乱数生成アルゴリズムは、単純な線形方程式である。よって一定の条件下（例:乱数生成器に与える初期値が固定される状況）において生成される乱数は一定の生成パターンを示す。また線形方程式を使用しているため、現在の乱数値から、次に生成される乱数値を類推することも可能である。

平成 28 年度に実装した、セキュリティ IP が提供する機能による乱数生成においては、真性乱数を初期値として与えられる擬似乱数生成器から生成結果を取得する。予測不可能な真性乱数を乱数生成器の初期値とするため、生成される乱数は不定パターンとなる。また生成アルゴリズムおよび中間値はソフトウェア実装から隔離されているため、生成された現在の乱数値から次に生成される乱数値を類推することも困難となる。

脆弱性水準の③と④の場合、生成される乱数は常に固定された値となり、生成パターンも固定される。これは乱数生成器の初期化が無いことに起因する。その結果、メッセージ再送攻撃に極めて弱くなることが想定される。

デバッグインタフェースに対して設定する脆弱性水準を表 3.2b.1-3 に記す。

表 3.2b.1-3　デバッグインタフェース脆弱性水準一覧

水準	接続用パスワード有無
①	有り
②	無し

(3) ソフトウェア仕様

本テーマで実装するセキュリティ評価向けソフトウェアの仕様について記述する。

① セキュリティ評価向けソフトウェア

本テーマで求められる機能要件を実現するためのセキュリティ評価向けソフトウェアを構成する要素を表 3.2b.1-4 に記す。

表 3.2b.1-4　セキュリティ評価向けソフトウェア構成要素一覧

要素名	説明
Boot/Loader 部	マイコンリセット時に実行を開始するモジュール。マイコンおよび周辺ペリフェラルの初期化、および ISO14229-1 診断サービスモジュール起動またはアプリケーションの起動を行う。
Application 部	リプログラミング対象となるモジュール。本評価向けソフトウェアでは、一定周期で LED 点滅または点灯を行う機能（ON/OFF パターンは複数ある）を実装する。これは評価者（攻撃者）が、リプログラミングに成功したことを目視するための機能である。また CAN インタフェースによる診断メッセージの受信処理およびリプログラミングモジュールの起動処理も実装する。
ISO14229-1 診断サービスモジュール	ISO14229-1 リプログラミング手順を実現するためのモジュール。診断サービスに依存する処理は本モジュール内に実装する。
リプログラミングモジュール	リプログラミング手順のうち、プログラムダウンロードを実現するためのモジュール。FLASH メモリ書き換えに関する処理を行う。
セキュリティアクセスモジュール	セキュリティアクセス手順を実現するためのモジュール。認証処理および認証状態管理を行う。
セキュリティモジュール	セキュリティアクセス手順処理で必要となるアルゴリズムを実装するモジュール。アルゴリズムインタフェースを提供する。 またセキュリティ IP が提供する機能を使用するためのインタフェース機能を提供する。

上記構成要素のうち、ISO14229-1 診断サービスモジュール、リプログラミングモジュール、セキュリティアクセスモジュールおよびセキュリティモジュールに関しては、②以降で仕様詳細を記述する。

セキュリティ評価ソフトウェアおよび Application、外部機器の構成を図 3.2b.1-6 に記す。

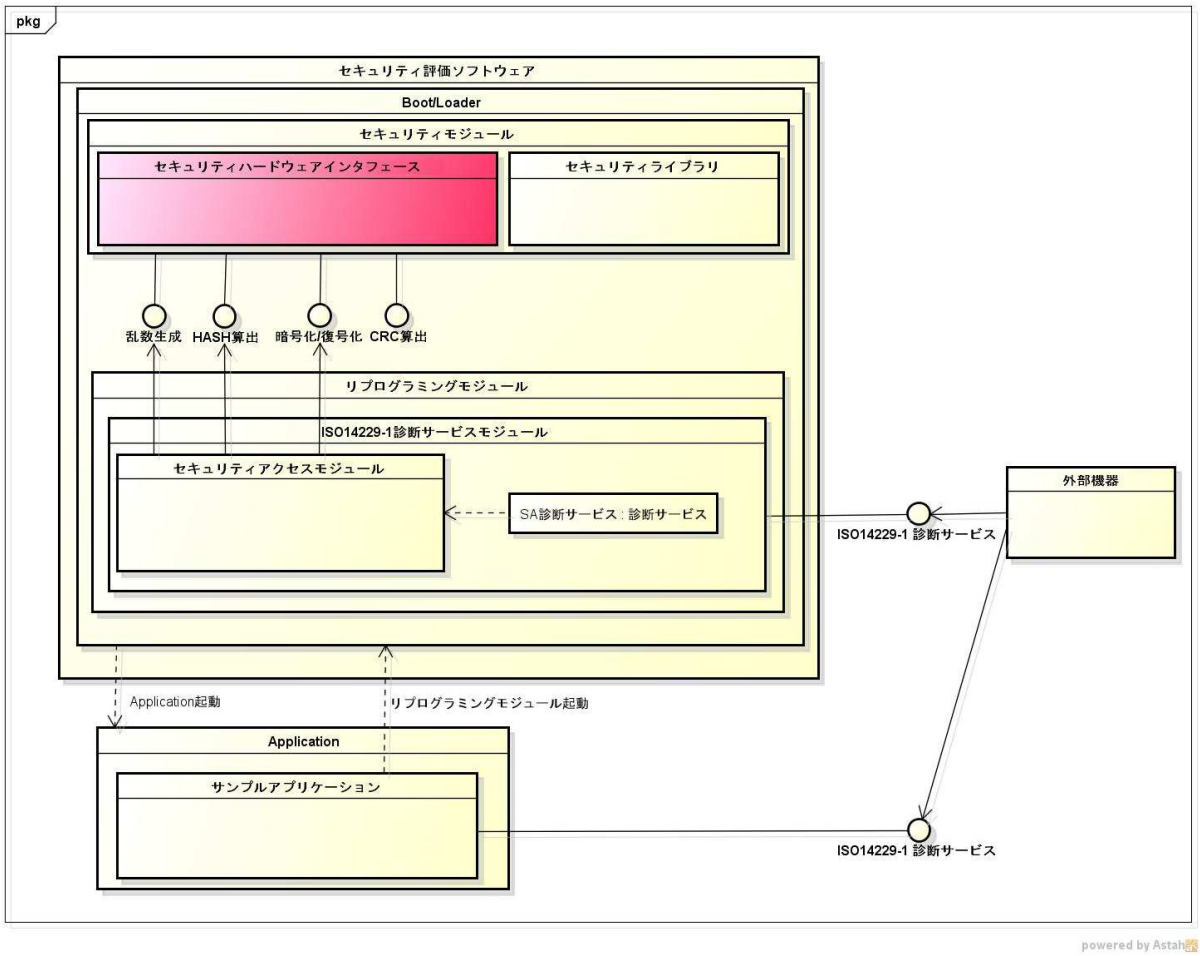


図 3.2b.1-6　セキュリティ評価向けソフトウェア構成

② ISO14229-1 診断サービスモジュール

ISO14229-1 診断サービスモジュールは、ISO14229-1 に規定されている診断サービスに依存する実装で構成される。実現する機能を以下に記す。

・診断要求メッセージの解析。
・セキュリティアクセスモジュールの実行。
・リプログラミングモジュールの実行。
・診断応答メッセージの構築と送信。
・その他診断サービスに関する実装。

本モジュールは、セキュリティアクセスモジュールとリプログラミングモジュールを内包する。処理する診断サービスに従って、いずれかのモジュールを実行する。

ISO14229-1 診断サービスモジュールを構成する要素を表 3.2b.1-5 に、ISO14229-1 診断サービスモジュール構成を図 3.2b.1-7 に記す。

表 3.2b.1-5　ISO14229-1 診断サービスモジュールを構成要素一覧

要素名	説明
インタフェース部	本評価ソフトウェアの場合 Boot/Loader 部から呼び出されるためのインタフェースを実装する。
フレームワーク部	診断サービス機能を実現するための実装部。周期処理管理やプロセス実行などを実装する。
診断サービス部	ISO14229-1 に規定されたリプログラミング手順を実現するための診断サービス固有実装部。診断要求メッセージの解析と判定、リプログラミング動作として必要なプラットフォームインタフェースの呼び出しとプラットフォーム処理状態確認、診断応答メッセージの構築と送信を行う。
プラットフォーム部	Boot/Loader プラットフォームに実装するデバイスドライバとのインタフェース変換処理群。診断サービスとプラットフォーム間の仕様差分は本実装で解決する。

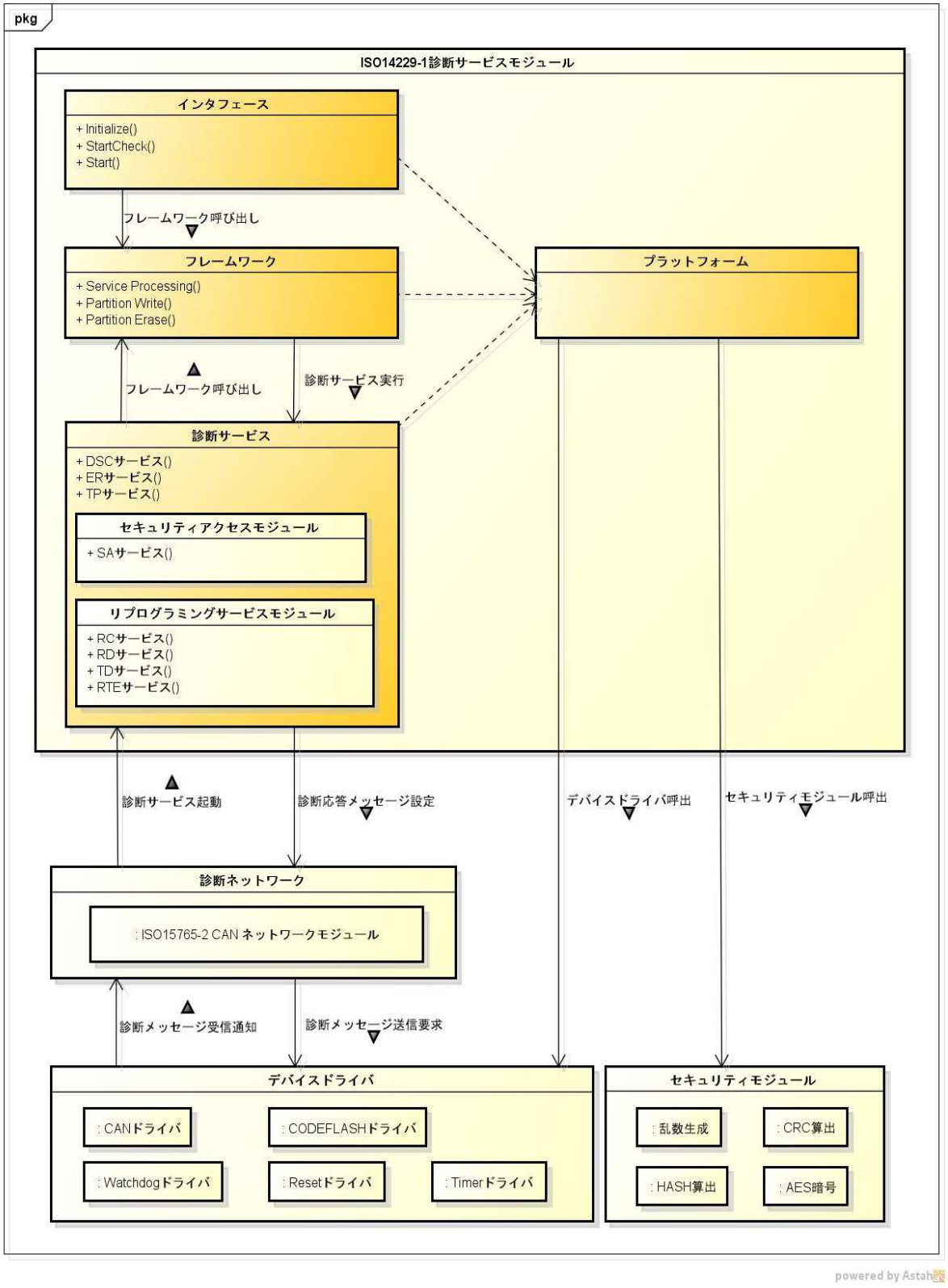


図 3.2b.1-7　ISO14229-1 診断サービスモジュール構成

③ リプログラミングモジュール

プログラムダウンロード機能を実現するリプログラミングモジュールを構成する要素を表 3.2b.1-6 に記す。

表 3.2b.1-6　リプログラミングモジュール構成要素一覧

要素名	説明
RC サービス実装	RoutineControl 診断サービス実装。本評価ソフトウェアの場合、FLASH メモリの消去を実行する。
RD サービス実装	RequestDownload 診断サービス実装。TransferData 診断サービスの実行に必要な手続きを行う。
TD サービス実装	TransferData 診断サービス実装。書き換え対象のデータを受信し、FLASH メモリに書き込む処理を実行する。
RTE サービス実装	RequestTransferData 診断サービス実装。FLASH メモリ書き込みすなわちプログラムダウンロードの終了処理を行う。

④ セキュリティアクセスモジュール

セキュリティアクセス手順を実現するセキュリティアクセスモジュールを構成する要素を表 3.2b.1-7 に記す。

表 3.2b.1-7　セキュリティアクセスモジュール構成要素一覧

要素名	説明
SA サービス実装	SecurityAccess 診断サービス実装。認証手続きに必要な SEED の生成と、外部機器から通知される KEY の認証処理を実行する。

単一の評価向けソフトウェアで複数の脆弱性水準を評価することを可能とするために、上記 SA サービス実装では、診断サービス要求メッセージ内のパラメータの Security Level（LEV）を設定する。異なる Security Level を指定することにより、Security Level に割り当てられた脆弱性水準に従ってセキュリティアクセス手順を実施する。これらの実装は単一の物理環境上において、脆弱性水準が異なる ECU としての仮想評価環境を実現する。

脆弱性水準の切り替えを実現する診断サービス要求メッセージについては、3.2b.4 付録 C 評価対象（コンポーネント資料） 表 3.2b.4-2 のパラメータ一覧に記載する。

※注:Security Level（LEV）は、ISO14229-1 に option 扱いで規定されているパラメータである。

⑤ セキュリティモジュール

セキュリティモジュールは、リプログラミング手順で使用する各種セキュリティ機能を実現するために必要な機能を実装するモジュールである。このモジュールでは、診断サービスに依存した実装は行わない。

セキュリティモジュールで実装する機能を表 3.2b.1-8 に記す。

表 3.2b.1-8　セキュリティモジュール実装機能一覧

機能名	説明
乱数生成	乱数を生成する機能。ソフトウェア生成およびハードウェア生成の選択を可能とする。 ソフトウェア生成向けでは、C 言語向けの標準乱数生成器実装を採用する。 ハードウェア生成向けでは、セキュリティ IP が提供する機能を使用して乱数を生成する。
HASH 算出	HASH 計算を行う機能。 オープンソースである mbedTLS から SHA1 実装を抽出し実装する。
AES 暗号処理	AES 暗号化/復号化を行う機能。 オープンソースである mbedTLS から AES128 実装を抽出し実装する。
CRC 算出	CRC 計算を行う機能。 CRC に関しては CCITT-16 で実装する。

上記機能の内、乱数生成機能に関しては、ソフトウェアライブラリの使用またはセキュリティ IP が提供する機能の使用を選択可能とする。またセキュリティモジュールは、セキュリティ IP を制御する機能を内部構造として実装する。セキュリティ IP の制御方法およびそれらを実現する実装構造は、セキュリティ IP の実装仕様に依存する。

⑥ デバッグインタフェース

本評価コンポーネントを開発、評価する際に使用するデバッガとマイコン評価ボードを接続するインタフェースは、LPD 接続方式を採用する。採用理由を以下に記す。

・評価用ボードに Low Pin Debug interface（LPD）接続用コネクタが実装されている
・評価用ボードに Joint Test Action Group（JTAG）接続用コネクタが実装されていない
・本マイコンの標準状態では JTAG 接続設定が無効となっている

また本評価で使用するマイコンでは、マイコン内に搭載されている OCD（On Chip Debugger）に対してセキュリティパスワードを設定することが可能である。このセキュリティパスワード設定がマイコン（OCD）とデバッガで不一致となる場合、デバッガからの FLASH コード参照を防止することが可能となる。つまりリバースエンジニアリングのようなハードウェアに対する攻撃を抑止できる。

セキュリティパスワードは、専用の FLASH メモリプログラマ（外部機器）から任意の値をマイコンに書き込むことで設定する。ただしセキュリティパスワード設定と確認手順、ハードウェア観点としての LPD 接続手順、マイコン内部の OCD へのアクセス方法などは、マイコンメーカが非公開としているデバイス仕様である。

デバッグインタフェースの設定を、以下に記す。

・JTAG 接続設定: 無効（JTAG 接続用専用端子設定=無効）

・LPD 接続設定: 有効（ソフトウェア設定項目は無し）

・セキュリティパスワード: 未設定（default 値: 0xFFFFFFFFFFFFFFFF）

(4) リプログラミング評価

標準 ECU コンポーネント上で動作するセキュリティ評価ソフトウェアが提供するリプログラミング機能に対する評価の観点を以下に記す。

・3.2b.1(3)②で規定したリプログラミング手順を外部機器から実行し、リプログラミング対象である標準 ECU コンポーネント内の Application 部プログラムが正常に更新されること。

・リプログラミング手順内のセキュリティアクセス手順において、設定した脆弱性水準毎に正常終了すなわち認証成功を確認できること。

リプログラミング評価対象の構成を図 3.2b.1-8 に記す。

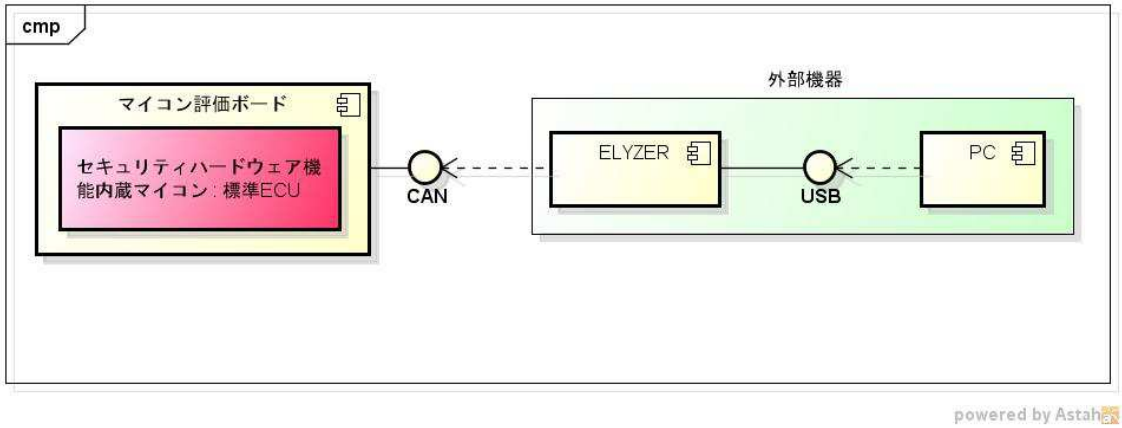


図 3.2b.1-8　リプログラミング評価対象構成

本評価では、外部機器としてイーソル株式会社製 ELYZER を使用する。ELYZER はハードウェアおよび PC 上で動作する専用アプリケーションで構成される。

ELYZER の接続例を図 3.2b.1-9 に記す。

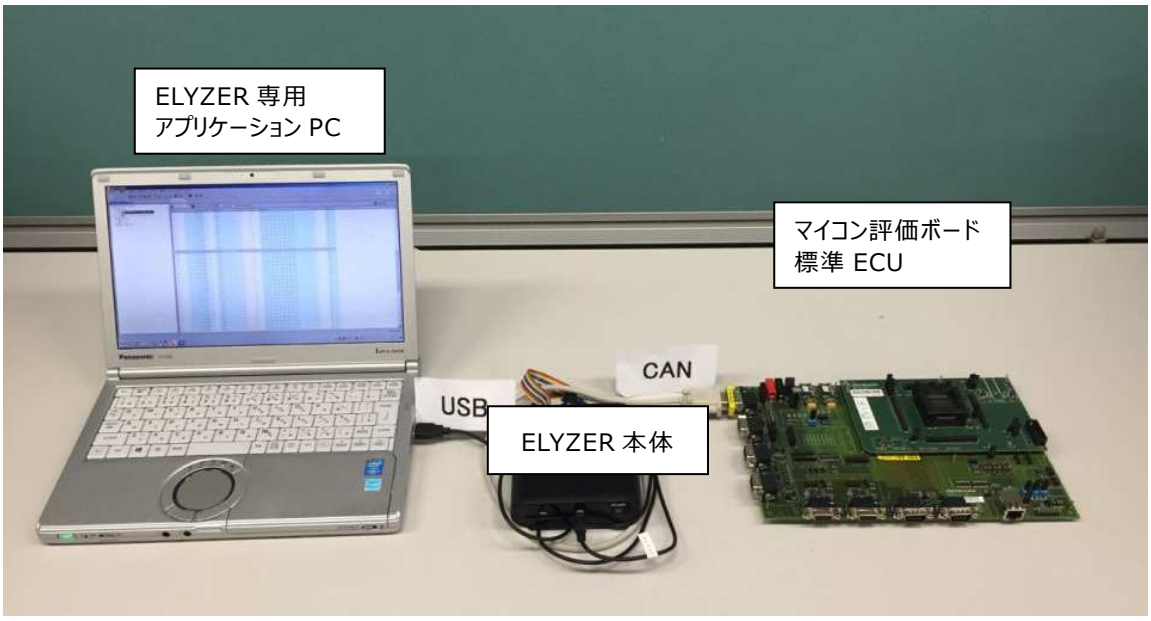


図 3.2b.1-9　ELYZER を用いた評価対象の接続例

ELYZER は「モデル」という概念で ECU との通信シーケンスを実現する。モデルではリプログラミング手順に必要な診断メッセージの送受信を記述する。この記述中にはセキュリティアクセス時に必要な認証動作も存在する。モデルは生成専用 EXCEL ファイルでの自動生成または手動編集によって、専用のインタープリタ言語として記述する。

ELYZER 専用アプリケーションのスクリーンショットを図 3.2b.1-10 に記す。

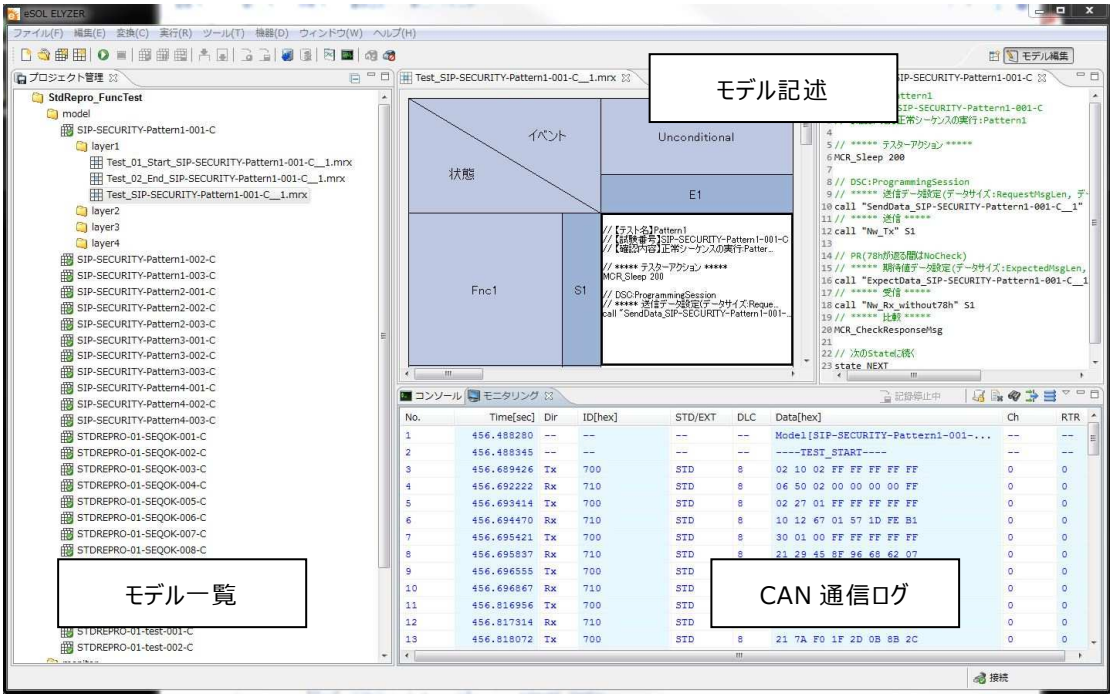


図 3.2b.1-10　　ELYZER 専用アプリケーション

また ELYZER は CAN アナライザ（モニタ）としての機能も持つ。セキュリティアクセス手順の確認時には、ELYZER で取得した CAN 通信ログ（csv 形式ファイル）の出力内容を使用する。CAN 通信ログの例を図 3.2b.1-11 に記す。

	A	B	C	D	E	F	G	H	I	J
1	Date Mon Mar 07 17:13:13 2016									
2	Timestamps absolute									
3	Ch0 Bitrate 500Kbps TSEG1 6 TSEG2 3 SJW 1 BRP 1 BSP 1									
4	Ch1 Bitrate 500Kbps TSEG1 5 TSEG2 2 SJW 1 BRP 4 BSP 1									
5	No.	Time[sec]	Dir	ID[hex]	STD/EXT	DLC	Data[hex]	Ch	RTR	
6	1	28.984263	--	--	--	--	Model[SIP-SECURITY-Pattern1-001-C] start	--	--	
7	2	28.984328	--	--	--	--	----TEST_START----	--	--	
8	3	29.024558	RX	1800ffff	EXT	8	ff 23 a3 aa 88 88 86 ff	0	0	
9	4	29.185407	TX	700	STD	8	02 10 02 ff ff ff ff ff	0	0	
10	5	29.186219	RX	710	STD	8	06 50 02 00 00 00 00 ff	0	0	
11	6	29.187412	TX	700	STD	8	02 27 01 ff ff ff ff ff	0	0	
12	7	29.188464	RX	710	STD	8	10 12 67 01 21 0a c5 28	0	0	
13	8	29.189412	TX	700	STD	8	30 01 00 ff ff ff ff ff	0	0	
14	9	29.189832	RX	710	STD	8	21 7e 98 52 41 6f 31 03	0	0	
15	10	29.190550	TX	700	STD	8	30 01 00 ff ff ff ff ff	0	0	
16	11	29.190860	RX	710	STD	8	22 e6 30 f2 ed 27 ff ff	0	0	
17	12	29.240949	TX	700	STD	8	10 22 27 02 13 15 93 37	0	0	
18	13	29.241395	RX	710	STD	8	30 01 00 ff ff ff ff ff	0	0	
19	14	29.242149	TX	700	STD	8	21 06 94 a0 a4 79 12 8e	0	0	
20	15	29.242421	RX	710	STD	8	30 01 00 ff ff ff ff ff	0	0	
21	16	29.243180	TX	700	STD	8	22 0d 5f 3c a1 c2 6a 6a	0	0	
22	17	29.243495	RX	710	STD	8	30 01 00 ff ff ff ff ff	0	0	
23	18	29.244260					7e fb 30 bd 00 c6	0	0	
24	19	29.244676					00 ff ff ff ff ff	0	0	
25	20	29.245434					cb 56 7f 07 bc 9a	0	0	
26	21	29.249028					67 02 3d 50 67 83	0	0	
27	22	29.249973	TX	700	STD	8	30 01 00 ff ff ff ff ff	0	0	
28	23	29.250389	RX	710	STD	8	21 cc e4 34 0a a2 d0 c3	0	0	
29	24	29.251107	TX	700	STD	8	30 01 00 ff ff ff ff ff	0	0	
30	25	29.251417	RX	710	STD	8	22 a9 e1 4a 6c d9 ff ff	0	0	
31	26		TX	700	STD	8	10 0d 31 01 ff 00 44 00	0	0	
32	27		RX	710	STD	8	30 01 00 ff ff ff ff ff	0	0	
33	28		TX	700	STD	8	21 01 00 00 00 01 00 00	0	0	
34	29	29.413713	RX	710	STD	8	04 71 01 ff 00 ff ff ff	0	0	

図 3.2b.1-11　CAN 通信ログ例

(5) まとめ

つながる車のコンポーネントと車内システムに対して、セキュリティ対策の妥当性を定量的に確認可能な評価技術を開発すべく、コンポーネントにおける評価方法・評価基準案の検討、評価対象となる標準的な ECU の開発を行ってきた。

平成 28 年度は、平成 27 年度で開発した標準 ECU コンポーネントに対しての機能拡張として、乱数生成機能においてソフトウェアライブラリによる生成と、マイコンに内蔵されるハードウェアのセキュリティ IP が提供する機能による生成の選択を可能とするための開発を行った。合わせて、これらの乱数生成機能を評価するために設定したセキュリティアクセス脆弱性水準を拡張し、同一 ECU において異なるセキュリティアクセス脆弱性を幅広く評価できるように配慮した。

その結果、開発した ECU は、ソフトウェアでの暗号アルゴリズム実装とハードウェア実装との両面を備え、より一層標準的に使える評価対象とすることができた。

3.2b.2　車内システム（鍵管理）に関連する他業界の事例調査

つながる車の車内システムにおいては、ECU のなりすまし対策や通信データの改竄防止を目的として、メッセージ認証等の暗号鍵を必要とする技術の導入が考えられている。また、車載マイコンに内蔵されるセキュリティ IP の中には、暗号鍵をセキュアに配送、保管、更新する機能が考えられたものも出てきている。

このように技術が開発される一方で、自動車業界の中では鍵をセキュアに管理・運用するための体系的な規格はまだ存在していない。本調査では技術的な仕組みだけでなく、鍵のライフサイクル全体において他業界との要件を比較・整理し、自動車業界で今後定めるべき推奨要件について整理する。

(1) 調査概要

① 概要

②、③に、鍵のライフサイクルにおける他業界ごとの取り組みについて調査する事前準備として以下の説明を行う。

・業界ごとの鍵管理要件の比較項目の説明
・調査対象とする業界の選定理由
・調査対象とした業界で採用されている鍵管理に関する規格

② 業界ごとの鍵管理要件の比較項目の説明

本調査では、業界ごとの鍵管理における要件を比較するため、鍵管理に関する項目がライフサイクルごとに定められている NIST SP800-57 part1 の要件を、比較項目として採用した。NIST SP800-57 part1 は、鍵管理のセキュリティ要件としてクレジットカード業界や銀行、電力等の業界から参照されている。

NIST SP800-57 part1 では、鍵のライフサイクルが運用前フェーズ、運用フェーズ、運用後フェーズ、破棄フェーズの 4 フェーズに分類されている。鍵の生成、配送、保管や回復における要件を明確化するため、NIST SP800-57 part1 にて定められている要件を、IPA の安全な暗号鍵のライフサイクルマネージメントに関する調査[1]や、CRYPTREC のリストガイド（鍵管理）[2]、ISO11568[3]の考え方を参考に、生成フェーズ、配布フェーズ、利用フェーズ、保管フェーズ、廃棄フェーズ、回復フェーズの、6 フェーズ全 18 要件に整理した。

(ｉ) 生成フェーズ

鍵の生成フェーズにおける、鍵の漏洩や推測を防ぐために考慮すべき要件について、NIST SP800-57 part1 の要件を基に表 3.2b.2-1 に整理した。NIST SP800-57 part1 では鍵ライフサイクルを規定した規格であるため、鍵自体の強度に関する要件は定められていない。

表 3.2b.2-1　鍵の生成フェーズで考慮すべき要件

要件番号	要件	要求事項
要件 1	適切な暗号モジュールの利用	全ての暗号鍵は、適切な暗号モジュールの内部にて生成されること
要件 2	生成する鍵へのアクセス制御	単一の個人による、生成時の鍵へのアクセスを防止すること
要件 3	適切な擬似乱数生成器の利用	適切な擬似乱数生成器を利用すること
要件 4	推測が困難な値の利用	秘密鍵は推測が困難な適切な乱数、マスタ鍵から生成されること
要件 5	再現困難な鍵の生成	一つの鍵が危殆化した場合に、他の鍵を再現することが不可能であること

(ⅱ) 配布フェーズ

鍵の配送フェーズにおける、鍵の漏洩やなりすまし、改竄等を防ぐために考慮すべき要件について、NIST SP800-57 part1 の要件を基に表 3.2b.2-2 に整理した。

表 3.2b.2-2　鍵の配布フェーズで考慮すべき要件

要件番号	要件	要求事項
要件 6	配送先の特定	鍵の配送は受取先のなりすまし等が行われていないことを確認し、配送先が確実であると特定できた場合にのみ行われること
要件 7	通信での鍵の配送時の保護	通信経路にて配送される鍵は、以下の要件に従い、適切に保護されること ・機密性の確保のために、鍵の配送に先立ち適切に配布された鍵暗号化鍵を利用して暗号化された上で配送されること ・完全性の確保のために、MAC、デジタル署名等を適用すること ・可用性の確保のために、多重化、誤り訂正符号等の暗号以外のメカニズムを適用すること
要件 8	通信経路の保護	通信経路での鍵のすり替え、改竄等を防ぐために、通信経路を保護すること
要件 9	手動での鍵の配送時の保護	手動にて配送される鍵は、以下の要件に従い、適切に保護されること ・郵送事業者等が提供するプロセスにより、鍵の配送元と配送先の両方が認証されていること ・鍵は暗号化され、耐タンパ性を有する HSM 等に保管した上で配送すること
要件 10	鍵の構成要素のロード、デバイスへの配送	鍵の構成要素のロードおよびデバイスへの配送の際は、以下の要件に従い、適切に保護されること ・鍵の構成要素のロードおよびデバイスへの配送は、相互監視下で行われること ・鍵の構成要素は知識分割により保護されること

(iii) 利用フェーズ

鍵の利用フェーズにおける、鍵の不正利用や危殆化を防ぐために考慮すべき要件について、NIST SP800-57 part1 の要件を基に表 3.2b.2-3 に整理した。

表 3.2b.2-3　鍵の利用フェーズで考慮すべき要件

要件番号	要件	要求事項
要件 11	鍵へのアクセス時の認証	認証等により、許可されている者のみがアクセスできる仕組みを用意すること
要件 12	鍵の更新	鍵の有効期限が終了した場合、もしくは鍵が危殆化した場合には新しい鍵へ更新すること

(iv) 保管フェーズ

鍵の保管フェーズにおける、鍵の改竄や漏洩を防ぐため、また鍵を即座に利用可能とするために考慮すべき要件について、NIST SP800-57 part1 の要件を基に表 3.2b.2-4 に整理した。

表 3.2b.2-4　鍵の保管フェーズで考慮すべき要件

要件番号	要件	要求事項
要件 13	鍵の適切な保管	鍵は機密性を確保するために、暗号化の実施、暗号モジュール内、アクセスが管理されているセキュアな保管庫のいずれかに保管されること
要件 14	鍵の完全性確保	鍵は改竄から、改竄の予防、改竄の検知、回復を行う仕組みにより保護されること
要件 15	鍵のバックアップ	鍵の保管の際は以下の要件に従い、適切に保護されること ・鍵のバックアップは、鍵の有効期限が終わる前に取得され、バックアップ元と異なる記憶媒体に保管されること ・バックアップされている鍵が漏洩する危険性を下げるために、厳格な運用を行うこと （例：バックアップ取得時の 2 人以上による相互監視等）

(v) 廃棄フェーズ

鍵の廃棄フェーズにおける、鍵の廃棄方法、鍵の利用を無効にするための適切な失効通知について考慮すべき要件を、NIST SP800-57 part1 の要件を基に表 3.2b.2-5 に整理した。

表 3.2b.2-5　鍵の廃棄フェーズで考慮すべき要件

要件番号	要件	要求事項
要件 16	鍵の廃棄方法	鍵の有効期間の終了時、あるいは鍵が危殆化した場合、鍵の全てのコピーを含め確実に廃棄されること
要件 17	鍵の失効通知	鍵の利用を停止する際は、全ての鍵の管理者に鍵の失効が通知されること

(vi) 回復フェーズ

鍵の回復フェーズにおける鍵へのアクセスができなくなった場合に考慮すべき要件を、NIST SP800-57 part1 の要件を基に表 3.2b.2-6 に整理した。

表 3.2b.2-6　鍵の回復フェーズで考慮すべき要件

要件番号	要件	要求事項
要件 18	暗号鍵の回復	鍵へのアクセスができなくなった場合に備え、バックアップからの回復フェーズにおける運用方針が事前に策定されること

③ 調査対象業界の選定と対象業界の鍵管理に関する規格

鍵管理の要件を調査する対象業界として、自動車業界で今後採用されるであろう鍵管理方式の以下の特徴を採用している業界を選定した。

・事前共有鍵を用いた鍵管理方式を採用していること。
・鍵の更新機能を実装していること。

本調査では公開されている規格を対象としており、業界や各企業にて独自に作成されていても、非公開となっている規格は対象外とした。そのため上記特徴に加えて、鍵管理に関する公開情報が存在することも選定条件とした。

業界選定の結果を表 3.2b.2-7 に示す。公共・金融・交通等のインフラ環境の中から、事前共有鍵方式を用いた鍵管理方式を採用しており、鍵更新の仕組みを有し、鍵管理に関する規格が公開されているという理由より、クレジットカード業界、銀行業界、電力業界を調査対象業界として選定した。

表 3.2b.2-7　業界選定の整理結果

業種（対象機器）	事前共有鍵の利用有無	鍵更新の有無	鍵管理に関する公開情報の有無（仕様・ガイドライン等）	デバイス数（参考）
クレジットカード業界(決済端末)	○	○	○	約 150 万台[4]
銀行（ATM 等）	○	○	○	約 14 万台[5]
電力（スマートメータ）	○	○	○	約 7823 万台[6]
放送（B-CAS カード）	○	×	○	約 2 億 3664 万枚[7]
通信(SIM カード)	×	×	×	約 1 億 5954 万枚[8]
交通（鉄道）	—	—	—	鉄道：約 12,967 両（JR 東日本）[9]

【凡例】○：当てはまる　×：当てはまらない　　—：詳細不明

調査対象とした各業界の選定に関する説明と、調査対象業界で採用した鍵管理に関する規格は以下の通りである。

(ⅰ) クレジットカード業界

クレジットカード業界では、決済情報という秘匿性が高い情報を取り扱う関係上、鍵管理について厳密な規格が存在する。

クレジットカードの情報を取り扱う場合、カードを読み取るクレジットカード決済端末（以下、POI）からセンターまで様々な通信経路にて会員データが送信されるため、全通信経路にて会員データを暗号化する必要がある。会員データのセキュリティの保護を実現する際、クレジットカード業界では、事前共有鍵方式が採用されている。また、POI に保管されている鍵の危殆化に対応するために、鍵更新の仕組みも採用されている。

クレジットカード業界において、POI における鍵管理の要件が定められている PCI P2PE を調査対象とした。

(ⅱ) 銀行業界

銀行業界のシステムは、口座情報や決済情報という秘匿性が高い情報を取り扱う関係上、鍵管理について厳密な規格が存在する。

口座情報や決済情報を取り扱う場合、ATM（現金自動預け払い機）からセンターまで様々な通信経路にて情報が送信されるため、全通信経路にて口座情報や決済情報を暗号化する必要がある。口座情報や決済情報のセキュリティの保護を実現する際、銀行業界では事前共有鍵方式が採用されている。また、ATM に保管されている鍵の危殆化に対応するために、鍵更新の仕組みも採用されている。

銀行業界において、共通鍵のライフサイクルにおける鍵管理の要件が定められている ISO11568-2:2012 を調査対象とした。

(ⅲ) 電力業界

電力業界ではスマートメータと電力会社間でやり取りされる電力使用量等は秘匿性が高い情報として定められている。

秘匿性の高い電力使用量等の情報を取り扱う場合、スマートメータからセンタまで様々な通信経路にて情報が送信されるため、全通信経路にて電力使用量等の情報を暗号化する必要がある。電力使用量等の情報のセキュリティの保護を実現する際、電力業界では事前共有鍵方式が採用されている。また、スマートメータに保管されている鍵の危殆化に対応するために、鍵更新の仕組みも採用されている。

電力業界において、スマートメータ等に関する鍵管理の要件が定められている NIST IR 7628 を調査対象とした。

(2) 調査結果

本調査結果では、鍵のライフサイクルにおける他業界(クレジットカード、銀行、電力)の規格を比較した。

① 鍵の生成

クレジットカード業界(PCI P2PE)、銀行業界(ISO11568-2:2012)、電力業界(NIST IR 7628)における鍵管理の生成フェーズの要件を比較した結果を下記に示す。

(ⅰ)【要件 1】適切な暗号モジュールの利用

NIST SP800-57 part1 では、全ての暗号鍵は適切な暗号モジュールの内部で生成されることが定められている。3 業界の規格を調査した結果（表 3.2b.2-8)、いずれの業界でも該当する要件が存在していることを確認できた。

銀行業界（ISO11568-2:2012）では、安全な暗号モジュール内部での鍵生成については定められているものの、具体的な暗号モジュールの要件は定められていない。その一方で、具体的な暗号モジュールの要件として、電力業界（NIST IR 7628）では、FIPS 140-2 Level2 以上への準拠を要求し、クレジットカード業界（PCI P2PE）では PCI SSC による PCI PTS 認定、もしくは FIPS 140-2 Level3 以上への準拠が定められている。

表 3.2b.2-8　【要件 1】適切な暗号モジュールの利用

業界 （規格）	記載有無	該当要件の要約
クレジットカード （PCI P2PE）	有	全ての暗号鍵は下記の要件のいずれかを満たす暗号モジュールの内部で生成されることが定められている。 ・PCI 認定の Hardware Security Module（以下、HSM という）あるいは Point Of Interaction（以下、POI という）機器であること ・FIPS 140-2 Level 3 以上にて認定された HSM であること
銀行 （ISO11568-2:2012）	有	全ての暗号鍵についての要件は存在しないが、鍵自体の暗号化は、安全な暗号モジュールの内部にて生成されることが定められている。
電力 （NIST IR 7628）	有	全ての暗号鍵は FIPS 140-2 Level2 以上の認定を取得した暗号モジュールの内部にて生成されることが定められている。

(ⅱ)【要件 2】生成する鍵へのアクセス制御

NIST SP800-57 part1 では、単一の個人による、生成時の鍵へのアクセスを防止することが定められている。3 業界の規格を調査した結果（表 3.2b.2-9)、いずれの業界でも該当する要件が存在することが確認できた。

電力業界（NIST IR 7628）では具体的なアクセス制御方法は記載されていないが、全ライフサイクルにおいて鍵が保護されなければならないことが要求されており、生成フェーズでも鍵が保護されなければならない。

クレジットカード業界（PCI P2PE）、銀行業界（ISO11568-2:2012）では、単一個人による鍵の生成は許可されていない。鍵の生成は、知識分割もしくは相互監視によって二人以上で行われることが要件として定められている。

表 3.2b.2-9　【要件 2】生成する鍵へのアクセス制御

業界 （規格）	記載有無	該当要件の要約
クレジットカード （PCI P2PE）	有	鍵の生成プロセスにおいて、相互監視によって鍵の生成が行われることが定められており、単一個人のみで生成してはならない。
銀行 （ISO11568-2:2012）	有	鍵は知識分割と相互監視によって保護されることが定められており、単一個人のみで生成してはならない。
電力 （NIST IR 7628）	有	アクセス制御方法については定められていないが、鍵は生成から有効期間中は保護されることが定められている。

(ⅲ)【要件 3】適切な擬似乱数生成器の利用

NIST SP800-57 part1 では、適切な擬似乱数生成器を利用することが定められている。3 業界の規格を調査した結果（表 3.2b.2-10）、いずれの業界でも該当する要件が存在することが確認できた。

適切な擬似乱数生成器として各業界ともに具体的な要件が定められている。クレジットカード業界（PCI P2PE）、電力業界（NIST IR 7628）では NIST SP800-22 への準拠が要求されており、銀行業界（ISO11568-2:2012）では ISO/IEC 18031 への準拠を一例として定められている。

表 3.2b.2-10　【要件 3】適切な擬似乱数生成器の利用

業界 （規格）	記載有無	該当要件の要約
クレジットカード （PCI P2PE）	有	適切な擬似乱数生成器として NIST SP800-22 に準拠している擬似乱数生成器を利用して乱数生成することが定められている。
銀行 （ISO11568-2:2012）	有	適切な擬似乱数生成器を利用することが求められており、一例として ISO/IEC 18031 に準拠した擬似乱数の算出例が記載されている。
電力 （NIST IR 7628）	有	適切な擬似乱数生成器として NIST が規定している乱数生成関数を使用することが定められている。

(ⅳ)【要件 4】生成する鍵への乱数の利用

NIST SP800-57 part1 では、秘密鍵は推測が困難な適切な乱数、マスタ鍵から生成されることが定められている。3 業界の規格を調査した結果（表 3.2b.2-11）、いずれの業界でも該当する要件が存在することが確認できた。

表 3.2b.2-11　【要件 4】生成する鍵への乱数の利用

業界 （規格）	記載有無	該当要件の要約
クレジットカード （PCI P2PE）	有	擬似乱数を用いて鍵の生成を行うことが定められている。
銀行 （ISO11568-2:2012）	有	擬似乱数を用いて鍵の生成を行うことが定められている。
電力 （NIST IR 7628）	有	擬似乱数を用いて鍵の生成を行うことが定められている。

(ⅴ)【要件 5】再現困難な鍵の生成

NIST SP800-57 part1 では、一つの鍵が危殆化した場合に、危殆化した鍵から他の正規の鍵を再現することが不可能であることが定められている。3 業界の規格を調査した結果（表 3.2b.2-12）、いずれの業界でも該当する要件が存在することが確認できた。

銀行業界（ISO11568-2：2012）では、鍵生成時に乱数を利用することにより鍵を再現できないことが要件として定められている。クレジットカード業界（PCI P2PE）および電力業界（NIST IR 7628）では、鍵生成時の乱数の利用と鍵が書きこまれるデバイスごとにユニークな鍵を使うことで再現できないことをが要件として定められている。

表 3.2b.2-12　【要件 5】再現困難な鍵の生成

業界 （規格）	記載有無	該当要件の要約
クレジットカード （PCI P2PE）	有	一つの鍵が危殆化した場合に、他の鍵を再現できないように、鍵生成時の乱数の利用とデバイスごとにユニークな鍵を使うことが定められている。
銀行 （ISO11568-2:2012）	有	一つの鍵が危殆化した場合に、その鍵から他の鍵を導くことができないように、鍵生成時の乱数の利用が定められている。
電力 （NIST IR 7628）	有	一つの鍵が危殆化した場合に、他の鍵を再現できないように、鍵生成時の乱数の利用とデバイスごとにユニークな鍵を使うことが定められている。

② 鍵の配布

クレジットカード業界（PCI P2PE）、銀行業界（ISO11568-2:2012）、電力業界（NIST IR 7628）における鍵管理の配布フェーズの要件を比較した結果を下記に示す。

(ⅰ)【要件 6】配送先の特定

NIST SP800-57 part1 では、鍵の配送は受取先のなりすまし等が行われていないことを確認し、配送先が確実であると特定できた場合にのみ行われることが定められている。3 業界の規格を調査した結果（表 3.2b.2-13、表 3.2b.2-12)、いずれの業界でも該当する要件が存在することが確認できた。

銀行業界（ISO11568-2：2012）では、配送先が改竄されていないことが確認できた時にのみ配送を行うことが定められている。それに対し、クレジットカード業界（PCI P2PE）と電力業界（NIST IR 7628）では、配送先の確認だけでなく配送元の確認も行うことが定められている。

表 3.2b.2-13　【要件 6】配送先の特定

業界 （規格）	記載有 無	該当要件の要約
クレジットカード （PCI P2PE）	有	鍵の配送先が確実であることを確認するために、配送元と配送先の間で相互機器認証を行うことが定められている。
銀行 （ISO11568-2:2012）	有	鍵の配送先が確実であることを確認するために、配送先が改竄されていないことが保証される場合にのみ鍵のロードを行うことが定められている。
電力 （NIST IR 7628）	有	鍵の配送先が確実であることを確認するために、配送元と配送先の両方から信頼されている鍵のみが配送されることが定められている。

(ⅱ)【要件 7】通信での鍵の配送時の保護

NIST SP800-57 part1 では、通信経路にて配送される鍵が適切に保護されることが定められている。3 業界の規格を調査した結果（表 3.2b.2-14)、いずれの業界でも該当する要件が存在することが確認できた。

クレジットカード業界（PCI P2PE）と銀行業界（ISO11568-2：2012）では、通信経路にて配送される鍵の保護として、機密性の保護のために、鍵暗号化鍵を利用して暗号化した上で配送することが要件として定められている。また、完全性、可用性の確保のために、ISO 11568 に準拠した検証メカニズムを導入することが合わせて定められている。一方、電力業界（NIST IR 7628）では、個別の具体的な要件はないものの、NIST SP800-57 の要件にしたがって鍵が配送されることが定められている。

表 3.2b.2-14　【要件 7】鍵データの保護

業界 （規格）	記載有無	該当要件の要約
クレジットカード （PCI P2PE）	有	通信経路にて配送される鍵は以下の要件に従い、適切に保護されることが定められている。 ・機密性の確保のために、鍵暗号化鍵を利用して暗号化された上で配送され、鍵暗号化鍵は配送される鍵と同等以上の強度を有していなければならないこと ・完全性、可用性の確保のために、ISO 11568 に準拠した検証メカニズムを導入しなければならないこと
銀行 （ISO11568-2:2012）	有	通信経路にて配送される鍵は以下の要件に従い、適切に保護されることが定められている。 ・機密性の確保のために、鍵暗号化鍵を利用して暗号化された上で配送され、鍵暗号化鍵は配送される鍵と同等以上の強度を有していなければならないこと ・完全性、可用性の確保のために、ISO 11568 に準拠した検証メカニズムを導入しなければならないこと
電力 （NIST IR 7628）	有	通信経路にて配送される鍵は NIST SP800-57 の要件に従い、適切に保護されることが定められている。

(ⅲ)【要件 8】通信経路の保護

NIST SP800-57 part1 では、通信経路での鍵のすり替え、改竄等を防ぐために、通信経路を保護することが定められている。3 業界の規格を調査した結果（表 3.2b.2-15、表 3.2b.2-12）、いずれの業界でも該当する要件が存在することが確認できた。

クレジットカード業界（PCI P2PE）では、利用してはならない経路が具体的に定められている。銀行業界（ISO11568-2：2012）では、ISO/IEC11770-2 に準拠したプロセスにて保護されることが定められている。

電力業界（NIST IR 7628）では、個別の具体的な要件はないものの、NIST SP800-57 の要件にしたがって鍵が配送されることが定められている。

表 3.2b.2-15　【要件 8】通信経路の保護

業界 （規格）	記載有無	該当要件の要約
クレジットカード （PCI P2PE）	有	鍵の通信経路として以下は利用してはならないことが定められている。 ・暗号キーあるいは暗号キーコンポーネントの口述筆記 ・ボイスメールに暗号キーあるいは暗号キーコンポーネントを記録 ・ファクシミリや電子メールにより送信 ・正当性の証明が不可能な梱包での運搬 ・画面に平文で表示
銀行 （ISO11568-2:2012）	有	通信経路での鍵は ISO/IEC11770-2 に準拠して保護されることが定められている。
電力 （NIST IR 7628）	有	通信経路は NIST SP800-57 の要件に従い、適切に保護されることが定められている。

(ⅳ)【要件 9】手動での鍵の配送時の保護

NIST SP800-57 part1 では、手動にて配送される鍵が適切に保護されることが定められている。3 業界の規格を調査した結果（表 3.2b.2-16、表 3.2b.2-15、表 3.2b.2-12)、クレジットカード業界（PCI P2PE）および電力業界（ISO11568-2:2012）では、該当する要件が存在することが確認できた。

クレジットカード業界（PCI P2PE）では、配送元、配送先においてアクセス権を有する者のみが物理的に鍵が保管された媒体にアクセスすることが定められており、また、媒体の要件として暗号モジュールを利用すること、耐タンパ性を有し、真正性を証明できることが定められている。

電力業界（NIST IR 7628）では、個別の具体的な要件はないものの、NIST SP800-57 の要件にしたがって鍵が配送されることが定められている。

表 3.2b.2-16　【要件 9】手動での鍵の配送

業界 （規格）	記載有無	該当要件の要約
クレジットカード （PCI P2PE）	有	手動にて配送される鍵は以下の要件に従い、適切に保護されることが定められている。 ・鍵の配送元と配送先の両方においてアクセス権を有する者だけがアクセスできること ・暗号モジュールに格納された上で、耐タンパ性のある真正性の証明が可能なプロセスにて配送すること
銀行 （ISO11568-2:2012）	無	当該項目の要件は存在しない。
電力 （NIST IR 7628）	有	手動にて配送される鍵は NIST SP800-57 の要件に従って保護されることが定められている。

(ｖ)【要件 10】鍵の構成要素のロード、デバイスへの配送

NIST SP800-57 part1 では、鍵の構成要素のロードおよびデバイスへの配送の際には以下の要件に従い、適切に保護されることが定められている。3 業界の規格を調査した結果（表 3.2b.2-17、表 3.2b.2-12)、いずれの業界でも該当する要件が存在することが確認できた。

クレジットカード業界（PCI P2PE）と銀行業界（ISO11568-2:2012）では、鍵の構成要素のロードは相互監視下で行われ、知識分割にて保護されることが定められている。

電力業界（NIST IR 7628）では、個別の具体的な要件はないものの、NIST SP800-57 の要件にしたがって鍵が配送されることが定められている。

表 3.2b.2-17　【要件 10】鍵のロード、デバイスへの配送

業界 （規格）	記載有無	該当要件の要約
クレジットカード （PCI P2PE）	有	鍵の構成要素のロードおよびデバイスへの配送の際には以下の要件に従い、適切に保護されることことが定められている。 ・鍵の構成要素のロードおよびデバイスへの配送は相互監視下で行われること ・鍵の構成要素は知識分割により保護されること
銀行 （ISO11568-2:2012）	有	鍵の構成要素のロードおよびデバイスへの配送の際には以下の要件に従い、適切に保護されることことが定められている。 ・鍵の構成要素のロードおよびデバイスへの配送は相互監視下で行われること ・鍵の構成要素は知識分割により保護されること
電力 （NIST IR 7628）	有	鍵の構成要素のロードおよびデバイスへの配送の際には NIST SP800-57 の要件に従い、適切に保護されることが定められている

③ 鍵の利用

クレジットカード業界（PCI P2PE）、銀行業界（ISO11568-2:2012）、電力業界（NIST IR 7628）における鍵管理の利用フェーズの要件を比較した結果を下記に示す。

(ｉ)【要件 11】鍵へのアクセス時の認証

NIST SP800-57 part1 では、認証等により許可されている者のみがアクセスできる仕組みを用意することが定められている。3 業界の規格を調査した結果（表 3.2b.2-18)、クレジットカード業界（PCI P2PE）でのみ要件が存在することが確認できた。

クレジットカード業界（PCI P2PE）では、鍵へのアクセス権を有している者のみが相互監視下でアクセスできることが定められている。

表 3.2b.2-18　【要件 11】鍵へのアクセス時の認証

業界 （規格）	記載有無	該当要件の要約
クレジットカード （PCI P2PE）	有	アクセス権を有する者によって相互監視がなされている場合にのみアクセスできることが定められている。
銀行 （ISO11568-2:2012）	無	当該項目の要件は存在しない。
電力 （NIST IR 7628）	無	当該項目の要件は存在しない。

(ⅱ) 【要件 12】鍵の更新

NIST SP800-57 part1 では、鍵の有効期間が終了した場合もしくは鍵が危殆化した場合には新しい鍵へ更新することが定められている。3 業界の規格を調査した結果（表 3.2b.2-19）、いずれの業界でも該当する要件が存在することが確認できた。

クレジットカード業界（PCI P2PE）および電力業界（NIST IR 7628）では、鍵更新時の要件として NIST SP800-57 に準拠することが要求されている。銀行業界（ISO11568-2:2012）では、鍵の更新が必要となる鍵の有効期間として ISO/TR 14742 に準拠することが要求されている。

表 3.2b.2-19　【要件 12】鍵の更新

業界 （規格）	記載有無	該当要件の要約
クレジットカード （PCI P2PE）	有	業界のベストプラクティスおよび NIST SP800-57 等の規格に基づいて規定されている鍵の有効期間が終了する前もしくは鍵が危殆化した場合には、新しい鍵へ更新することが定められている。
銀行 （ISO11568-2:2012）	有	ISO/TR 14742 に定められた鍵の有効期間が終了する前もしくは鍵の危殆化が疑わしい場合には、新しい鍵へ更新することが定められている。 新しい鍵が危殆化していない古い鍵を利用して生成されること要件として定められている。
電力 （NIST IR 7628）	有	NIST SP800-57 に従い、鍵の有効期間が終了した場合もしくは鍵が危殆化した場合には新しい鍵へ更新することが定められている。

④ 鍵の保管

クレジットカード業界（PCI P2PE）、銀行業界（ISO11568-2:2012）、電力業界（NIST IR 7628）における鍵管理の保管フェーズの要件を比較した結果を下記に示す。

(ⅰ) 【要件 13】鍵の適切な保管

NIST SP800-57 part1 では、鍵は機密性を確保するために、暗号化の実施、暗号モジュール内、アクセスが管理されているセキュアな保管庫のいずれかに保管されることが定められている。3 業界の規格を調査した結果（表 3.2b.2-20）、いずれの業界でも該当する要件が存在することが確認できた。

クレジットカード業界（PCI P2PE）では、暗号モジュール内部での保管については定められているものの、具体的な暗号モジュールの要件は定められていない。その一方で、具体的な暗号モジュールの要件として、銀行業界（ISO11568-2:2012）では、ISO 13491-1 および ISO 13491-2 への準拠が要求されており、電力業界（NIST IR 7628）では、FIPS 140-2 Level2 以上への準拠が定められている。

表 3.2b.2-20 【要件 13】鍵の適切な保管

業界 （規格）	記載有無	該当要件の要約
クレジットカード （PCI P2PE）	有	鍵は機密性を確保するために、暗号化の実施、暗号モジュール内での保管による保護のいずれかがなされなければならないと定められている。
銀行 （ISO11568-2:2012）	有	鍵は機密性を確保するために、ISO 13491-1 および ISO 13491-2 に準拠する暗号モジュール内に保管されなければならないと定められている。
電力 （NIST IR 7628）	有	鍵は機密性を確保するために、FIPS 140-2 Level2 以上に準拠した暗号モジュール内に保管することが定められている。

(ⅱ)【要件 14】鍵の完全性確保

NIST SP800-57 part1 では、鍵は改竄からの予防、検知、回復を行う仕組みにより保護されることが定められている。3 業界の規格を調査した結果（表 3.2b.2-21）、いずれの業界でも該当する要件が存在することが確認できた。

クレジットカード業界（PCI P2PE）および銀行業界（ISO11568-2:2012）では、鍵の改竄からの保護のために、デジタル署名（ISO 11568-4 準拠）や鍵検証コードを用いることが定められている。一方、電力業界（NIST IR 7628）では、鍵の改竄からの保護のために、FIPS 140-2 Level2 以上に準拠した暗号モジュール内で保管することが定められている。

表 3.2b.2-21　【要件 14】鍵の完全性確保

業界 （規格）	記載有無	該当要件の要約
クレジットカード （PCI P2PE）	有	鍵は改竄からの保護のために、以下の例のような手法を用いることが定められている。 ・デジタル署名（ISO 11568-4 準拠） ・鍵検証コードを活用する
銀行 （ISO11568-2:2012）	有	鍵は改竄からの保護のために、以下の例のような手法を用いることが定められている。 ・デジタル署名（ISO 11568-4 準拠） ・鍵検証コードを活用する
電力 （NIST IR 7628）	有	鍵は改竄からの保護のために、鍵は FIPS 140-2 Level2 以上の認定を取得した暗号モジュールの内部にて保管されることが定められている。

(ⅲ)【要件 15】暗号鍵のバックアップ

NIST SP800-57 part1 では、有効期間中の鍵はバックアップが取得され、厳格な運用を行うことが定められている。3 業界の規格を調査した結果（表 3.2b.2-22）、クレジットカード業界（PCI P2PE）および銀行業界（ISO11568-2：2012）では、該当する要件が存在することが確認できた。

クレジットカード業界（PCI P2PE）では、バックアップ時の要件として相互監視により作成され、バックアップ元と同じ要件にて保管されるという厳格な運用を行うことが定められている。また、銀行業界（ISO11568-2:2012）では、相互監視による作成の要件はないものの、バックアップ元と同じ要件にて保管されることが定められている。

表 3.2b.2-22　【要件 15】暗号鍵のバックアップ

業界 （規格）	記載有無	該当要件の要約
クレジットカード （PCI P2PE）	有	有効期間中の鍵のバックアップは相互監視により作成され、バックアップ元と同じ要件で保管されることを定められている。
銀行 （ISO11568-2:2012）	有	有効期間中の鍵のバックアップが作成され、暗号モジュールかもしくは暗号化されて保管されることが定められている。
電力 （NIST IR 7628）	無	当該項目の要件は存在しない。

⑤ 鍵の廃棄

クレジットカード業界（PCI P2PE）、銀行業界（ISO11568-2:2012）、電力業界（NIST IR 7628）における鍵管理の廃棄フェーズの要件を比較した結果を下記に示す。

(ⅰ)【要件 16】鍵の廃棄

NIST SP800-57 part1 では、鍵の有効期間の終了時、あるいは鍵が危殆化した場合、鍵の全てのコピーを含め確実に廃棄されることが定められている。3 業界の規格を調査した結果（表 3.2b.2-23)、いずれの業界でも該当する要件が存在することが確認できた。

電力業界（NIST IR 7628）では、鍵の有効期間の終了時、あるいは鍵が危殆化した場合にリアルタイムに鍵を廃棄することが定められているが、具体的なその方法は定められていない。クレジットカード業界（PCI P2PE）および銀行業界（ISO11568-2：2012）では、ISO-9564 あるいは ISO11568 に準拠した手続きに従って確実に廃棄されることが定められている。

表 3.2b.2-23　【要件 16】暗号鍵の廃棄

業界 （規格）	記載有無	該当要件の要約
クレジットカード （PCI P2PE）	有	鍵の有効期間の終了時、あるいは鍵が危殆化した場合、相互監視の下、ISO-9564 あるいは ISO11568 に準拠した手続きに従って確実に廃棄されることが定められている。
銀行 （ISO11568-2:2012）	有	鍵の有効期間の終了時、あるいは鍵が危殆化した場合、ISO9564-1 に準拠した手続きに従って廃棄されることが定められている。
電力 （NIST IR 7628）	有	鍵の有効期間の終了時、あるいは鍵が危殆化した場合、リアルタイムに物理的または電子的手段で回復できないように廃棄することが定められている。

(ⅱ)【要件 17】鍵の失効通知

NIST SP800-57 part1 では、鍵の利用を停止する際は、全ての鍵の管理者に鍵の失効が通知されることが定められている。3 業界の規格を調査した結果（表 3.2b.2-24)、電力業界（NIST IR 7628）でのみ該当する要件が存在することが確認できた。

電力業界（NIST IR 7628）では、鍵の危殆化が疑われる場合に鍵の失効通知および、その後の対応プロセスが定められている。

表 3.2b.2-24　【要件 17】暗号鍵の失効通知

業界 （規格）	記載有無	該当要件の要約
クレジットカード （PCI P2PE）	無	当該項目の要件は存在しない。
銀行 （ISO11568-2:2012）	無	当該項目の要件は存在しない。
電力 （NIST IR 7628）	有	鍵の有効期間内において、鍵の危殆化が疑われた場合での鍵の失効の通知およびその後の対応プロセスを確立することが定められている。

⑥ 鍵の回復

クレジットカード業界（PCI P2PE）、銀行業界（ISO11568-2:2012）、電力業界（NIST IR 7628）における鍵管理の回復フェーズの要件を比較した結果を下記に示す。

(ⅰ)【要件 18】暗号鍵の回復

NIST SP800-57 part1 では、鍵へのアクセスができなくなった場合に備え、バックアップからの回復フェーズにおける運用方針が事前に策定されることが定められている。3 業界の規格を調査した結果（表 3.2b.2-25)、銀行業界（ISO11568-2:2012）でのみ該当する要件が存在することが確認できた。具体的には、バックアップを取得し、回復フェーズでは配布フェーズの要件に従った運用を実施することが定められている。

表 3.2b.2-25　【要件 18】暗号鍵の回復

業界 （規格）	記載有無	該当要件の要約
クレジットカード （PCI P2PE）	無	当該項目の要件は存在しない。
銀行 （ISO11568-2:2012）	有	鍵の回復時は配布フェーズの要件に従った運用を実施する方針が、事前に定められている。
電力 （NIST IR 7628）	無	当該項目の要件は存在しない。

(3) 考察

本項では、鍵のライフサイクルにおける、クレジットカード業界、銀行業界、電力業界における要件を比較した。その結果、3 業界で共通して定められている要件を明らかにすることができた。共通して定められている要件は、今後、自動車業界でも定められるべき要件と考えられる。

以降は、鍵のライフサイクルのフェーズごとに自動車業界において定められるべき要件を考察する。

① 生成フェーズ

クレジットカード業界、銀行業界、電力業界では、NIST SP800-57 part1 の生成フェーズにおける全要件が定められていることが確認できた。業界ごとに定められている要件の内容には差異が認められ、中でもクレジットカード業界では、鍵の生成時に利用される暗号モジュール、擬似乱数生成器および生成される鍵のアクセス制御の要件が具体化されている。

自動車業界でも他業界同様、全要件が定められることが望ましく、要件の内容は今回調査した業界が参照している規格を参考とすることが有効であると考えられる。

② 配送フェーズ

クレジットカード業界、電力業界では、NIST SP800-57 part1 の配送フェーズにおける全要件が定められていることが確認できた。銀行業界では、手動での鍵の配送時の保護について該当する要件が存在しなかったものの、それ以外の要件は定められている。

自動車業界でも、基本的には通常の生産工程を考えると先行する業界と同様、要件 6、要件 7、要件 8 に記載された内容を取り込んで通信による配送が望ましい。しかしながら、市場に出てからのディーラ、サービス工場等での作業現場においては、適切な通信手段の確保が困難な地域も存在する。そういった自動車の使われる場面全体を考慮すると、手動での鍵の配送が必要な場合も要件に定められることが望ましい。

クレジットカード業界、電力業界同様、全要件が定められることが望ましく、要件の内容は今回調査した業界が参照している規格を参考とすることが有効であると考えられる。

③ 利用フェーズ

クレジットカード業界では、NIST SP800-57 part1 の利用フェーズにおける全要件が定められていることが確認できた。銀行業界、電力業界では、鍵へのアクセス時の認証については該当する要件が存在しなかったものの、鍵の更新は要件として定められている。

自動車業界においても、鍵の更新や利用中の鍵へのアクセスが必要であると想定されるため、クレジットカード業界同様、全要件が定められることが望ましい。要件の内容は今回調査した業界が参照している規格を参考とすることが有効であると考えられる。

④ 保管フェーズ

NIST SP800-57 part1 の保管フェーズにおける全要件が、クレジットカード業界、銀行業界では定められていることが確認できた。

自動車業界においても、鍵の適切な保管と完全性の確保については、ECU への鍵書込みを想定すると必要な要件であり、今回調査した業界が参照している規格を参考とすることが有効であると考えられる。

⑤ 廃棄フェーズ

電力業界では、NIST SP800-57 part1 の廃棄フェーズにおける全要件が定められていることが確認できた。クレジットカード業界、銀行業界では、暗号鍵の失効通知について要件は定められていなかったものの、暗号鍵の廃棄については定められている。

自動車業界においては、ライフサイクル全体を考慮すると、今回調査した他業界と異なる場面がいくつか存在する。中古車の流通はその事例のひとつである。ECU が物理事象を制御する際、物理的なデータ送受信のメッセージ認証等で使われる鍵であれば、車両の持ち主が変わっても鍵の廃棄・更新が必須要件となる可能性は低いと想定している。しかしながら、前の持ち主の個人情報を保護するような、あるいは個人情報に関与するような通信のセキュリティ対策で使われていた鍵は、適切な廃棄処置が重要な要件になると考えられる。また、廃車時にも車両は不特定の者の手に渡るため、鍵はその利用目的に応じて、有効期限切れの扱いにするものや、秘匿性を保ったまま回復できないように廃棄すべきも

の等に層別しておく必要がある。そのため、自動車業界においても電力業界同様、全要件が定められ、鍵の利用目的に応じて適切な処置がなされることが望ましい。

⑥ 回復フェーズ

銀行業界では、NIST SP800-57 part1 の回復フェーズにおける要件が定められていることが確認できた。クレジットカード業界（PCI P2PE）、電力業界（NIST IR 7628）では、回復フェーズに関する要件は定められていない。

自動車業界でもバックアップからの鍵の回復が必要な場合は、銀行業界同様、要件が定められることが望ましく、要件の内容は銀行業界を参考とすることが有効であると考えられる。

(4) まとめ

自動車における鍵管理は ECU 内に限らず、鍵を生成するセンター、ECU に鍵を書き込む ECU 工場、車両工場、市場における整備一般をこなすディーラやサービス工場、中古車の流通等、今回調査を行った他業界に比べると様々な現場が存在している。その様々な現場において通信経路の安全性やユースケースごとに要件を定めていかないと、車両のセキュリティが確保できなくなる懸念がある。この自動車業界全体の構図に対しては、自動車メーカ各社が個別に対応するのではなく、業界で共通に利用できる指針、あるいは規格の存在が望ましい。

本調査の対象とした他業界の中では、特に P2PE では NISTSP800-57 Part1 の要求事項を包含しつつも、業界での業務や実装されるアプリケーションを想定したユースケースごとの鍵管理要件が、技術要件、運用要件に渡り詳細に記載されていた。今後、自動車業界において鍵管理の要件を充実させるにあたっては、P2PE のように業務や実装されるアプリケーションを想定した規格を参考にすることが大いに有効になると考えられる。

3.2b.3 車内システム（鍵管理）に対する評価方法・評価技術の検討

先の項では、つながる車の車内システムにおいて、メッセージ認証などの暗号鍵を必要とするセキュリティ対策が導入されることから、鍵とその運用面について他業界の調査を行った。自動車と同様に、事前共有鍵方式を用いて鍵更新を行う他業界を調査対象した結果から、鍵管理には通信経路の暗号化と言った技術面だけでなく、鍵の生成から配送、保管、廃棄に至るまでのライフサイクル全体に渡る運用面も重要であると認識した。

平成 28 年度において取り組む「車内システムにおける鍵管理」では、具体的には車載マイコンに内蔵されるセキュリティハードウェア（以降、セキュリティ IP と称する）を使用した鍵配布手順を利用する。技術的にはこのセキュリティ IP を利用することで、セキュアな鍵の配送、鍵をマイコン内蔵メモリにセキュアに保管すると言ったことが、ある一定レベルで担保されることが期待できる。しかしながら、このセキュリティ IP を利用しても、書き込む鍵そのものの強度が十分でなかったり、運用面が適切でない場合には、鍵情

報の漏洩など重要な問題につながる脆弱性を作り込む懸念となる。

そこで、平成 28 年度本テーマで開発する評価対象は、コンポーネントの評価対象で設けた、乱数のエントロピーで見られるような技術的水準だけでなく、鍵管理における運用面の評価も可能なシステムを開発する。

(1) 開発対象の概要

本テーマで開発する評価対象（車内システム）の概要について記述する。

① 車内システムの概要

コンポーネントとしての標準 ECU 群で実現する車内システムは、自動走行機能を構成するという性質を鑑み、各標準 ECU との連携という観点において、安全面および信頼面について慎重に配慮しなければならない。これを IPA が提唱しているソフトウェア品質 6 特性（機能性、信頼性、使用性、効率性、保守性、移植性）の観点から分析すると、上記 6 特性の中でも機能性に含まれるセキュリティ（Security）に着目する必要がある。その理由を以下に記す。

- 標準 ECU が外部の情報ネットワークに Gateway（Firewall）を介して接続される場合、情報ネットワークを経由した脅威の対象になることが想定されるため。
- 特に ECU への通信路および通信データを改変するような脅威が発生し、その攻撃が成功した場合、ECU の不正動作や無効化が想定されるため。

上記理由のいずれも各標準 ECU や自動走行機能に対して重大な影響を与えるものである。設計者はこれらの影響を抑止するためにセキュリティ特性に着目し、ECU が使用する通信経路上のデータ保護を可能とするセキュア通信モデルを構築する必要が有る。

セキュア通信モデルにおいては、通信データを保護するために暗号化手順を使用する。この暗号化手順では各種の鍵データを使用する（例：認証鍵や検証鍵）。この鍵データが、何らかの手段によって暴露された場合、セキュア通信モデルは危殆化する。この事態に備えて、予め ECU あるいは車両生産時にペアの鍵データを標準 ECU に配布しておいたり、事態発生時には、市場で新しい鍵データを配布する必要がある。この鍵データを安全に配布するために、安全な手段を規定する必要がある。この手段を上記ソフトウェア品質 6 特性の観点から見ると、機能性に含まれるセキュリティと、使用性に含まれる運用性および使用性標準適合性に該当する。

本テーマにおいて開発する標準 ECU としての車内システムは、上記特性に対する評価が有効となるような機能として、安全な鍵データを配布する手段（以降、鍵配布手順と称する）に対する評価が実施できる機能を実装する。

開発する車内システムの環境構成図を図 3.2b.3-1 に記す。評価の実施に必要な機能として、車内システム外の枠内には次の複数の機能を実装する。

・鍵データを生成する車外の機能
・生成された鍵データを車外の外部機器から受け取り標準 ECU へ配布する、車内システム機能の一部

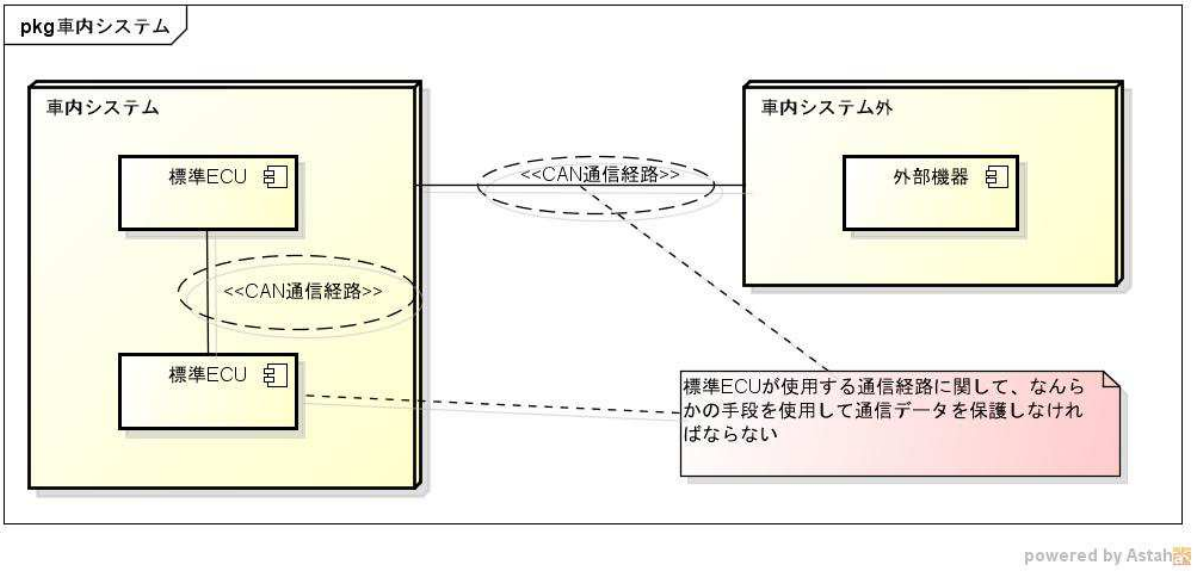


図 3.2b.3-1　車内システム環境構成図

② デバッグ環境全体

車内システムの開発期間および評価期間では図 3.3b.3-2 の動作環境を使用した。E1 エミュレータを使用する環境はマイコン評価用として実績がある環境構成である。

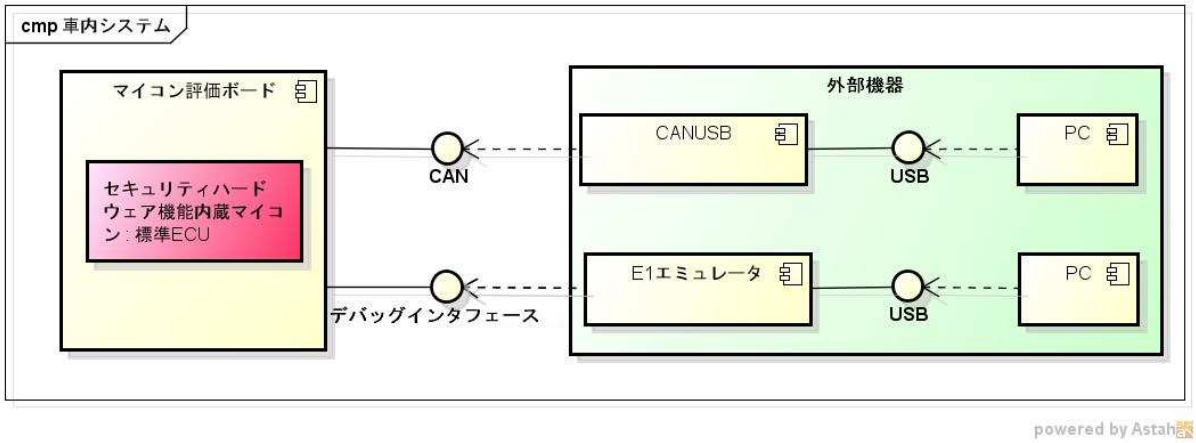


図 3.2b.3-2　車内システムデバッグ環境構成図

また両期間中においては、鍵配布手順を実現する外部機器として CANUSB（Lawicel 製）を使用する。これは鍵配布手順における CAN 通信動作を実現することを目的とする。

(2) 要求仕様

本テーマで開発する評価対象（車内システム）への要求仕様について記述する。

① 鍵配布手順仕様

評価対象（車内システム）に必要な機能を以下に記す。

・セキュリティ評価の対象となるような鍵配布手順を実現する機能
・マイコンに内蔵されるセキュリティ IP を使用した鍵配布手順を実現するために必要な、セキュリティ IP が提供する各種機能を使用するためのインタフェース機能

上記鍵配布手順は、実際の ECU に近い形式で実装する。その為、基本的な ECU の機能やインタフェースを実現できる車載マイコン環境を用意する。パソコンおよびエミュレータ環境上での仮想動作による機能実現は行わない。

評価対象（車内システム）の要件を表 3.2b.3-1 に記す。

表 3.2b.3-1　評価対象（車内システム）要件一覧

項目	説明
マイコン	一般的な 32bit 車載マイコン。セキュリティ IP を内蔵していること。 （例: ルネサス エレクトロニクス株式会社製 RH850 F1L/premium）。
通信規格	一般的な ECU 間通信、診断、リプログラミング、デバッグ等で用いられる CAN 通信。
評価ボード	使用する車載マイコンに対応した機能評価ボード。 CAN 通信インタフェースおよびデバッグポートを備えること。 動作状態を目視できる表示器（LED、LCD 等）を備えること。

※注: CAN ネットワーク仕様に関しては ISO15765-2 Road vehicles - Diagnostics on Controller Area Networks （CAN） - Part 2: Network layer services に準拠する。

② デバッグインタフェース仕様

車内システムにおけるデバッグインタフェース仕様は、コンポーネント開発におけるデバッグインタフェース仕様と同一である。よって本項での記述は割愛する。

参照：3.2b.1 コンポーネント（リプログラミング）に対する評価方法・評価技術の検討 (3) 要求仕様 ④デバッグインタフェース仕様

③ 脆弱性水準

車内システムでの鍵配布手順においては、機能実装レベルでの脆弱性水準を設定することができない。これはセキュリティ IP が提供する機能により、鍵配布手順がハードウェアで固定化されていること、およびそれらの機能がソフトウェアレベルで開示されていないためである。そのため、ある一定水準の技術によって鍵配布のセキュリティ性は担保されており、人為的ミスを作り込む余地が減っている。

これは、本項のはじめに述べた通りであるが、鍵管理は技術だけでなく運用面と一体に

なって脆弱性水準が決まるものであり、車内システムが提供する鍵配布手順への脆弱性水準は、以下のとおり鍵の生成を含めた運用レベルの観点で設定する。

本テーマで開発する評価対象は、評価者が設定した任意の鍵を書き込むことが可能である。また、鍵の書込みに必要な認証鍵も任意に設定することができる。そのため、意図的にエントロピーの小さい鍵を生成して書き込む、あるいは評価用認証鍵の種類を減らす等して、脆弱性水準を操作することができる。

(3) ソフトウェア仕様

本テーマで開発する評価対象(車内システム)を実現するソフトウェアの仕様について、構成要素ごとに表 3.2b.3-2 に記述する。基本構造は、セキュリティ評価コンポーネントとして先に開発したソフトウェアに準じる。

表 3.2b.3-2　鍵配布手順評価向けソフトウェア構成要素一覧

要素名	説明
Boot/Loader 部	マイコンリセット時に実行を開始するモジュール。マイコンおよび周辺ペリフェラルの初期化、および ISO14229-1 診断サービスモジュール起動またはアプリケーションの起動を行う。
Application 部	リプログラミング対象となるモジュール。本評価向けソフトウェアでは、一定周期で LED 点滅または点灯を行う機能（ON/OFF パターンは複数ある）を実装する。また CAN インタフェースによる診断メッセージの受信処理を実装する。
ISO14229-1 診断サービスモジュール	ISO14229-1 リプログラミング手順を実現するためのモジュール。診断サービスに依存する処理は本モジュール内に実装する。
セキュリティモジュール	鍵配布手順処理で必要となるセキュリティ IP が提供する機能を使用するためのインタフェース機能を提供する。

※セキュリティモジュール以外の要素仕様は、セキュリティ評価コンポーネント向けソフトウェアと同一である。

車内システムにおいて実現する鍵配布手順仕様については、セキュリティ IP が提供する機能仕様に依存する。鍵配布手順の概要を図 3.2b.3-3 に記す。

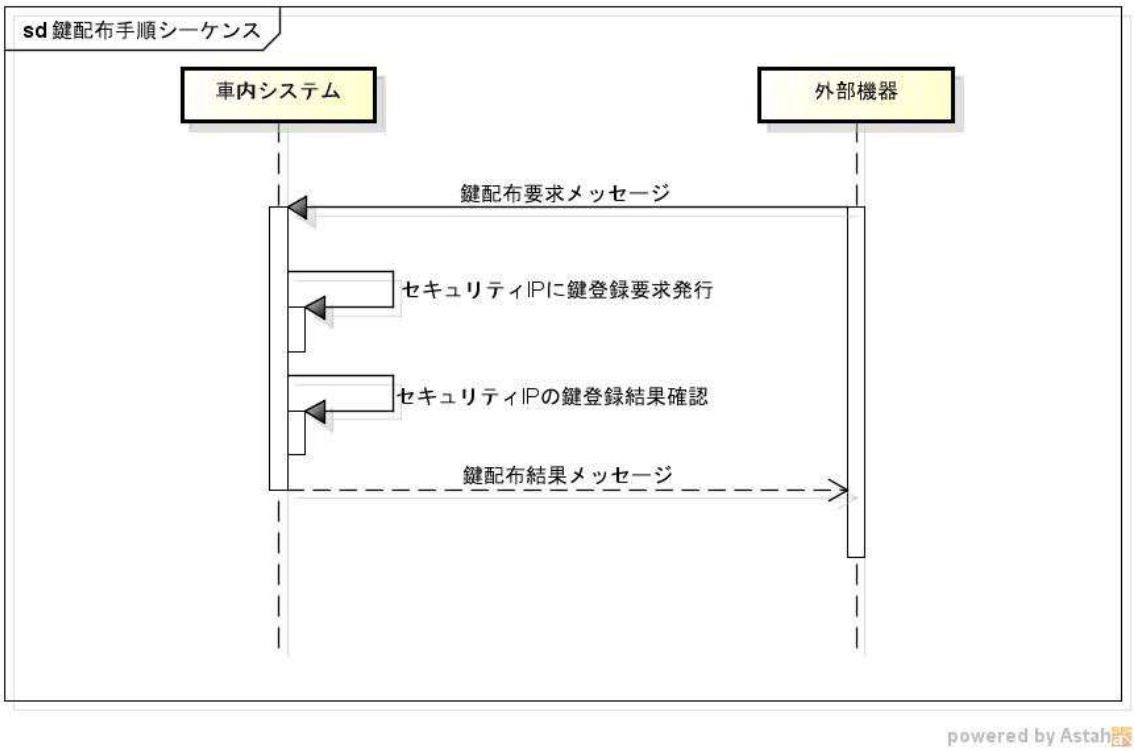


図 3.2b.3-3　鍵配布手順シーケンス

上記のシーケンスが示すように、外部機器から発行された鍵配布要求メッセージは、車内システム内において加工せずに、セキュリティ IP へ渡している。セキュリティ IP は渡されたデータに対しての秘密演算を行い、演算結果が正常であればセキュリティ IP 内部に鍵データを格納する。

セキュリティ IP への要求発行後、車内システムはセキュリティ IP に対して登録結果の完了状態をモニタリングする。登録結果完了を検出した後は、セキュリティ IP から取得した登録結果に従って鍵配布結果メッセージを外部機器に通知する。

外部機器は、鍵配布結果メッセージの内容により、鍵配布要求の実行結果を知ることができる。

(4) 鍵配布手順評価

車内システムソフトウェアが提供する鍵配布手順に対する評価の観点を以下に記す。

・先に規定する鍵配布手順を使用して、任意の鍵の値を正常に配布できること。

鍵配布手順評価対象の構成を図 3.2b.3-4 に記す。

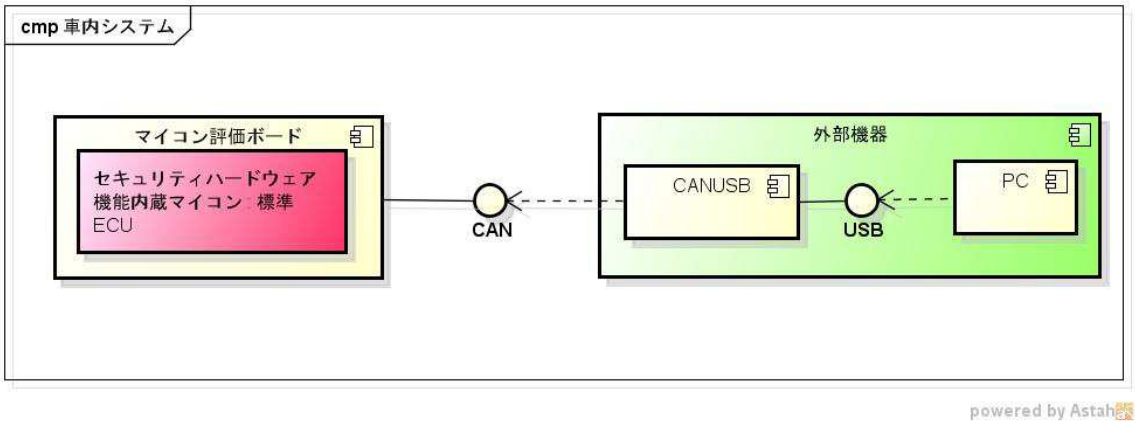


図 3.2b.3-4　鍵配布手順評価対象の構成

本評価では、外部機器として Lawicel 製 CANUSB を使用する。CANUSB は USB-CAN 間の通信データを変換する機器である。

さらに PC 上で動作する鍵配布ツールを使用する。このツールはセキュリティ IP が提供する鍵配布機能に基づいた手順を実施する。その際に通信経路として PC の USB ポートに接続された CANUSB を使用して、CAN 通信を行う。

鍵配布手順評価対象の構成で説明した CANUSB の実機接続例を図 3.2b.3-5 に記す。

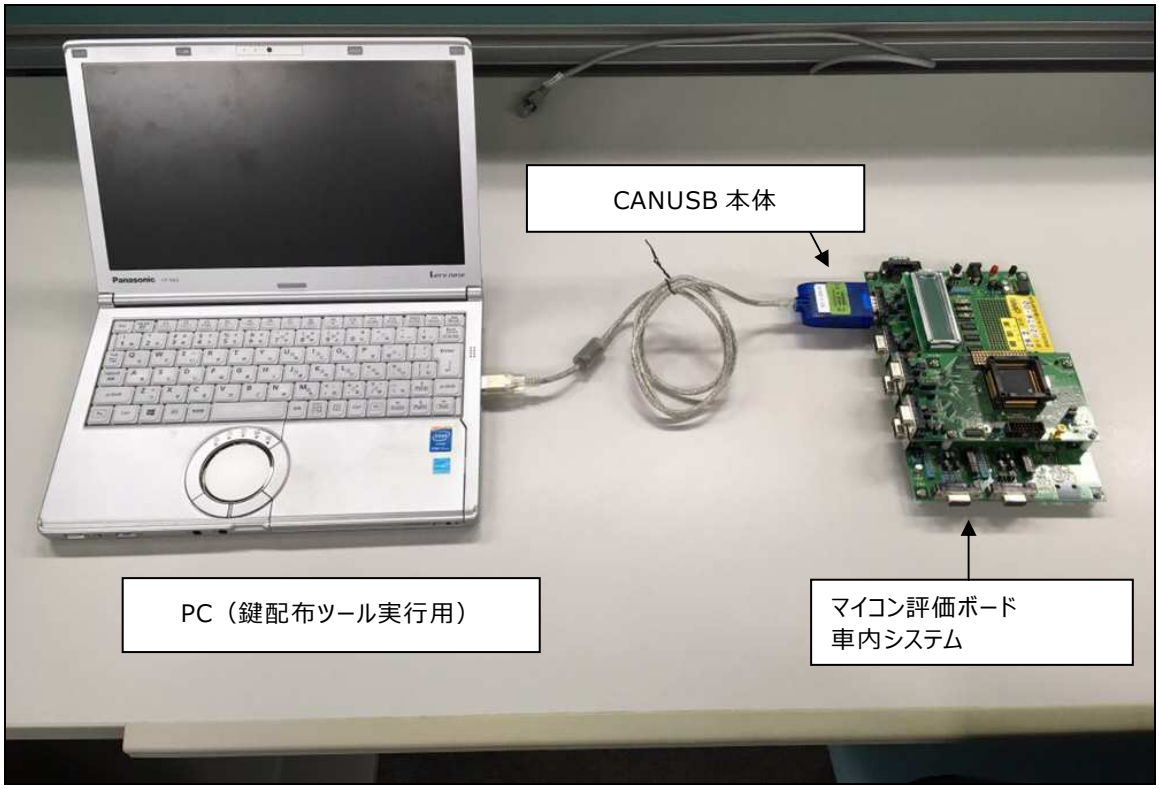


図 3.2b.3-5　CANUSB を用いた鍵配布手順評価対象の実機接続例

鍵配布ツールは、鍵配布手順の結果を CAN 通信ログとしてログファイルに出力する。ログ出力例を図 3.2b.3-6 に記す。

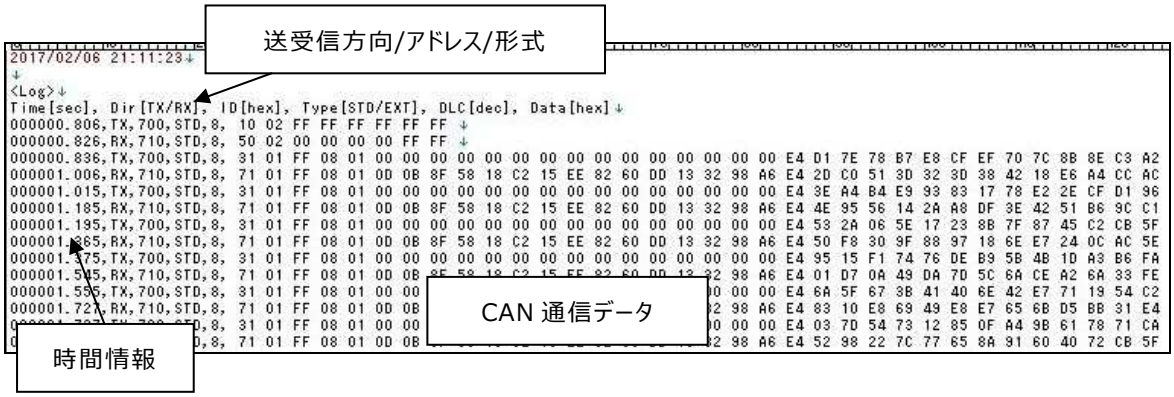


図 3.2b.3-6　鍵配布ツールログ出力例

(5) まとめ

つながる車のコンポーネントと車内システムに対して、セキュリティ対策の妥当性を定量的に確認可能な評価技術を開発すべく、これまでコンポーネントにおける評価方法・評価基準案の検討、評価対象となる標準的な ECU の開発を行ってきた。

平成 28 年度では、平成 27 年度に開発した標準 ECU コンポーネントを利用して車内システムの評価に向けて機能拡張した。セキュリティ IP が提供する機能を使用する鍵配布機能の開発を行い、鍵管理の運用面も含めた評価が可能な評価対象を開発した。

参考文献

[1] 安全な暗号鍵のライフサイクルマネージメントに関する調査、
https://www.ipa.go.jp/files/000013896.pdf
[2] 2010 年度版リストガイド（鍵管理）
https://www.cryptrec.go.jp/report/c10_guide2010_keymanagement_final.pdf
[3] ISO 11568
http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=34937
[4] クレジットカードの台数予測
http://www.meti.go.jp/meti_lib/report/2015fy/000432.pdf
[5] ATM の台数予測
http://data.imf.org/?sk=E5DCAB7E-A5CA-4892-A6EA-598B5463A34C
[6] スマートメーターの台数予測
http://www.meti.go.jp/committee/sougouenergy/denryoku_gas/kihonseisaku/pdf/001_07_01.pdf
[7] B-CAS カードの台数予測、https://www.b-cas.co.jp/co_info/finance/
[8] SIM カードの台数予測、http://www.tca.or.jp/database/
[9]鉄道の台数予測、https://www.jreast.co.jp/order/procurement/rolling_stock.html

3.2b.4　付録

A　用語

ATM（Automated/Automatic Teller Machine）

紙幣（及び硬貨）、通帳、磁気カード・IC カードの受入口、支払口を備え、金融機関や貸金業者、現金出納を行う業者の提供するサービスが、顧客自身の操作によって取引できる機械。

CRYPTREC（Cryptography Research and Evaluation Committees）

日本発の暗号技術評価プロジェクト。

暗号の安全性に関する情報を提供することを目的として、共通鍵暗号、公開鍵暗号、ハッシュ関数、擬似乱数生成系の 4 種類の暗号技術に対し公募を行い、それぞれに対して国内外の暗号研究者による評価を行い、評価レポートや推奨可能な暗号のリストを作成している。

また、既存の 4 種類の暗号技術に加え暗号利用モード、メッセージ認証コード、エンティティ認証に対しても評価を行った。

FIPS 140-2

暗号モジュールに関するセキュリティ要件の仕様を規定する米国連邦標準規格であり、2001 年 5 月 25 日に発行されたものが FIPS 140-2 であり、米国連邦政府の省庁等各機関が利用する、ハードウェア及びソフトウェア両方を含む "暗号モジュール" に関する要件を規定したもの。

HSM（Hardware Security Module）

暗号鍵ライフサイクルの保護に特化して設計された専用の暗号化プロセッサ。HSM は、暗号化によるデータ保護の信頼の基盤として機能し、暗号鍵を安全に管理し、処理し、強化された耐タンパ性のデバイス内に保管することができる。

IPA

日本における IT 国家戦略を技術面、人材面から支えるために設立された、経済産業省所管の中期目標管理法人たる独立行政法人。

ISO 9564

金融サービスにおける個人識別番号 （PIN：Personal Information Number）管理およびセキュリティの国際標準規格。

ISO/IEC 18031

擬似乱数生成に必要とされる要件および実行方法を定めた国際標準ガイドライン。

ISO/IEC11770-2

対象暗号を用いて事前共有鍵を確立するためのメカニズム（鍵の配送・鍵の確立のための条件設定）を定めた国際標準ガイドライン。

ISO/TR 14742

金融サービスにおける金融サービスで使用する暗号アルゴリズムの推奨リストを定めた国際標準規格。主に暗号アルゴリズムと鍵長に関する推奨事項を提供している。

ISO11568-2:2012

金融サービスにおける共通鍵のライフサイクルにおける鍵管理を定めた国際標準規格。主な対象として automated teller machine（ATM）等がある。

MAC（Message Authentication Code）

メッセージ認証符号の略で、メッセージを認証するための短い情報。

NIST（National Institute of Standards and Technology）

科学技術分野における計測と標準に関する研究を行う米国商務省に属する政府機関。

NIST IR 7628

NIST により 2010 年に発表されたスマートグリッドに関するセキュリティガイドライン。

NIST SP800-22

暗号化アプリケーションのための、乱数と擬似乱数生成器の統計学的なテストケース等がまとまったフレームワーク

NIST SP800-57

NIST が 2005 年に SP（Special Publication）シリーズとして発行した、暗号鍵管理の全体像を示したフレームワーク。暗号鍵管理におけるベストプラクティスや特定アプリケーションでの鍵管理の方式について書かれている。

PCI PTS（Payment Card Industry Pin Transaction Security）

クレジットカード情報を扱う加盟店において、PIN コード入力を行う端末で確保されるべきセキュリティ要件。

PCI P2PE（Payment Card Industry Point-to-Point Encryption）

PCI SSC により制定、運用、管理されている国際的な情報セキュリティの基準。クレジット端末から決済ネットワークの手前まで、クレジットカード情報を暗号化した状態で処理するための高レベルのセキュリティ機能が備わった決済の基準。

PCI SSC（Payment Card Industry Security Standards Council）

American Express、Discover Financial Services、JCB International、MasterCard、Visa Inc.によって設立された、PCI 関連基準の策定・維持、評価手順の確立、認定審査会社の教育・試験等を実施している管理団体。

POI（Point Of Interaction）

加盟店端末装置の略。カードからデータを読み取る最初のポイントのことを言う。POI は、ハードウェアとソフトウェアで構成される電子取引認識製品であり、カード会員がカード取引を行うことができるようにする。

誤り訂正符号

データに符号誤り（エラー）が発生した場合にそれを検出、あるいは検出し訂正（前方誤り訂正）すること。

鍵の改竄

権限を持たないものが鍵を書き換えてしまう行為。

鍵の危殆化

考案された当時の鍵の暗号研究の水準やコンピュータの処理能力では容易に解読できなかった鍵の暗号アルゴリズムが、新しい攻撃手法の発見やコンピュータ性能の飛躍的な向上により、十分に安全とは言えなくなること。

鍵の構成要素

鍵を管理（生成・廃棄等）する際に必要な乱数生成等の元となる鍵の長さの値やテンプレート等を指す。

公開鍵暗号

暗号化と復号に別個の鍵（手順）を使い、暗号化の為の鍵を公開

できるようにした暗号方式。

相互監視

組織における不正を防止する仕組み。お互いの作業を監視することにより、特定の従業員の操作・権限の集中や、広範な裁量の付与を避けることが目的。

知識分割

2 つ以上の事業体が別々に暗号鍵の構成要素・部品を持っており、個々の知識では暗号鍵を生成できないようにした状態を指す。

デジタル署名

書面上の手書き署名のセキュリティ特性を模倣するために用いられる公開鍵暗号技術の一種。

伝送誤り

データ等の伝送の際に、伝送内容を誤って送信すること。

認証

そのものの正当性を検証すること。利用する際にその権利があるかどうかやそのものの正しさを確認すること。

マスタ鍵

鍵を暗号化するために使用される主となる鍵。マスタ鍵が脆弱な状態となった場合、生成した鍵全ての信頼性がなくなってしまう。

ユニークな鍵

大量に生成される鍵の内容がそれぞれ異なり、一意性のある鍵のこと。

暗号モジュール

暗号化機能やハッシュ機能、署名機能等のセキュリティ機能を実装したハードウェア、ソフトウェア等から構成されるプログラムの総称。

擬似乱数生成器

乱数列（乱数）のように見えるが、実際には確定的な計算によって求めている擬似乱数列による乱数を生成する機器・装置。

暗号学的に安全な擬似乱数生成器は、一様分布性（出力される乱数系列に 0 と 1 の偏りがないこと）や、予測不可能性（過去の乱数系列から将来の乱数系列を予測できないこと）の性質を満たす乱数を生成する。

事前共有鍵方式

通信を暗号化する際に、暗号鍵を事前に別の手段で交換して共有しておく方式。

B 評価対象（コンポーネント）資料

・マイコン評価ボード LED 点灯パターン一覧

評価向け標準 ECU の動作状態は LED の点灯、消灯パターンで目視することが可能である。各標準 ECU の動作状態と対応する LED 点灯、消灯パターンを表 3.2b.4-1 に記す。

表 3.2b.4-1　マイコン評価ボード LED 点灯パターン一覧

状態	LED 点灯パターン
リプログラミングモジュール動作中または リプログラミング手順開始待ち。	単一の LED のみ、約 500ms 周期で点滅する。
サンプルアプリケーション 1 実行中。	LED1 と 2 が点灯。3 と 4 が消灯。※注
サンプルアプリケーション 2 実行中。	LED1 と 2 が消灯。3 と 4 が点灯。※注

※注：使用するマイコン評価ボードによって、点灯する LED の組み合わせが異なる場合がある。

・セキュリティアクセス脆弱性水準切り替えパラメーター一覧

評価ソフトウェアのセキュリティアクセス脆弱性水準を切り替えるためのパラメータは要求メッセージ送信側で設定する。本評価の場合、要求メッセージ送信側は外部機器ツールとなる。各性水準に対応したセキュリティアクセス要求メッセージを表 3.2b.4-2 に記す。

表 3.2b.4-2　セキュリティアクセス脆弱性水準切り替えパラメーター一覧

セキュリティアクセス 脆弱性水準	SecurityAccess/SEED 要求メッセージ （図 3.2b.1-5 参照/ メッセージ長 2byte：16 進）	SecurityAccess/KEY 要求メッセージ （図 3.2b.1-5 参照/ メッセージ長 34byte：16 進）
生成乱数空間長 128bit / 乱数生成器初期化有り	27 01	27 02 xx xx xx… （xx: 認証データ 32byte）
生成乱数空間長 4bit （先頭から 124bit は 0）/ 乱数生成器初期化有り	27 15	27 16 xx xx xx… （xx: 認証データ 32byte）
生成乱数空間長 128bit / 乱数生成器初期化無し	27 25	27 26 xx xx xx… （xx: 認証データ 32byte）
生成乱数空間長 4bit （先頭から 124bit は 0） / 乱数生成器初期化無し	27 35	27 36 xx xx xx… （xx: 認証データ 32byte）
生成乱数空間長 128bit /マイコン内蔵のセキュリティハードウェア機能による乱数生成 / 乱数生成器初期化有り	27 05	27 06 xx xx xx… （xx: 認証データ 32byte）

・セキュリティ評価向け eSOL ELYZER モデル一覧

セキュリティ評価向けに作成した eSOL ELYZER モデルを表 3.2b.4-3 に記す。

表 3.2b.4-3　セキュリティ評価向け eSOL ELYZER モデル一覧

対応する脆弱性水準	モデル名	実行するリプログラミング手順
生成乱数空間長 128bit /ソフトウェアによる乱数/ 乱数生成器初期化有り	SIP-SECURITY-Pattern1-001-C	リプログラミング正常手順（DSC→SA SEED→SA KEY → RC → RD → TD → RTE → ER）
	SIP-SECURITY-Pattern1-002-C	セキュリティアクセスループ①（DSC→SA SEED→SA KEY）
	SIP-SECURITY-Pattern1-003-C	セキュリティアクセスループ②（DSC→SA SEED→SA SEED）
生成乱数空間長 4bit（先頭から 124bit は 0）/ソフトウェアによる乱数生成 / 乱数生成器初期化有り	SIP-SECURITY-Pattern2-001-C	リプログラミング正常手順（DSC→SA SEED→SA KEY → RC → RD → TD → RTE → ER）
	SIP-SECURITY-Pattern2-002-C	セキュリティアクセスループ①（DSC→SA SEED→SA KEY）
	SIP-SECURITY-Pattern2-003-C	セキュリティアクセスループ②（DSC→SA SEED→SA SEED）
生成乱数空間長 128bit /ソフトウェアによる乱数生成 / 乱数生成器初期化無し	SIP-SECURITY-Pattern3-001-C	リプログラミング正常手順（DSC→SA SEED→SA KEY → RC → RD → TD → RTE → ER）
	SIP-SECURITY-Pattern3-002-C	セキュリティアクセスループ①（DSC→SA SEED→SA KEY）
	SIP-SECURITY-Pattern3-003-C	セキュリティアクセスループ②（DSC→SA SEED→SA SEED）
生成乱数空間長 4bit（先頭から 124bit は 0）/ソフトウェアによる乱数生成 / 乱数生成器初期化無し	SIP-SECURITY-Pattern4-001-C	リプログラミング正常手順（DSC→SA SEED→SA KEY → RC → RD → TD → RTE → ER）
	SIP-SECURITY-Pattern4-002-C	セキュリティアクセスループ①（DSC→SA SEED→SA KEY）
	SIP-SECURITY-Pattern4-003-C	セキュリティアクセスループ②（DSC→SA SEED→SA SEED）
生成乱数空間長 128bit /マイコン内蔵のセキュリティハードウェア機能による乱数生成 / 乱数生成器初期化有り	SIP-SECURITY-Pattern5-001-C	リプログラミング正常手順（DSC→SA SEED→SA KEY → RC → RD → TD → RTE → ER）
	SIP-SECURITY-Pattern5-002-C	セキュリティアクセスループ①（DSC→SA SEED→SA KEY）
	SIP-SECURITY-Pattern5-003-C	セキュリティアクセスループ②（DSC→SA SEED→SA SEED）

【開発環境】

・ルネサス エレクトロニクス株式会社製 RH850/F1L premium マイコン
・ルネサス エレクトロニクス株式会社製 CS+（統合開発環境/R850 コンパイラ等含む）
・ルネサス エレクトロニクス株式会社製 FlashProgrammer（セキュリティハードウェア機能有効/無効設定）
・ルネサス エレクトロニクス株式会社製 CPU 評価ボード
・テセラ・テクノロジー株式会社製 CPU 評価ボード
・イーソル株式会社製 ELYZER （CAN アナライザ/リプログラミングシーケンス実行環境）
・Microsoft 社製 EXCEL 2010 （ELYZER 実行 model 生成用）
・Microsoft 社製 VisualStadio2012 （ELYZER SecurityDLL 作成用）

C 評価対象（車内システム）資料

・鍵配布ツール一覧

鍵配布手順評価向けに作成したツールについて、表 3.2b.4-4 に記す。

表 3.2b.4-4　鍵配布ツール一覧

ツール名（実行ファイル名）	概要
MasterKeyDistribution.exe	鍵配布用認証鍵配布ツール。評価向けには非開示とする。
KeyDistribuion.exe	評価向け鍵配布ツール。
KDF.exe	KDF アルゴリズムツール
KeyRemove.exe	鍵消去ツール。評価向けには非開示とする。

【開発環境】

・ルネサス エレクトロニクス株式会社製 RH850/F1L premium マイコン

・ルネサス エレクトロニクス株式会社製 CS+（統合開発環境/R850 コンパイラ等含む）

・ルネサス エレクトロニクス株式会社製 FlashProgrammer（セキュリティハードウェア機能有効/無効設定）

・ルネサス エレクトロニクス株式会社製 CPU 評価ボード

・テセラ・テクノロジー株式会社製 CPU 評価ボード

・Lawicel 製 CANUSB （CAN to USB Gateway/CAN 通信-USB 変換器）

・Microsoft 社製 VisualStadio2012 （鍵配布ツール作成用）

D 車載マイコン内蔵セキュリティハードウェア機能資料

平成28年度の評価対象マイコンに内蔵するセキュリティハードウェアは、SHE（Security Hardware Extension）規格に準拠している。これはEVITAプロジェクトで考案されたコンセプトを元にしたハードウェアセキュリティモジュールの規格であり、欧州OEMを中心に標準化活動が行われている。

この規格の特徴は、「演算途中のデータおよび格納したデータをソフトウェアに開示しない」ことである。ソフトウェアはSHE規格に定められた手順に従って、必要なデータを入力しコマンドを発行する。SHE準拠モジュールは入力データとコマンドに従って演算を行い、演算結果を出力する。ソフトウェアはこの演算結果を取得することで、発行したコマンドの結果を知ることができる。その結果、ソフトウェアはSHE準拠のセキュリティモジュールを適切に使用することによって保護された演算結果を取得し、外部からの各種セキュリティ攻撃に対して耐性を獲得することが可能となる。

SHEによって規定されている機能の一覧を以下に記す。

・真性（物理）乱数を元にした擬似乱数生成機能
・暗号化、復号化機能（AES128-CBC、AES128-ECB）
・MAC生成、照合機能（CMAC）
・セキュアBoot機能
・暗号化/復号化、MAC生成/照合、セキュアBootで使用するの鍵登録機能

3.2c　車外連携システム・車両レベルにおける評価技術の検討

3.2c.1　対策すべきポイントとその評価方法の見直し

(1) 検討の背景

平成 27 年度は自動走行システムアーキテクチャからモデル化し、システム実装から脅威分析を行った。以降、自動走行システムアーキテクチャについての変更は発生していないが、昨今のサイバーセキュリティの事例[1]などを鑑みると、システムの上に構築されるアプリケーションの視点からもセキュリティ脅威について検討し、システム実装から導出した対策の有効性について検証する必要がある。

サイバー攻撃のエントリポイントとしてアプリケーションの視点から最も脅威になるのは内部プログラムの書き換え機能であると考えられる。

車両を構成する Electric Control Unit（以下、ECU という）を対象としたプログラムの書き換えは「リプログラミング」と呼ばれ、機能改善やリコール時の対策手段を含む保守を目的とした機能として実装されている。

リプログラミングは機能的には車両に共通的な保守機能であり、ECU をリプログラミングが可能な状態にするまでの手順[2]と更新データの送信プロトコル[3]については標準化されているが、リプログラミングに関する全てが何らかの標準化された規約に基づいて行われる訳では無い。

標準化されている部分に関しては構造などが公開されている状態とも言えるので、よりサイバー攻撃の対象になりやすい。

図 3.2c.1-1 にこの手順と規格化の現状を図示する。

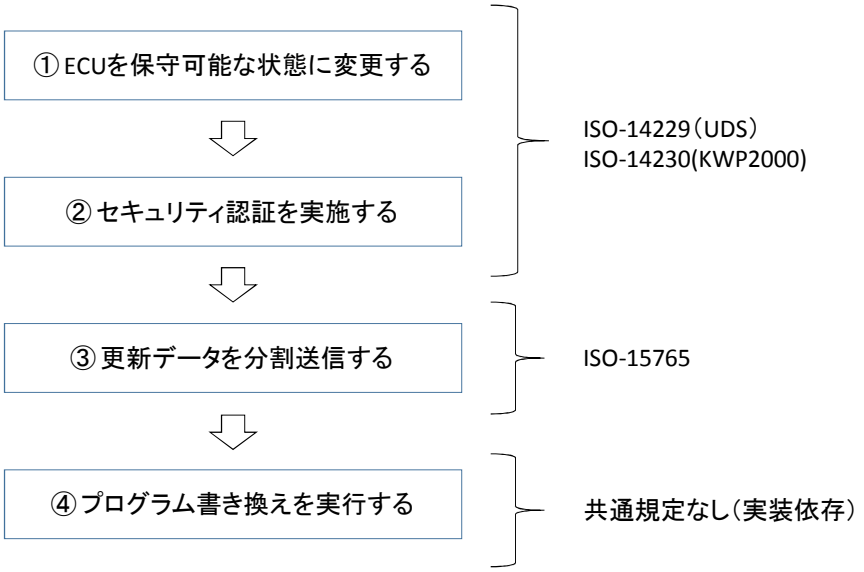


図 3.2c.1-1　リプログラミングの標準手順

従来の保守作業は認定された整備工場で行われるなど、リプログラミングに使われる装置やリプログラミングのデータについてはセキュリティ的に保護された環境・状態下にあるため、悪意のある作業者が関わらなければ改ざんが難しい環境にあった。その点ではセキュリティ対策を車両自身ではなく、整備工場の物理セキュリティなど車両外に求めることでリスク回避が可能であったと言える。

この保護された環境・状況は、構造的には外部から遮断された、いわゆる産業制御システムと同様ということができるが、昨今の Stuxnet[4]の事例からも判るように、サイバーセキュリティ的には決して安心できる状態であると言い切れない状態になってきている[5]。

また、近年の Information and Communication Technology（以下 ICT という）でのサイバー攻撃の動向を含めて考察を進めると、パソコンなどに侵入して内部のデータを暗号化することで使用できない状態とし、その解除キーを入手するために金品を要求する身代金型のマルウェア、いわゆるランサムウェアの流行に着目する必要がある[6])。

現時点で車両に関するランサムウェアの実例が報告されている訳ではないが、ICT とは異なり、車両の場合は「移動する」という目的があり、この目的をターゲット（人質）としたランサムウェアが想定される。この場合には、「走る・曲がる・止まる」の機能に対する攻撃ではなく、「動かなくする」あるいは「意図しない動きをする」といった攻撃が想定される。

一方で、電子化が進み高級車に搭載されたソフトウェアのコードは 1 億行を超えているといわれるまで車両が内蔵するプログラムサイズは増加している。

その保守コストの削減や利用者の利便性の観点で、パソコンやスマートフォンで行われているソフトウェア更新のように、ネットワークを通した配信を受けてのソフトウェア更新、いわゆるリモートリプログラミングが求められているのが世界的動向である（参考資料[7][8])。なお、このリモートリプログラミングのうち通信経路として無線通信を利用する場合は Over The Air （以下 OTA という）と呼ばれている。

参考資料をベースに、想定される車両に対するリモートリプログラミングシステムの全体イメージを図 3.2c.1-2 に示す。

この図からも判るように、外部境界は①と②が想定される。①の境界に関しては接続が限定されることから、専用線や仮想プライベートネットワーク（以下 VPN という）によるセキュリティ対策が期待できるが、②に関してはインターネット網が想定されるため一般の ICT 機器と同様のセキュリティ対策を検討する必要がある。

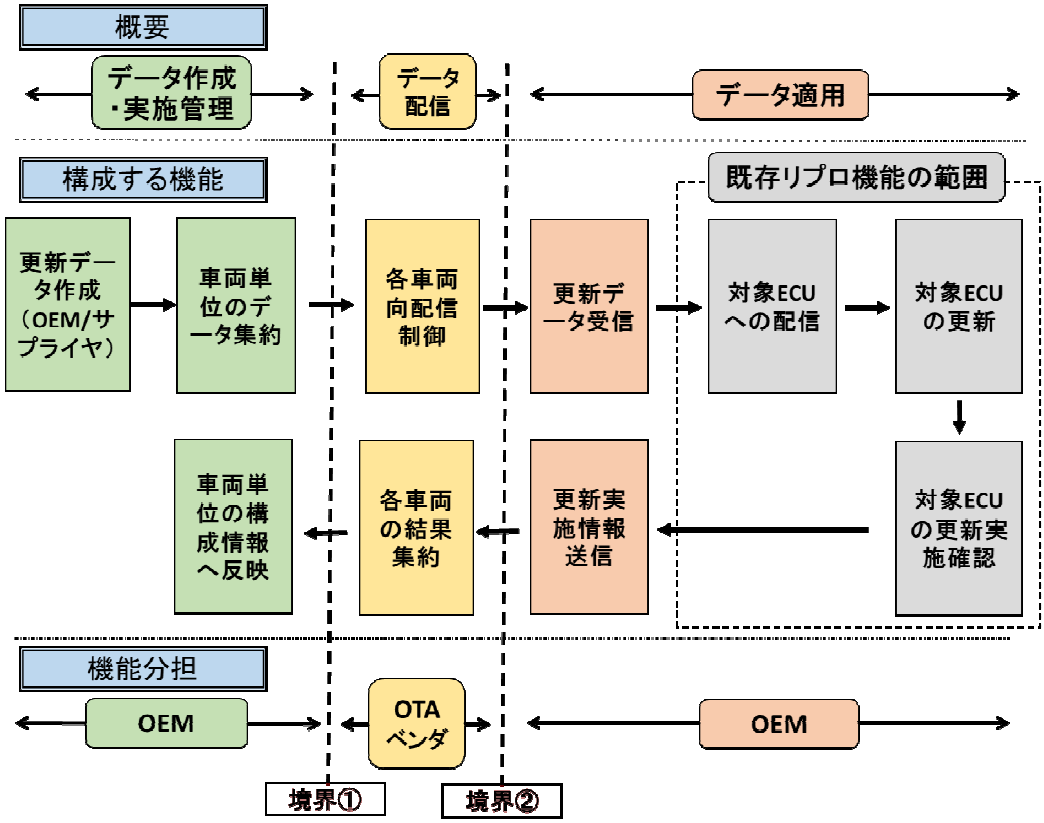


図 3.2c.1-2　リモートリプログラミングシステムの全体イメージ

これらの参考資料によれば、ソフトウェア起因のリコールは 15%にまで増加しており、例えば、フォードにおける 60 万台のリコールや GM におけるエンジン ECU の不具合による不完全燃焼の問題のリコールについてもソフトウェア起因であるとしている。

ソフトウェアのみがリコールの起因の場合には OTA を利用することで、整備工場などに持ち込まなくてもリモートリプログラミングにより改善が可能になる。結果として利用者の利便性（車両を整備工場に持ち込む手間や時間制約、車両が使えない時間の短縮など）も向上すると共に、OEM にとってもユーザの整備工場への持込を待たずにリプログラミングの実施が同時に・早期に実施できることから短期間のうちにリコールを完了することができるメリットがある。仮に整備工場に持ち込まれたとしても専用の更新用治具を使用する必要が無いため、治具待ちが原因による工場滞留時間も短縮される。

このような OTA サービスを提供する OTA ベンダ各社は、自社のみでのセキュリティ確保は難しいとの判断から、セキュリティ専門会社との提携、協業、統合などの実施によりセキュリティ対策の向上が進められており、特にクラウドセキュリティやダウンロードプロトコルについて ICT におけるサイバーセキュリティの知見を展開すべく、ICT ベンダとの協業が進められている。

また、セキュリティ対策として OTA の最終的な対象物である ECU に関して、ハードウェアとソフトウェアスタックのレイヤー毎に強力なセキュリティ対策を施す必要がある（いわゆる多層防御）ことが、参考資料においても指摘されている。

これは、表面的には ECU 自体での多層防御の必要性の指摘ではあるが、車両として考えた場合に個々の ECU 自体でレイヤー毎の対策はリソース的にもリスクとの費用効果の視点でも現実解では無いため、OTA システムのターゲットである車両全体への指摘と理解することが妥当である。

これらを踏まえて、次項では ICT におけるレイヤー毎の対策である多層防御の考え方の車両への適用について考察する。

(2) 侵入検知技術導入の必要性

例えば ICT での情報システムにおける多層防御の考え方を適用したとして、独立行政法人情報処理推進機構（以下、IPA）から発行されている「組織の重要情報の窃取を目的としたサイバー攻撃に関する注意喚起」[9]では、次のような対策が提示されている。

【対策 1】: 入口（ネットワーク経路）をしっかり守る
【対策 2】: ファイアウォールを抜けてもシステムにつけ入られる隙（脆弱性）を与えない
【対策 3】: ウイルスの活動（組織内蔓延（まんえん）や外部通信）を阻害、抑止する。
【対策 4】: 重要な情報はその利用を制限（アクセス制御）する
【対策 5】: 情報にアクセスされても保護するための鍵（暗号）をかける
【対策 6】: 操作や動き（ログ証跡）を監視・分析し不審な行為を早期に発見する
【対策 7】: 万一被害が発生したら早急な対応（ポリシーと体制）をとる

これは情報窃取されること、具体的にはシステムに侵害されて情報が持ち出されることを前提に対策を提案されたものである（図 3.2c.1-3 参照）。

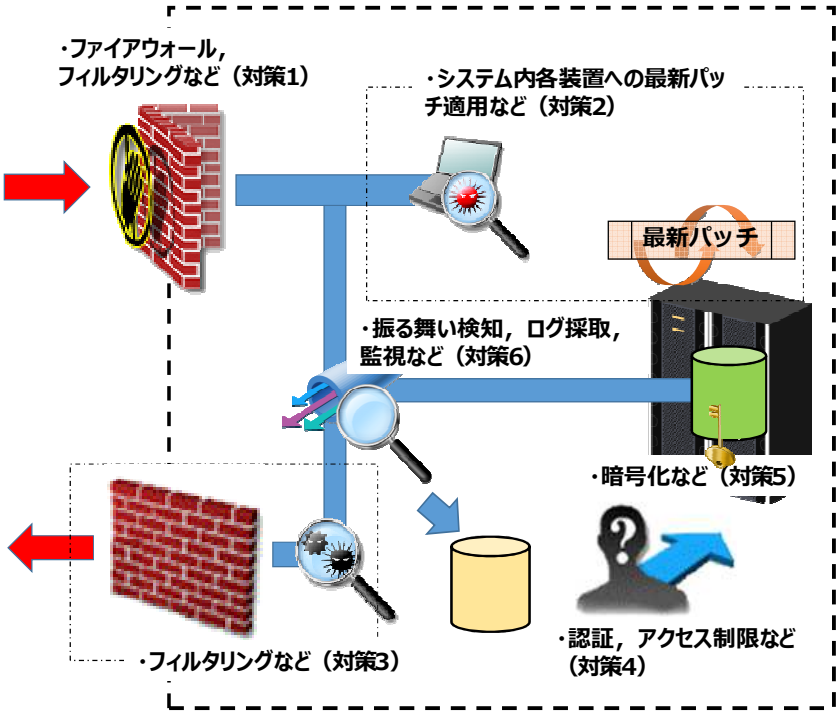


図 3.2c.1-3　ICT における多層防御を考慮したセキュリティ対策例

つまり想定しているサイバー攻撃の完了（目的の達成）は情報の窃取であり、システムへの侵入、改ざんは目的に至るまでの手段でしかない。

一方、現在想定している自動運転車両へのサイバー攻撃の標的は、自動車としての走行安全性に対する侵害としている。このため、ここに上げられている全部の対策では無く、システムの侵入および改ざんまでの対策を参考にすれば良いことになる。

提案の対策のうち、システム侵害および制御の乗っ取りなどを含むデータ破壊に関する対策は以下の四つが対象になる。

【対策 1】：入口（ネットワーク経路）をしっかり守る

【対策 2】：ファイアウォールを抜けてもシステムにつけ入られる隙（脆弱性）を与えない

【対策 4】：重要な情報はその利用を制限（アクセス制御）する

【対策 6】：操作や動き（ログ証跡）を監視・分析し不審な行為を早期に発見する

これらの対策を本プロジェクトで想定している自動走行システムアーキテクチャにあてはめると図 3.2c.1-4 のようになる。

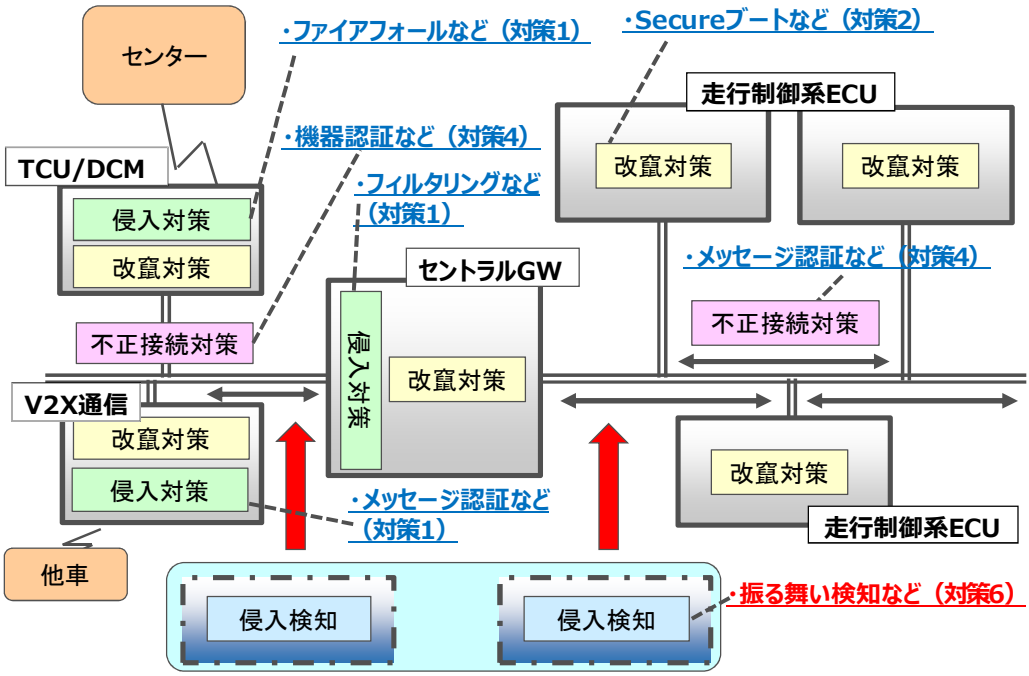


図 3.2c.1-4　自動走行システムアーキテクチャで考慮すべきセキュリティ対策

個々の対策に関しては、参考資料[10]などでも現状の対策技術の実装状況などが分析されている。また、この参考資料でも多層防御の観点で対策を分類しており、対策技術のアプローチとしては妥当であると考えられる。

従って、車両レベルでのセキュリティを考える場合には、各 ECU やシステムに対する改ざん対策や侵入対策などの直接的なセキュリティ対策技術の評価だけでなく、侵入検知技術のように、間接的なセキュリティ対策技術の評価も必要である。

ICT における侵入検知機能は Intrusion Detection System（以降 IDS という）と呼ばれている。この侵入検知機能に引き続いてネットワークの遮断などの防御機能も備えた Intrusion

Protect System（以降 IPS という）もあり、この二つを総称して IDS/IPS と呼ばれている。

この IDS/IPS は方式（設置場所）、方法（検知手段）、目的（用途）などの観点でいくつかに分類できる。

① 方式による分類

具体的には装置（機能）の設置場所による分類であり、代表的には以下の二つの方式がある。

(ⅰ) ネットワーク上で通信トラフィックを監視する（ネットワーク型）

(ⅱ)サーバなどのコンピュータ上で実行コマンドなどの動作ログやファイル操作を監視する（ホスト型）

② 方法による分類

具体的には検知手段（技術）による分類であり、代表的には以下の三つの方式がある。

(ⅰ)事前に用意したブラックリストにピックアップされている通信内容のパターンやルール（いわゆるシグニチャ）との一致を検知する（シグニチャ型）。

(ⅱ)攻撃の類型的動作要素を検知する（ヒューリスティック型）。これは①の派生・発展系とも言える。

(ⅲ)通常の操作・状態（ログインやコマンド実行時刻、ＣＰＵやネットワークなどのリソースの負荷状態など）との差を検知する（異常検知型）。

③ 検知後の動作による分類

具体的には上記のような手段で侵入・異常を検知した後にどのような動作をするかによる分類であり、代表的には以下の二つの方式がある。

(ⅰ)ネットワークシステムを遮断する（いわゆる IPS）。

(ⅱ)システム管理者に通知のみを送信する（いわゆる IDS）。

現実のシステムでは、守るべきデータやシステム運用を総合的に考慮して、リスクヘッジと投資コストのバランスから上記のどのタイプの IPS/IDS システムを導入するかが決定されている[11]。

②の方法による分類からも判るように、例えばシグニチャ型では新しい攻撃方法などシグニチャに無いパターンでの攻撃を見逃す、いわゆる“検知漏れ”が発生する可能性がある。また異常検知型では突発的に発生するイレギュラな運用を攻撃があったと検知する、いわゆる“誤検知”が発生する可能性がある。

これらの検知率が IPS/IDS システムの一つの評価指標になり得るが、現時点では共通的な評価指標は存在していない。

なお、本事業における自動走行システムアーキテクチャでは考慮していないが、車両内部の情報記録・管理も重要な課題である。今後、コネクテッドカーが進化し、いろいろな利便性の向上を図るため、クレジットカード番号などの決済手段に係わる情報や、所有者、

運行者、乗員などの個人情報またはそれに紐付く情報を車両側に保持することが想定される。

さらに、事故が発生した場合に、事故原因・責任所在の特定のために、自動運転の制御を行ったログデータなど、車両の挙動を特定するためのデータが裁判所などの第三者から求められる可能性[12][13]がある。現時点ではこのようなデータの保持が義務づけられている訳では無いが、今後の社会的要求などから、このようなデータを車両側が一時的にでも保持する必要があれば、前述の利便性向上時と同様に情報の保持および持ち出しの視点でのセキュリティ対策までを検討する必要が生じる。

現時点の自動走行システムアーキテクチャでは、情報の機密性に関する対処は十分に織り込まれているとは言えないが、今後は重要度が増していくと考えられる。

(3) CAN における侵入検知

近年、車外情報と連携して車を制御する機能を車に搭載する検討が進められている。

例えば、クラウドから 3D マップを取り込み効率のよい経路を選択したり、他車・歩行者・信号等のインフラと通信し安全な走行を実現したり、スマートフォンのような持ち込み機器と連携し快適な状態にパーソナライズを行うことが検討されている。これらの実現のためには、車外情報を取り込み解析して装置を制御するシステムが必要である。

このシステムは得られた情報に従い制御されるため、情報に誤りがあると正しく動作せず、自動運転車の場合には、人命にかかわる影響が生じる可能性がある。また、車外情報はネットワーク・無線通信を経由し取得するため、ICT と同等なサイバー攻撃にさらされる可能性が高い。そこで主要構成要素からなる単純なシステム（図 3.2c.1-5）を定義した後、どのような脅威が存在するのかを検討し、対策を施すことを考える。

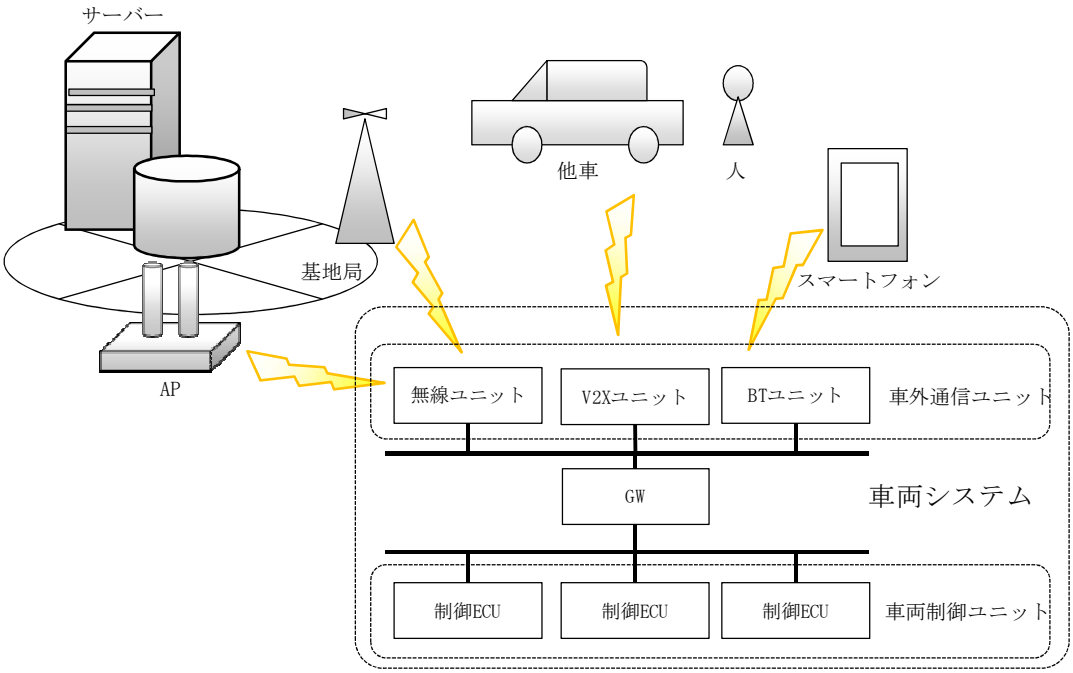


図 3.2c.1-5　車外連携システム

上記のような車外連携システムにおいて、車両は、車外のサーバと LTE 等のキャリアインフラ、Wi-Fi などを経由してつながり、周囲の車や交通インフラや人とは V2X 通信でつながり、入手した情報を基に車両の制御を行うと考える。これら通信を経由して入手する情報は車外通信ユニットから車載ネットワークを経由して制御 ECU に伝えられる。このようなシステムにおいて、自動運転車は車外からの情報を基に走る、曲がる、止まる等の重要な動作を決定するため、制御 ECU が受け取る情報を操作し正規とは異なる情報に変換して車の動作を制御することが可能である。攻撃者は制御 ECU が受け取る情報に注目し、誤った情報を送りつけるような攻撃を行うことが考えられ、制御 ECU が接続されている車載ネットワークに攻撃データを注入させると考えるのが妥当である。

具体的には、図 3.2c.1-5 のように車両内 ECU はネットワークでつながれ、メッセージのやり取りを行って機能を実現している。このようなシステムに外部から攻撃する場合、外部から車載ネットワークに攻撃メッセージ(偽メッセージ)を注入することが必要である。それを実現するために、まず車外通信ユニットを攻撃しここを拠点にして、制御 ECU に対して攻撃メッセージを注入すると考えられる。

そこで、外部から攻撃メッセージを注入する脅威に対し、機器認証技術が有効な対策になるかを検討するため、ISO9798[14]で規定された認証方式を調査した。その結果、ISO9798 の機器認証は機器が正しいことを他の機器が認証するプロトコルであり、攻撃メッセージを注入する機器を車両ネットワークに追加する攻撃に対して有効な手段とはならないことがわかった。このような攻撃に対抗するには、車外から侵入した攻撃メッセージを検知して攻撃のレベルに応じシステムを安全な状態に遷移させることが必要になると考えて、攻撃メッセージが注入されたことを検知する侵入検知技術を検討することにした。

車の制御 ECU が接続されている車載ネットワークは CAN 等の単純な通信手段が用いられているケースが多い。本研究はこの CAN を対象に侵入検知の評価基準を検討する。

検知技術を検討する前に、ECU 間通信方法について整理する。CAN は、メッセージ ID の取りうる範囲が狭い（通常の CAN は 256 個)、1 パケットで転送できるデータ量が少ない（8byte)、フロー制御機能がなく宛先も送信元も設定できない（ブロードキャスト型）等制限があるプロトコルである。このため通常の通信で行われるリクエスト／レスポンス型のメッセージ通信を行うことは下記に示す理由で難しいと判断した。

(ⅰ) メッセージ ID を消費する

宛先が指定できないため宛先ごとに ID を分ける必要がある。さらに送信元も区別できないため、リクエスト／レスポンス型の通信を行うには 1 シーケンスあたり二つの異なる ID を消費する。

ID の設定可能個数が少ないため、全ての ECU 通信をリクエスト／レスポンスのメッセージ通信にすることは難しい。

(ⅱ)リクエストの繰り返し送信が必要

リクエストが相手に届いたことを確認できないので、レスポンスが来るまでリクエストを再送し続ける必要がある。このため、ネットワークを混雑させメッセージ遅延を誘発する可能性がある。ネットワーク負荷を考慮した場合、リクエストせずにレスポンスを受け

取れるのが望ましい。

さらに、車を構成する ECU は異なるメーカで構成されていることが多い。このため、メーカ間でリクエスト／レスポンスのメッセージの I/F を合わせる手段が必要になる。また、リクエスト／レスポンスのシーケンスの互換性検証も必要になる。ゆえに異なるメーカ ECU 間でリクエスト／レスポンス型のメッセージ通信の採用は難しいと考えられる。

リクエスト／レスポンス型を使わずに各 ECU が情報をやり取りするには、各 ECU が自状態を通知し、各 ECU が自制御に必要な情報をこの通知されている情報から取得する方法が考えられる（図 3.2c.1-6 参照）。この方法は、メーカ間でのすり合わせが必要なく、メッセージ ID 消費も抑えられ、車載ネットワーク負荷の増加も抑制できると考えられる。ゆえに、ECU 間の通信にこの方法が採用されていると考えることは妥当である。

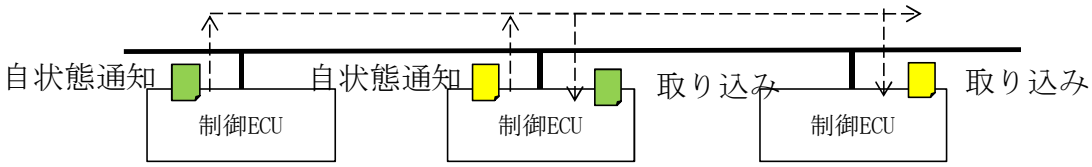


図 3.2c.1-6　制御 ECU 間メッセージの例

この方式の場合、状態通知を定期的に行なわないと必要な情報が必要とする ECU に伝わらない。また、実際の車では多くの ECU がこの方式で通信していると考えられるが、議論を行いやすくするため一つのメッセージ ID に着目し検討を進める。本研究では車両システムへの攻撃は車外から攻撃メッセージを注入することと定義した。前述したようなシステムにおいて、車載ネットワークに攻撃メッセージが注入されると、各 ECU が定期的に送信しているメッセージの周期が乱れる（図 3.2c.1-7 参照）。メッセージ周期の乱れを検知した場合、車外から攻撃メッセージが注入されたと判断するのが妥当である。なお、故障した ECU がある場合にも同様にメッセージの周期が乱れるが、本研究では ECU の故障は研究の範囲外とし、対象のシステムには存在しないと仮定し検討を行った。

具体的には、監視対象のメッセージを選定し、設計情報に基づいてネットワーク上に流れるメッセージの周期を監視する。設計周期以外のタイミングで監視対象メッセージを受信した場合、攻撃を受けたと判断する。なお、実際の車両では、ECU の性能や CAN プロトコルのメッセージ衝突回避機能などにより、攻撃を受けていなくとも周期の乱れは発生すると想定される。実際の製品はこれらを考慮しなければならないが、本研究は周期の乱れは一定の範囲内（基準周期±XX％：XX＜100）にとどまると仮定した。

なお、今回は攻撃検知を目的とした研究であるため、装置の故障による周期乱れは発生しない前提としたが、実際には装置故障は想定しなければならない。最終的な攻撃と装置故障の区別については他の手段も用いて総合的に判断する方法を用いる必要がある。

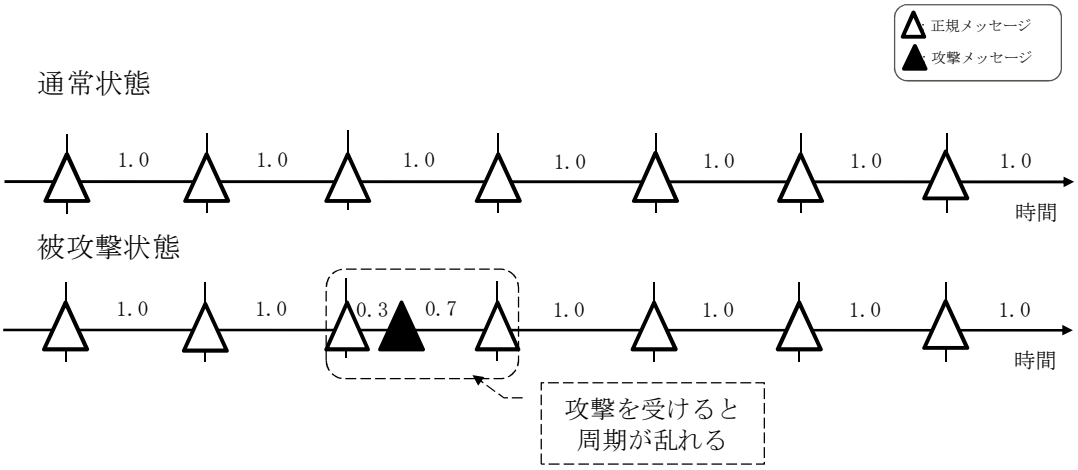


図 3.2c.1-7　攻撃による周期乱れの例

周期監視型検知以外の方法に、メッセージに装置固有のシグネチャを付与し、ネットワーク内のメッセージのシグネチャを監視するものがある。この方法は個車・装置ごとに固有なシグネチャを検知する必要があり、シグネチャの判別のしやすさ等に検知結果が左右されるため評価指標の一般化は難しい。そこで今回は、車載ネットワークを監視する監視型としてメッセージの周期監視型検知を対象にした評価指標を作成することとした。

(4) 海外の動向（参考）

Cybersecurity Best Practice for Modern Vehicles[15]において、NHTSA が自動車のサイバーセキュリティを向上させることを目的に実施するべき推薦事項の記載を行っている。

ただ文献内で実際に推薦しているセキュリティ機能への階層的なアプローチについての実施手順も、Frameworks of Improving Critical Infrastructure Cybersecurity[16]という、重要基盤に対してのセキュリティの機能改善を目的とした（対象物を車載に限定しない）、一般的なフレームワークを参照している。

Frameworks of Improving Critical Infrastructure Cybersecurity においては、framework の手順の中で侵入検知も識別する項目の中に含まれている。本文献中で特に侵入検知と関わりが深い Detection のカテゴリ内の項目を表 3.2c.1-1～3 に示す（参考文献[17]）.

表 3.2c.1-1　Frameworks of Improving Critical Cybersecurity にて記載されている侵入検知に関わるコアフレームワーク（その 1）

Category	Subcategory	Informative Reference
Anomalies and Events (DE.AE): Anomalous activity is detected in a timely manner and the potential impact of events is understood.	**DE.AE-1:** A baseline of network operations and expected data flows for users and systems is established and managed	**COBIT 5** DSS03.01 **ISA 62443-2-1:2009** 4.4.3.3 **NIST SP 800-53 Rev. 4** AC-4, CA-3, CM-2, SI-4
	DE.AE-2: Detected events are analyzed to understand attack targets and methods	**ISA 62443-2-1:2009** 4.3.4.5.6, 4.3.4.5.7, 4.3.4.5.8 **ISA 62443-3-3:2013** SR 2.8, SR 2.9, SR 2.10, SR 2.11, SR 2.12, SR 3.9, SR 6.1, SR 6.2 **ISO/IEC 27001:2013** A.16.1.1, A.16.1.4 **NIST SP 800-53 Rev. 4** AU-6, CA-7, IR-4, SI-4
	DE.AE-3: Event data are aggregated and correlated from multiple sources and sensors	**ISA 62443-3-3:2013** SR 6.1 **NIST SP 800-53 Rev. 4** AU-6, CA-7, IR-4, IR-5, IR-8, SI-4
	DE.AE-4: Impact of events is determined	**COBIT 5** APO12.06 **NIST SP 800-53 Rev. 4** CP-2, IR-4, RA-3, SI-4
	DE.AE-5: Incident alert thresholds are established	**COBIT 5** APO12.06 **ISA 62443-2-1:2009** 4.2.3.10 **NIST SP 800-53 Rev. 4** IR-4, IR-5, IR-8

次頁へ→

表 3.2c.1-2　Frameworks of Improving Critical Cybersecurity にて記載されている
侵入検知に関わるコアフレームワーク（その 2）

続き→

Security Continuous Monitoring (DE.CM): The information system and assets are monitored at discrete intervals to identify cybersecurity events and verify the effectiveness of protective measures.	**DE.CM-1:** The network is monitored to detect potential cybersecurity events	**CCS CSC** 14, 16 **COBIT 5** DSS05.07 **ISA 62443-3-3:2013** SR 6.2 **NIST SP 800-53 Rev. 4** AC-2, AU-12, CA-7, CM-3, SC-5, SC-7, SI-4
	DE.CM-2: The physical environment is monitored to detect potential cybersecurity events	**ISA 62443-2-1:2009** 4.3.3.3.8 **NIST SP 800-53 Rev. 4** CA-7, PE-3, PE-6, PE-20
	DE.CM-3: Personnel activity is monitored to detect potential cybersecurity events	**ISA 62443-3-3:2013** SR 6.2 **ISO/IEC 27001:2013** A.12.4.1 **NIST SP 800-53 Rev. 4** AC-2, AU-12, AU-13, CA-7, CM-10, CM-11
	DE.CM-4: Malicious code is detected	**CCS CSC** 5 **COBIT 5** DSS05.01 **ISA 62443-2-1:2009** 4.3.4.3.8 **ISA 62443-3-3:2013** SR 3.2 **ISO/IEC 27001:2013** A.12.2.1 **NIST SP 800-53 Rev. 4** SI-3
	DE.CM-5: Unauthorized mobile code is detected	**ISA 62443-3-3:2013** SR 2.4 **ISO/IEC 27001:2013** A.12.5.1 **NIST SP 800-53 Rev. 4** SC-18, SI-4, SC-44
	DE.CM-6: External service provider activity is monitored to detect potential cybersecurity events	**COBIT 5** APO07.06 **ISO/IEC 27001:2013** A.14.2.7, A.15.2.1 **NIST SP 800-53 Rev. 4** CA-7, PS-7, SA-4, SA-9, SI-4
	DE.CM-7: Monitoring for unauthorized personnel, connections, devices, and	**NIST SP 800-53 Rev. 4** AU-12, CA-7, CM-3, CM-8, PE-3, PE-6, PE-20, SI-4
	DE.CM-8: Vulnerability scans are performed	**COBIT 5** BAI03.10 **ISA 62443-2-1:2009** 4.2.3.1, 4.2.3.7 **ISO/IEC 27001:2013** A.12.6.1 **NIST SP 800-53 Rev. 4** RA-5

次頁へ→

表 3.2c.1-3　Frameworks of Improving Critical Cybersecurity にて記載されている
侵入検知に関わるコアフレームワーク（その 3）

続き→

Detection Processes (DE.DP): Detection processes and procedures are maintained and tested to ensure timely and adequate awareness of anomalous events.	**DE.DP-1:** Roles and responsibilities for detection are well defined to ensure accountability	**CCS CSC** 5 **COBIT 5** DSS05.01 **ISA 62443-2-1:2009** 4.4.3.1 **ISO/IEC 27001:2013** A.6.1.1 **NIST SP 800-53 Rev. 4** CA-2, CA-7, PM-14
	DE.DP-2: Detection activities comply with all applicable requirements	**ISA 62443-2-1:2009** 4.4.3.2 **ISO/IEC 27001:2013** A.18.1.4 **NIST SP 800-53 Rev. 4** CA-2, CA-7, PM-14, SI-4
	DE.DP-3: Detection processes are tested	**COBIT 5** APO13.02 **ISA 62443-2-1:2009** 4.4.3.2 **ISA 62443-3-3:2013** SR 3.3 **ISO/IEC 27001:2013** A.14.2.8 **NIST SP 800-53 Rev. 4** CA-2, CA-7, PE-3, PM-14, SI-3, SI-4
	DE.DP-4: Event detection information is communicated to appropriate parties	**COBIT 5** APO12.06 **ISA 62443-2-1:2009** 4.3.4.5.9 **ISA 62443-3-3:2013** SR 6.1 **ISO/IEC 27001:2013** A.16.1.2 **NIST SP 800-53 Rev. 4** AU-6, CA-2, CA-7, RA-5, SI-4
	DE.DP-5: Detection processes are continuously improved	**COBIT 5** APO11.06, **ISA 62443-2-1:2009** 4.4.3.4 **ISO/IEC 27001:2013** A.16.1.6 **NIST SP 800-53 Rev. 4**, CA-2, CA-7, PL-2, RA-5, SI-4, PM-

3.2c.2　対策技術の評価方法・基準についての評価環境の構築

侵入検知対策技術に対して、検知精度を評価するための評価指標の定義を以下に行う。

なお、本報告書で対象としている侵入検知技術は、重要なメッセージは周期的に送信されていることに鑑みて、CAN Bus 上を流れる周期性を持つメッセージに対して、メッセージ周期の乱れが発生していることを検知する技術とする。なお、一定時間ごとに区切った範囲において正しく攻撃と判断できれば、たとえ個々のメッセージに対する判断が誤っていたとしても検知成功と定義している。

侵入検知の評価尺度については表 3.2c.2-1 の 4 つの状態パラメータを用いて検知精度を導出する。これらのパラメータは、ICT 分野の異常検知でも一般的に用いられている。

表 3.2c.2-1　侵入検知時の現状状態と検知判断による状態パラメータ

		検知判断	
		異常ありと判断	異常なしと判断
現実状態	異常あり (Condition Positive)	True Positive	False Negative
	異常なし (Condition Negative)	False Positive	True Negative

・True Positive Rate(TPR)：検知率
現実に攻撃があった状態（回数）のうち、攻撃があると判断した状態（回数）の割合である。

$$\mathrm{TPR} = \frac{\sum(True\ Positive)}{\sum(Condition\ Positive)}$$

・True Negative Rate(TNR)：正常メッセージ認識率
現実に攻撃がなかった状態（通常メッセージを送信した回数）のうち、攻撃がなかったと判断した状態（回数）の割合である。

$$\mathrm{TNR} = \frac{\sum(True\ Negative)}{\sum(Condition\ Nagative)}$$

・False Positive Rate(FPR)：誤検知率
現実に攻撃がなかった状態（通常メッセージを送信した回数）のうち、攻撃があると誤判断した状態（回数）の割合である。

$$\mathrm{TNR} = \frac{\sum(False\ Positive)}{\sum(Condition\ Negative)}$$

・False Negative Rate(FNR)：見逃し率
現実に攻撃があった状態（回数）のうち、攻撃がなかったと誤判断した状態（回数）の割合である。

$$\mathrm{TNR} = \frac{\sum(False\ Negative)}{\sum(Condition\ Positive)}$$

しかし、これらの指標をもって精度を評価する場合、どのような単位のうち、どのような状態になった場合を検知しているかという定義を行う必要がある。特に検知率（TPR）や見逃し率（FNR）の精度を評価する際に、攻撃の種類によっては指標値にばらつきが発生するため明確に設定する必要がある。現状このような車載ネットワークに関する明確な定義はない。そのため、本報告書では複数の定義を行い、それぞれの検知精度について考察を行う。

① 個々の正規メッセージ（攻撃メッセージ）に対して一つの侵入の検知が可能（メッセージ単位での評価手法）

図 3.2c.2-1 にメッセージ単位の車載ネットワークに対する攻撃パターンと検知結果の一例を示す。

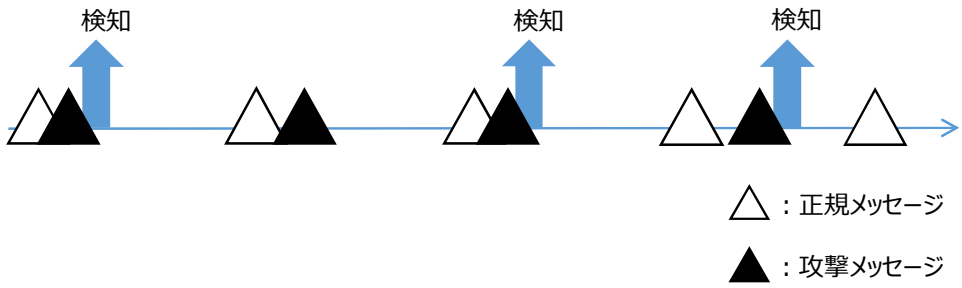


図 3.2c.2-1　車載ネットワークに対する攻撃パターンと検知結果の一例（メッセージ単位）

図 3.2c.2-1 のようなメッセージの流れと、侵入の検知結果があったと仮定した場合、この評価指標を用いると以下のような結果となる。

Condition Positive = 4
Condition Negative = 1
True Positive = 3
True Negative = 1
False Positive = 0
False Negative = 1

従ってそれぞれの検知精度は、以下のように求められる。

TPR（検知率）　= 3/4 = 75%
TNR（正常メッセージ認識率）　= 1/1 = 100%
FPR（誤検知率）　= 1-TNR = 0%
FNR（見逃し率）　= 1-TPR = 25%

② 単位時間のうちに複数あるいは一つの攻撃が送信された場合に一つ以上の侵入の検知が可能（単位時間での評価手法）

愚直に攻撃を検知する場合、上記①で述べた指標で評価をするほうが良いが、DoS 攻撃などの可用性阻害を狙う攻撃に対しては、必ずしも一つ一つの攻撃メッセージを検知する必要はないと考える。そのため。第 2 の定義として、測定全体を単位時間による区間に区切り、その区間中において検知ができていれば正しく検知できたと判断する尺度を用いる。

なお、設定するべき単位時間は攻撃の種類や、侵入検知を適応するシステムに応じて異なる。今回の実験においては、対象の周期メッセージの周期に合わせて、単位時間を設定した。

図 3.2c.2-2 に単位時間あたりの車載ネットワークに対する攻撃パターンと検知結果の一例を示す。

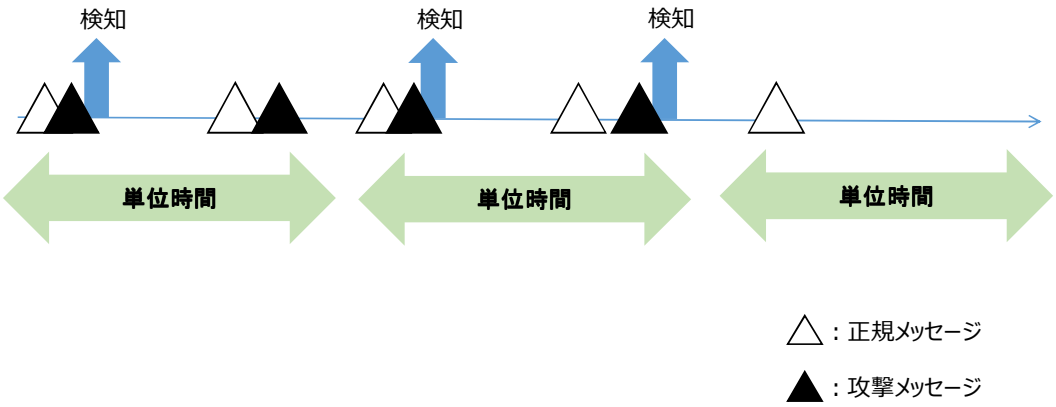


図 3.2c.2-2　車載ネットワークに対する攻撃パターンと
検知結果の一例（単位時間）

図 3.2c.2-2 のようなメッセージの流れと、侵入の検知結果があったと仮定した場合、単位時間あたりに攻撃が発生しているか、あるいは侵入を検知しているかに基づいて検知指標を求める。

Condition Positive = 2
Condition Negative = 1
True Positive = 2
True Negative = 1
False Positive = 0
False Negative = 0

この結果、それぞれの検知精度は、以下のようになる。

TPR（検知率）　= 2/2 = 100%
TNR（正常メッセージ認識率）　= 1/1 = 100%
FPR（誤検知率）　= 1-TNR = 0%
FNR（見逃し率）　= 1-TPR = 0%

(1) 実験環境

車両における侵入検知技術の評価指針・基準を設定し、妥当性を検証する為、自動運転の簡易モデル車両である ZMP 社製 RoboCar MV2[18]を改造して、共通モデルに合わせた評価環境（実験システム）を構築した。

図 3.2c.2-3 に評価環境（実験システム）外観図を示し、以下に構築した実験装置の内容を記載する。

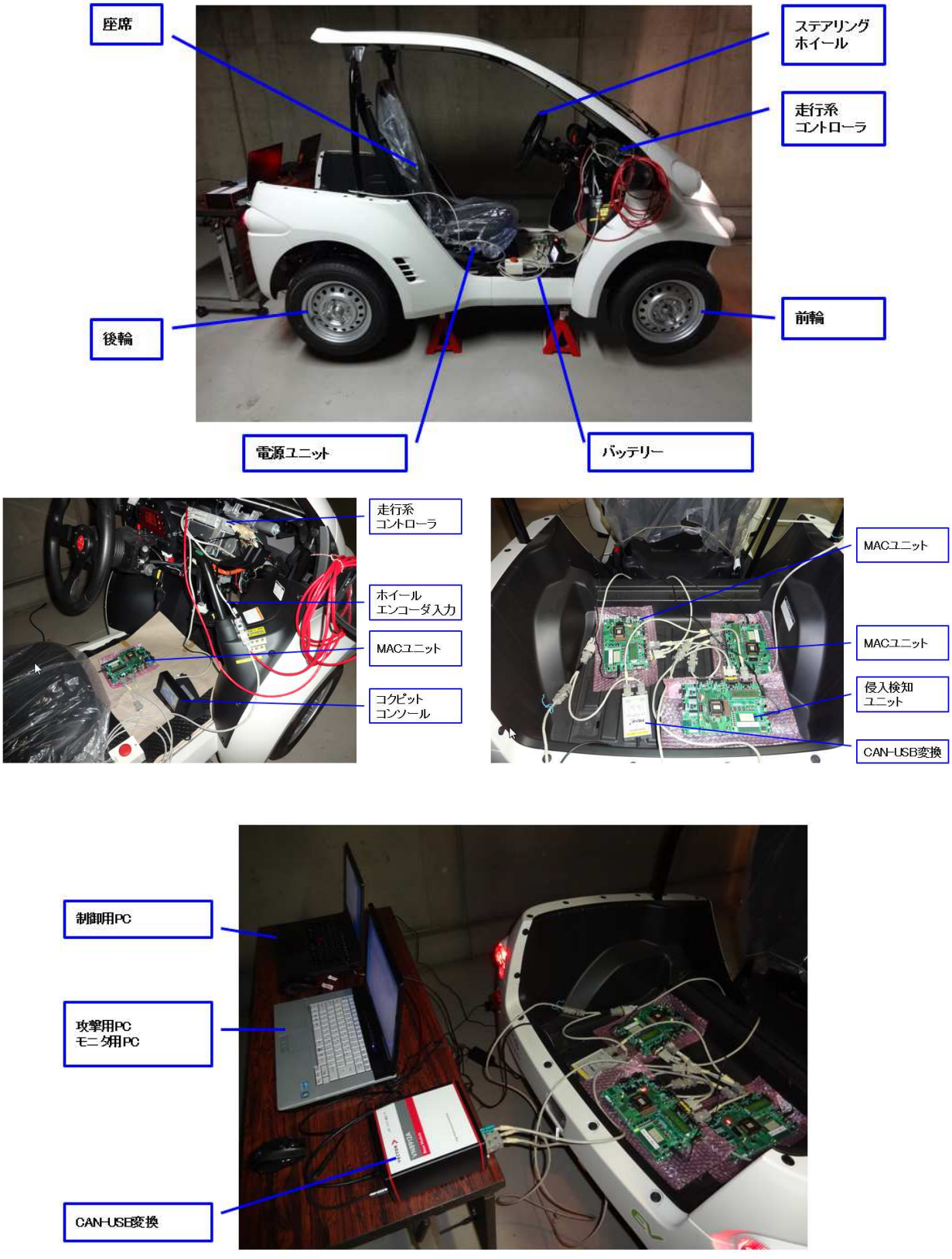


図 3.2c.2-3　評価環境（実験システム）外観図

図 3.2c.2-4 に評価環境（実験システム）のシステム構成を示す。

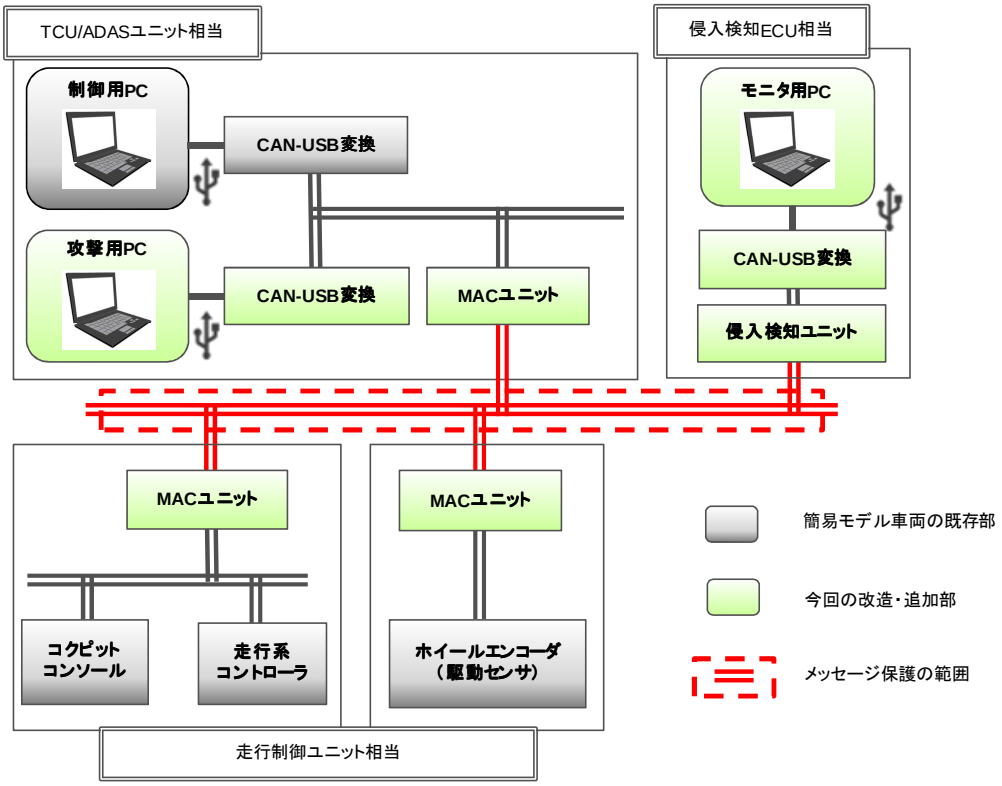


図 3.2c.2-4　評価環境のシステム構成

ベースとなる自動運転の簡易モデル車両にはセキュリティ対策が搭載されていないため、保護すべき入出力ポイントにメッセージ認証コード（Message Authentication Code）検証ユニット（以下、「MAC ユニット」という）を追加した。

MAC ユニット、侵入検知ユニットのハードウェアはルネサス社のマイコン評価 KIT FL-850/F1L-176 を使用し、それぞれにメッセージ認証コード検証機能、侵入検知機能のソフトウェアを実装している。ルネサス社のマイコン評価 KIT FL-850/F1L-176 の仕様を表 3.2c2-2 に示し、搭載マイコンであるルネサス社の RH850/F1L の仕様を表 3.2c2-3 に示す。

表 3.2c2-2　マイコン評価 KIT FL-850/F1L-176 の仕様

製品名	FL-850/F1L-176-S
インタフェース	CPU ボード ・Nexus デバッグ・コネクタ（14pin/1pin） ベースボード ·CAN 6ch ·RS-232C ·USB（バーチャル COM ポート）
その他周辺機能	CPU ボード ·A/D 端子に接続できるフィルタ回路 ·電流測定用ジャンパピン ベースボード ·UART 通信による搭載 LCD への文字表示 ·チャタリング防止回路付きプッシュ・スイッチ ·可変抵抗によるマイコン A/D 端子へのアナログ電圧印可
外形寸法	CPU ボード：77 x 133 (mm) ベースボード：210 x 133 (mm)
付属品	AC アダプタ(Input:100～240V, Output:+5V:)

表 3.2c2-3　RH850/F1L の仕様

製品名	RH850/F1L(176pin LQFP Package)
CPU 周波数	80MHz(Max)
Memory	2048KB Program Flash 128KB RAM 64KB Data Flash

実験では、MAC ユニットにより保護された CAN Bus 上に、侵入検知ユニットを接続して攻撃の監視を行った。

実験を進める中で、外部からエラーフレーム送信の攻撃を受けた場合、MAC ユニットが一種のフィルタのように動作してエラーフレームが除外され、保護された CAN Bus 上にエラーフレームが侵入しない状況が発生した。このため、エラーフレーム送信の攻撃時は MAC ユニットを経由せず CAN Bus に接続する図 3.2c.2-5 の構成とした。

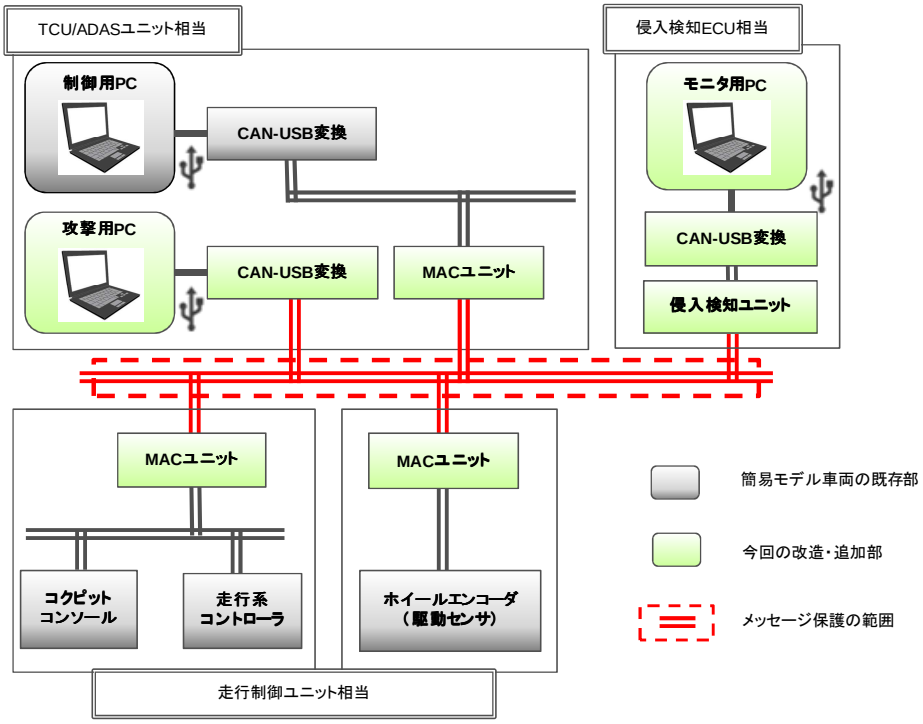


図 3.2c.2-5　評価環境のシステム構成（エラーフレーム送信攻撃時）

(2) 実験内容および結果

① テスト方法

図 3.2c.2-6、図 3.2c.2-7 に評価環境（実験システム）におけるデータの流れを示す。

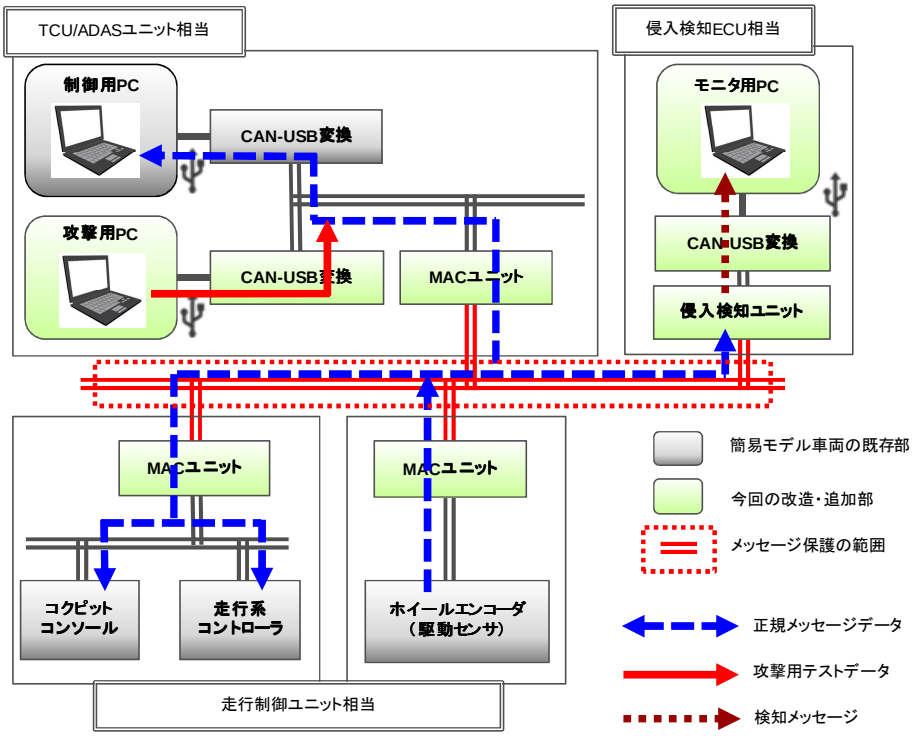


図 3.2c.2-6　評価環境（実験システム）のデータの流れ

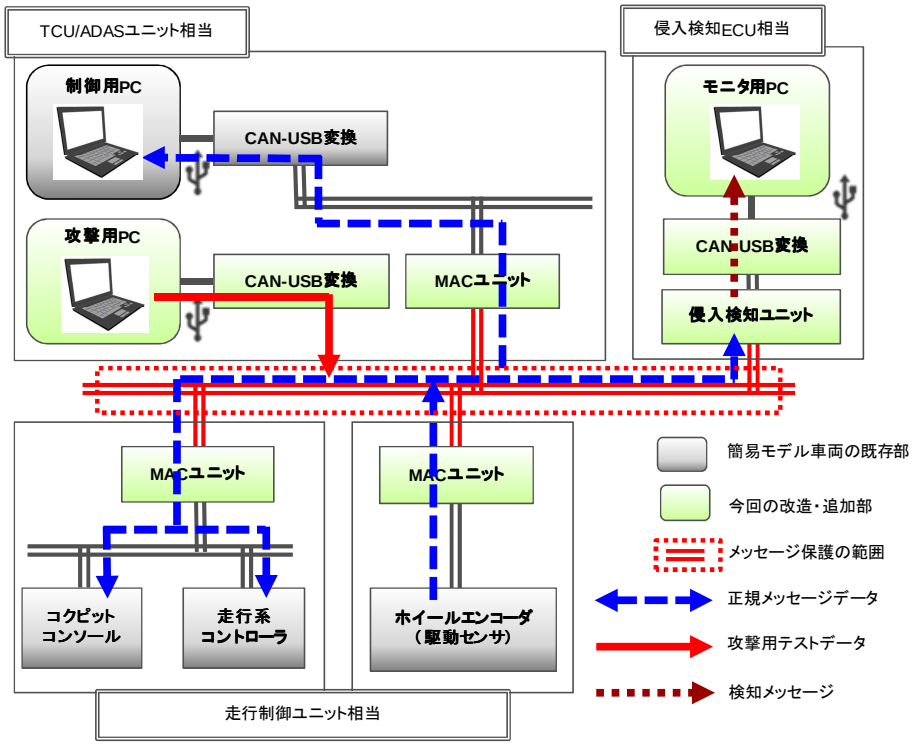


図 3.2c.2-7　評価環境（実験システム）のデータの流れ
（エラーフレーム送信攻撃時）

攻撃用 PC に車両ネットワーク開発・テストツールである Vector 社製 CANoe V9.0 をインストールし、PC と CAN bus の接続には CAN-USB 変換を行うプロトコル変換機として Vector 社製 VN8912A を使用した。

モニタ用 PC には同じく Vector 社製 CANoe V8.5 をインストールし、PC と CAN bus の接続も同じく Vector 社製 VN8912A を用いた。なお、CANoe のバージョンの違いは入手時期の差によるものである。

攻撃用のデータは CANoe のオートメーションシーケンス機能を利用して攻撃用 PC に用意した。

自動運転時（図 3.2c.2-6 および図 3.2c.2-7 で正規メッセージデータが流れている状態）に、外部からの侵入により TCU/ADAS ユニット相当の制御用 PC が乗っ取られるケースを仮定し、攻撃用 PC の CANoe からテストデータを流して攻撃した。

メッセージ保護をすり抜けて CAN Bus に侵入してきた攻撃を侵入検知ユニットで検知し、検知メッセージをモニタ用 PC の CANoe に送信することで検知結果を確認する。同様に、保護された CAN Bus 上にエラーフレーム攻撃が侵入してきた前提で、モニタ用 PC で検知結果を確認する。

CAN Bus 上のメッセージが正常メッセージか攻撃メッセージかの区別については、ログデータに記録されているメッセージの内容を目視で解析することにより行った。

② テストデータ作成

IPA から発行されている「情報セキュリティ 10 大脅威 2016」[19]、NISC から発行されている「情報システムに係る政府調達におけるセキュリティ要件策定（SBD）マニュアル」-付録 A. 対策要件集[20]、および ISO/IEC15408[21]を参考に、今回の評価環境（実験システム）に対する攻撃のテストデータについて検討した。

最初に、最新トレンドの ICT における情報セキュリティ脅威およびその攻撃手順を整理し、直接的な攻撃手順を車両観点でマッピングしてテストデータの方向性を検討した。表 3.2c2-4 に ICT における情報セキュリティ脅威を示す。

表 3.2c2-4　ICT における情報セキュリティ脅威

攻撃事例	侵入経路	No	攻撃手順	分類	攻撃データ
標的型攻撃による情報流出	インターネット(メール)	1	電子メールの添付ファイルを開封してマルウェアに感染	侵入	－
		2	当該端末を外部から操作して機密情報を外部へ送信	攻撃	偽のCANメッセージを流すことにより、制御系機能(アクセル、ブレーキ、ステアリング、シフト)を操作する
内部不正による情報漏えい	インターネット/ 記憶媒体	1	社員が正規の手順を使って機密情報にアクセスし、インターネットまたは記憶媒体を使って機密情報を持ち出し	侵入 攻撃	マッピング不可
ウェブサービスからの個人情報の窃取	インターネット	1	脆弱性スキャナー(Acunetix,Nikto)等でアプリケーション上の脆弱性を高速に検索し、脆弱性を利用して侵入	侵入	－
		2	SQLインジェクション等の手法を用いてデータベースシステムを不正に操作し機密情報を取得して外部へ送信	攻撃	偽のCANメッセージを流すことにより、制御系機能(アクセル、ブレーキ、ステアリング、シフト)を操作する
サービス妨害攻撃によるサービス停止	インターネット	1	脆弱性スキャナー(Acunetix,Nikto)等でアプリケーション上の脆弱性を高速に検索し、脆弱性を利用して侵入	侵入	－
		2	DDoSツール(LOIC)等で200リクエスト/秒を複数の端末から一斉送信してシステムダウンさせる	攻撃	CANバスへ大量のメッセージを流す(DoS攻撃)ことで、制御系機能(アクセル、ブレーキ、ステアリング、シフト)を動作不能(性能低下)にさせる
ウェブサイトの改ざん	インターネット	1-1	標的型攻撃によりマルウェアに感染して管理者ID・PWDを盗まれ、サイトを直接改ざんされる	侵入 攻撃	－
		1-2	WEBアプリケーションの脆弱性をつく攻撃(SQLインジェクション等)によりサイトを改ざんされる	侵入 攻撃	偽のCANメッセージにより事前に車の状態を偽装しておくことで、制御系機能(アクセル、ブレーキ、ステアリング、シフト)を誤動作させる
公知の脆弱性を悪用	インターネット	1	公知の脆弱性を利用して侵入して攻撃	侵入 攻撃	マッピング不可
ランサムウェアを使った詐欺・恐喝	インターネット(メール)	1-1	電子メールの添付ファイルを開封してランサムウェアに感染	侵入	－
	インターネット(サイト)	1-2	改ざんされたWEBサイトにアクセスしてランサムウェアに感染	侵入	－
		2	PC上のファイルを勝手に暗号化してユーザに通知し、身代金を要求する	攻撃	偽のCANメッセージを流して、制御系機能(アクセス、ブレーキ、ステアリング、シフト)の実行条件に抵触する状態を作り出すことで、動作不能にする
インターネットバンキング・クレジットカード情報の不正利用	インターネット(サイト)	1	偽のWEBサイトに誘導し、カード番号や口座番号、暗証番号を不正に収集する（フィッシング詐欺）	攻撃	マッピング不可
ウェブサービスへの不正ログイン	インターネット	1	不正なツールによるパスワード解析により不正ログインされる	侵入	マッピング不可
過失による情報漏えい	その他	1	盗難や置忘れ等	その他	マッピング不可

次に、過去に発生した車両攻撃事例およびその攻撃手順[22]を整理し、直接的な攻撃手順を元にテストデータの方向性を検討した。表 3.2c2-5 に過去の車両攻撃事例を示す。

表 3.2c2-5　過去の車両攻撃事例

攻撃事例	侵入経路	No	攻撃手順	分類	攻撃データ
Jeep Cherokee 遠隔操作	Wi-Fi/ 携帯電話網	1	ポートスキャンしてアクセス可能なポートを検索	侵入	－
		2	ヘッドユニット経由でCANに接続されるCPUファームウェアを書き換え		－
		3	CANメッセージをキャプチャーして解析	盗聴	－
		4	不正CPUファームから偽のCANメッセージを送信して車両を遠隔操作(ドアロック解除、ブレーキ無効化、アクセル/ステアリング操作)	攻撃	偽のCANメッセージを流すことにより、制御系機能(アクセル、ブレーキ、ステアリング、シフト)を操作する
BMW 遠隔ドアロック解除	携帯電話網(2G)	1	なりすまし基地局を設置	侵入	－
		2	車両からの接続要求に対して偽の接続許可応答を送信		－
		3	メッセージ(HTTP)を盗聴して操作APIを割り出す	盗聴	－
		4	偽の操作APIを発行して正規のルートから車両のドアロックを解除	攻撃	操作APIを受け取った後のCANメッセージは正規のメッセージであるため対象外
フォード　エスケープ 操作	OBD-Ⅱ	1	OBD-Ⅱにケーブルを接続 or (通信機能を持つOBD-Ⅱドングル)	侵入	－
		2-1	ハンドル監視のECUへ大量のリクエストを送信してパワーステアリングを停止させ、ハンドル操舵の範囲を制限	攻撃	CANバスへ大量のメッセージを流す(DoS攻撃)ことで、制御系機能(アクセル、ブレーキ、ステアリング、シフト)を動作不能(性能低下)にさせる
		2-2	偽のCANメッセージによりブレーキのブリーディング(気泡除去機能)を指示することで、ブレーキを動作させない	攻撃	偽のCANメッセージを流して、制御系機能(アクセス、ブレーキ、ステアリング、シフト)の実行条件に抵触する状態を作り出すことで、動作不能にする
トヨタ　プリウス 操作	OBD-Ⅱ	1	OBD-Ⅱにケーブルを接続 or (通信機能を持つOBD-Ⅱドングル)	侵入	－
		2	偽のCANメッセージにより低速走行かつバック中と状態を偽装することで、自動駐車ハンドル支援機能を誤動作させる	攻撃	偽のCANメッセージにより事前に車の状態を偽装しておくことで、制御系機能(アクセル、ブレーキ、ステアリング、シフト)を誤動作させる
三菱　アウトランダー 遠隔操作	Wi-Fi	1	クラッキングツールを使用して無線LANの事前共有鍵を解読	侵入	－
		2	メッセージを盗聴して操作APIを割り出す	盗聴	－
		3	偽の操作APIを発行して正規のルートからライトを消灯する	攻撃	操作APIを受け取った後のCANメッセージは正規のメッセージであるため対象外

ICT における情報セキュリティ脅威、および車両攻撃事例と、各攻撃パターンの対応付けを表 3.2c2-6 に示す。

大きく 4 種類の攻撃パターンにより、ICT における情報セキュリティ脅威と車両攻撃事例を網羅できる。

表 3.2c2-6　各攻撃パターンの対応付け

ICTにおける情報セキュリティ脅威	なりすまし	不正設定	状態偽装	DoS攻撃	コメント
標的型攻撃による情報流出	●	–	–	–	
内部不正による情報漏えい	–	–	–	–	車両攻撃にマッピング不可
ウェブサービスからの個人情報の窃取	●	–	–	–	
サービス妨害攻撃によるサービス停止	–	–	–	●	
ウェブサイトの改ざん	–	–	●	–	
公知の脆弱性を悪用	–	–	–	–	車両攻撃にマッピング不可
ランサムウェアを使った詐欺・恐喝	–	●	–	–	
クレジットカード情報の不正利用	–	–	–	–	車両攻撃にマッピング不可
ウェブサービスへの不正ログイン	–	–	–	–	車両攻撃にマッピング不可
過失による情報漏えい	–	–	–	–	車両攻撃にマッピング不可

車両攻撃事例	なりすまし	不正設定	状態偽装	DoS攻撃	コメント
Jeep Cherokee遠隔操作	●	–	–	–	
BMW 遠隔ドアロック解除	–	–	–	–	CANより上位への攻撃
フォード　エスケープ操作	–	●	–	●	
トヨタ　プリウス操作	–	–	●	–	
三菱　アウトランダー遠隔操作	–	–	–	–	CANより上位への攻撃

DoS 攻撃については様々な種類の攻撃手法が存在する為、ICT における IDS/IPS 装置の評価項目を参考に、車両観点の DoS 攻撃パターンにマッピングする。

表 3.2c2-7 に DoS 攻撃パターンを示す。

表 3.2c2-7　Dos 攻撃パターン

攻撃分類	No	ICT攻撃パターン	車両攻撃パターン
不正パケット	1	不正IP/TCP/UDPヘッダ長	×（CANに対応するフィールドなし）
	2	不正IP/TCP/UDPデータ長	不正なデータ長のCANパケットを送信
	3	不正IPバージョン番号	×（CANに対応するフィールドなし）
	4	不正IP/TCP/UDPチェックサム値	不正なCRC値のCANパケットを送信
	5	不正送信元IPアドレス	×（CANに対応するフィールドなし）
	6	不正送信先IPアドレス	当該のECUで受け取るはずがないCANメッセージを送信
	7	Land攻撃	×（CANに対応するフィールドなし）
	8	Ping of Death攻撃	最大長を超えるパケット長のCANパケットを送信
	9	不正TCPシーケンス番号	×（CANに対応するフィールドなし）
	10	不正TCP/UDPポート番号	×（CANに対応するフィールドなし）
	11	Fraggle攻撃	×（CANに送信元を設定するフィールドなし）
	12	FTP Bounce攻撃	×（車載ネットワークにFTPサーバなし）
	13	不正IP/TCPオプション	×（CANに対応するフィールドなし）
	14	URL OverFlow攻撃	最大長を超えるデータ長のCANパケットを送信
	15	Spam Mail攻撃	No.14と同じ
	16	MIME Overflow攻撃	No.14と同じ
不正フラグメントパケット	17	Overlapped Fragment攻撃	×（CANにフラグメントはない）
	18	Empty Fragment攻撃	×（CANにフラグメントはない）
	19	Very Small IP Fragment攻撃	×（CANにフラグメントはない）
	20	Too Many IP Fragment攻撃	×（CANにフラグメントはない）
	21	Very Small TCP Fragment攻撃	×（CANにフラグメントはない）
パケット大量送信	22	Smurf攻撃	大量のCANパケットによる要求を送信し、そのACK応答を含め、パケット流量を増やして負荷をかける
	23	UDP Bomb攻撃	特定のフィールドに無効な値を含むCANパケットを大量送信
	24	TCP/UDP Flood攻撃	No.22と同じ

車両攻撃にマッピングが不可なパターン

以上の最新トレンドの ICT における情報セキュリティ脅威、車両攻撃事例、ICT で用いられている IDS/IPS 装置の評価項目を元に表 3.2c2-8 の攻撃パターンを抽出し、今回の評価環境（実験システム）におけるテストデータ作成の対象とした。

図 3.2c2-8 に CANoe のオートメーションシーケンス機能を利用して作成した攻撃用のテストデータの一例を示す。

表 3.2c2-8　テストデータ作成の対象とした攻撃パターン

攻撃パターン			攻撃の狙い
なりすまし操作			攻撃ノードから有効な値のメッセージを不正送信し、故意に車両を誤動作させる
不正設定			攻撃ノードから有効な値のメッセージを不正送信し、機能または操作性に関わる設定を故意に変更し、ユーザーがその機能または操作を実行する際に、意図とは異なる動作をさせる
車両状態の偽装			攻撃ノードから有効な値のメッセージを不正送信し、車両状態情報（UI表示）を故意に偽装する
DoS攻撃	不正データ	不正データ長	攻撃ノードからDLCとデータ長が不一致したメッセージを不正送信し、CANバス上の受信ノードに故意にエラーを発生させる
		不正CRC値　※1	攻撃ノードからCRCフィールドが不正な（エラーを起こす）メッセージを不正送信し、CANバス上の受信ノードに故意にエラーを発生させる
		想定外メッセージ　※2	攻撃ノードから既存のノードが送信するメッセージを不正送信し、そのメッセージの本来の送信ノードに故意にエラーを発生させる
		最大データ長超え	攻撃ノードから最大データ長（8バイト）を超えるデータフィールドのメッセージを不正送信し、CANバス上の受信ノードに故意にエラーを発生させる
	データ大量送信	有効データの大量送信	攻撃ノードから有効な値のメッセージを大量に不正送信し、受信ノード（ECU）に処理遅延を引き起こす、あるいは、攻撃メッセージより低優先のメッセージがCANバスに流れることを妨害する
		無効データの大量送信	攻撃ノードから無効な値のメッセージを大量に不正送信し、受信ノード（ECU）に処理遅延を引き起こす、あるいは、攻撃メッセージより低優先のメッセージがCANバスに流れることを妨害する、もしくはCANバス上の受信ノードに故意にエラーを発生させる
		未使用IDの大量送信	攻撃ノードから未使用IDのメッセージを大量に不正送信し、受信ノード（ECU）に処理遅延を引き起こす、あるいは受信ノードの処理を妨害する
		エラーフレームの大量送信	攻撃ノードからエラーフレームを大量に不正送信し、受信ノードに故意にエラーを発生させる、あるいは受信ノードの処理を妨害する

※1：評価環境（実験システム）のCAN-USB変換が対応していない為、作成なし

※2：自動運転の簡易モデル車両での想定外メッセージは状態通知メッセージのみに限定されるため、攻撃データは「車両状態の偽装」のものを使用

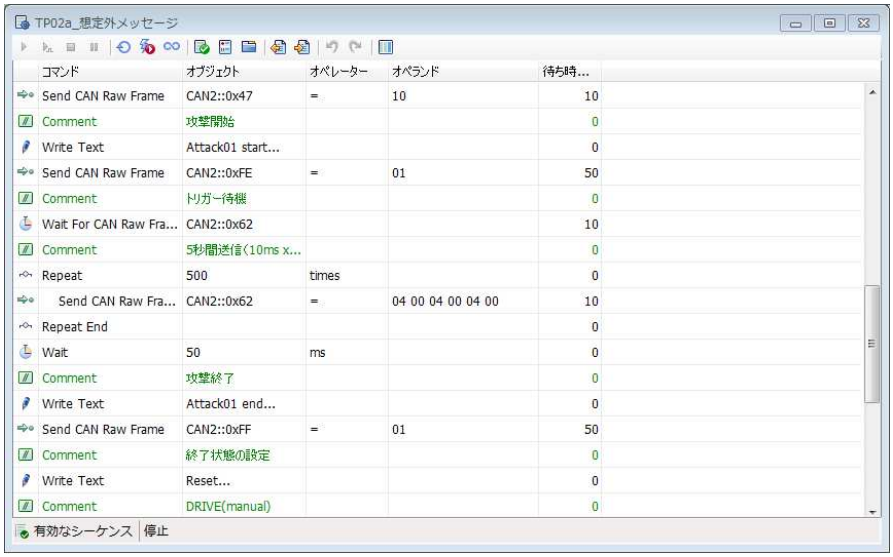


図 3.2c2-8　攻撃用テストデータの一例

① 攻撃データ注入時の挙動

表 3.2c2-8 で示した攻撃パターンに基づいて、具体化した攻撃を 54 パターン作成し、実行・解析した。この結果を、周期乱れの発生や攻撃を検知できているか、などの挙動に基づいて体系化を行い、代表的に以下の 4 つのパターンにまとめ分析を行った。

(ⅰ) 単発攻撃
(ⅱ) 周期攻撃
(ⅲ) DoS 攻撃（有効データの大量送信）
(ⅳ) DoS 攻撃（エラーフレームの大量送信）

各攻撃を行った場合に実験システムの CAN Bus 上で観測された影響を以下にまとめる。

(ⅰ) 単発攻撃
・攻撃の詳細
ホイールエンコーダのカウント値（CAN ID 0x161）を受信した 10ms 後に、不正データメッセージを送信する。なお、送信時の DLC は 9、10、11、12 でそれぞれ試行する。
・攻撃した際の CAN Bus の挙動
いずれの攻撃メッセージに対しても、周期性を持つ各正規メッセージの周期乱れが発生することはなかった。

(ⅱ) 周期攻撃
・攻撃の詳細
「正規のペダルストローク状態通知」並びに「ペダル状態通知」の直後に状態と異なる値を周期送信する（500 回）。
・攻撃した際の CAN Bus の挙動
正規のメッセージに周期的に攻撃メッセージを挿入した場合も、正規メッセージの周期乱れなどは発生しなかった。

(ⅲ) DoS 攻撃（有効データの大量送信）
・攻撃の詳細
状態と異なる有効データを 500 回連続送信する。
・攻撃した際の CAN Bus の挙動
DoS 攻撃中は 0.1ms の間隔でメッセージが送られてくるため正規のメッセージは送信不可能な状況になっている。
DoS 攻撃が終了した後に正規メッセージが再び送信され始めるが、暫くの間メッセージの周期は短くなり、時間経過とともに通常状態に戻る。

(ⅳ) DoS 攻撃（エラーフレームの大量送信）
・攻撃の詳細
エラーフレームを 500 回連続で送信する。
・攻撃した際の CAN Bus の挙動
エラーフレームは約 0.5ms 周期で送信されていることを確認したが、周期メッセージの周期は攻撃が行われていないときと比べて大きく変化することはなかった。それに伴う周期のズレも検知できなかったため、侵入検知の通知メッセージは一度も送信されなかった。

(3) 考察

① 単発攻撃

この攻撃を行った際に CAN Bus 上で正規メッセージの周期乱れなどの変化は現れなかった。

表 3.2c2-9 は、ともに送信周期が 10ms であるホイールエンコーダのカウント値(CAN ID 0x161)と Main バッテリー電流(CAN ID 0x126)に対して攻撃が行われた時の周期の揺らぎを示している。

表 3.2c2-9　単発攻撃時の正規メッセージの周期変化

			ホイールエンコーダのカウント値（CAN ID 0x161）			Main バッテリー電流 (CAN ID 0x126)		
イベント	タイムスタンプ [s]	間隔時間 [ms]	タイムスタンプ [s]	間隔時間 [ms]	周期の揺らぎ [%]	タイムスタンプ [s]	間隔時間 [ms]	周期の揺らぎ [%]
攻撃開始	23.43229	—	—					
—			23.439	10.000	0.00	23.434	10.012	0.12
			23.449	9.996	0.04	23.444	10.011	0.11
			23.459	10.000	0.00	23.454	10.008	0.08
			23.469	9.998	0.02	23.464	10.008	0.08
			23.479	9.999	0.01	23.474	10.010	0.10
			23.489	10.002	0.02	23.484	10.007	0.07
			23.499	9.997	0.03	23.494	10.012	0.12
攻撃メッセージ注入	23.50035	1.211	—					
攻撃検知	23.50071	0.358	—					
—			23.509	9.999	0.01	23.504	10.009	0.09
			23.519	10.002	0.02	23.514	10.007	0.07
			23.529	9.996	0.04	23.524	10.012	0.12
			23.539	9.999	0.01	23.534	10.007	0.07
			23.549	10.002	0.02	23.544	10.012	0.12
攻撃終了	23.5503	—	—					

表の結果からわかるように、攻撃を注入した後も周期の変化はなく、攻撃期間中でもっとも大きな揺らぎでも 0.12%と小さく、影響はないと言える。

侵入検知精度を評価する各パラメータについては以下のようになった。

(ⅰ) メッセージ単位での評価

メッセージ単位での評価結果を表 3.2c2-10 に示す。

表 3.2c2-10　単発攻撃に対するメッセージ単位での評価結果

パラメータ	値
True Positive	1
True Negative	11
Condition Positive	1
Condition Negative	11
TPR(検知率)	100%
TNR(正常メッセージ認識率)	100%
FPR(誤検知率)	0%
FNR(見逃し率)	0%

攻撃メッセージ 1 回を攻撃と検知しているため、TPR は 100%となった。

(ⅱ) 単位時間での評価

単位時間での評価結果を表 3.2c2-11 に示す。正規メッセージの 1 周期分、3 周期分、6 周期分、10 周期分を単位時間として評価を行った。

表 3.2c2-11　単発攻撃に対する単位時間での評価結果

パラメータ	値			
	1 周期	3 周期	6 周期	10 周期
True Positive	1	1	1	1
True Negative	11	2	0	0
Condition Positive	1	1	1	1
Condition Negative	11	2	0	0
TPR(検知率)	100%	100%	100%	100%
TNR(正常メッセージ認識率)	100%	100%	—	—
FPR(誤検知率)	0%	0%	—	—
FNR(見逃し率)	0%	0%	0%	0%

単位時間ごとの評価において、攻撃があった区間で攻撃を検知しているため、TPR は 100%となった。単位時間の長さを変化させることで Condition Negative の値も同様に変化するが、検知精度を示す値に影響はない。

従って、今回のような攻撃メッセージが送信されたのち、100%攻撃を検知できている場合、メッセージ単位、単位時間それぞれの精度評価に差異はないことが分かった。

② 周期攻撃

この攻撃を行った際に、CAN Bus 上での正規メッセージの周期乱れなどの変化は現れなかった。実際の攻撃データでは、正規メッセージを受信した約 0.9ms 後に攻撃メッセージを送信している。そのため、わずかな周期幅の揺らぎや遅延によって正規メッセージより先に攻撃メッセージが到着するということも十分に起こりうる。

実攻撃データを確認したところ、大きく分けて以下の二つの現象が起こることが分かった。

実際の攻撃データを見ても、半数近くの攻撃メッセージが正規メッセージよりも先に到着していることが確認できた。

a) 周期のゆらぎによって、正規メッセージよりも先に攻撃メッセージが来ることがある
b) 正規メッセージの欠落によって（＝攻撃メッセージを正規メッセージと認識する）、検知が行われない場合がある

a) のような現象が発生することがあるため、純粋に周期を乱しているメッセージ単体を攻撃メッセージとして検知、あるいは排除するような侵入検知方式は誤検知の可能性が高くなる。

厳密に判定を行う必要がある場合には、周期を乱したメッセージのうち、疑わしいものが複数ある場合はその複数を選別するような検知方法が必要となる。また、b)のようにメッセージ欠落が発生した場合に攻撃メッセージを正規のメッセージと受け入れてしまうことのないような対策が求められる。

指標毎の侵入検知の精度は以下のとおりである。

(ｉ)メッセージ単位での評価

メッセージ単位での評価結果を表 3.2c2-12 に示す。

表 3.2c2-12　周期攻撃に対するメッセージ単位での評価結果

パラメータ	値
True Positive	87
True Negative	93
Condition Positive	500
Condition Negative	470
TPR(検知率)	17%
TNR(正常メッセージ認識率)	20%
FPR(誤検知率)	80%
FNR(見逃し率)	83%

メッセージ単位での評価結果は誤検知の影響が大きく、今回の実証データでは周期の揺らぎによって正規メッセージを攻撃メッセージが追い越すような事象が発生したため、正規メッセージを攻撃メッセージとみなすことが多く発生した。そのため TPR、 TNR が大

きく下がった。

(ⅱ) 単位時間での評価

単位時間での評価結果を表 3.2c2-13 に示す。正規メッセージの 1 周期分、3 周期分、6 周期分、10 周期分を 1 区画として評価を行った。

表 3.2c2-13　周期攻撃に対する単位時間での評価結果

パラメータ	値			
	1 周期	3 周期	6 周期	10 周期
True Positive	499	176	88	52
True Negative	6	2	1	0
Condition Positive	499	176	88	52
Condition Negative	11	3	1	0
TPR（検知率）	100%	100%	100%	100%
TNR（正常メッセージ認識率）	55%	67%	100%	―
FPR（誤検知率）	45%	33%	0%	―
FNR（見逃し率）	0%	0%	0%	0%

各精度の評価指標を比較すると、メッセージ単位での評価に比べて単位時間での評価の方が TPR、TNR の値が良くなっていることが分かる。単位時間での精度評価では、どのメッセージが攻撃のメッセージなのかということは考慮せず、単位時間内の攻撃の有無に着目するため、TPR、TNR の値はメッセージ単位よりも良くなる。

従って周期攻撃に対する検知評価には 2 通りの方法がある。メッセージ単位での評価手法を用いる場合、TPR、TNR が下がる可能性がある。そこで、侵入検知に加えて検知したメッセージの中身を検証して正規メッセージと攻撃メッセージを区別するような市区を設けると、当然 TPR、TNR の値は上がるが、メッセージの内容までの検証を行うスペックが侵入検知ユニットに求められ、コストアップに繋がる。

また、単位時間での評価手法を用いて、メッセージの区別は付けずに区間毎の攻撃の有無だけ検知する方法である。

どちらの方式でも侵入の検知までにかかる時間がどこまで許容されるか、どこまでメッセージ自体の正当性を認識する必要があるかなどの使用する場所に求められる要件により機能としての十分性が変わってくるため、実際に使用するケースにより総合的に判断することが必要である。

③ DoS 攻撃（有効データの大量送信）

状態と異なる値を連続送信するのは、いわゆる DoS 攻撃とみなすことができる。

一度検知メッセージを送信するごとに 2~3 個の攻撃メッセージが含まれていることから、攻撃メッセージの送信周期よりも侵入検知の応答周期の方が長いことは自明である。

DoS 攻撃の周期がおよそ 0.1ms ごとに送信されているのに対して侵入検知の応答メッセ

ージは周期が約 0.24ms であった。このため、侵入検知の応答可能な周期は 0.24ms であると考えられる。

10ms 周期のメッセージに対して、0.1ms 周期でメッセージを送信する場合、正規のメッセージを一回送るまでに 100 回 DoS 攻撃を行うことができる。しかし、今回 DoS 攻撃が 100 回来た後にも正規メッセージが送られることはなかった。おそらく CAN Bus の特性上、送られた全てのメッセージを受信する必要があり、その処理に時間がさらにかかってしまうため、送信することができなかったと考える。

正規のメッセージは DoS 攻撃が終了するとともに再度送信され始める。実際の攻撃データ上では、DoS 攻撃が始まる前に送られた正規メッセージから、500 回全ての DoS 攻撃が終了した後である 46.5ms 後に再度送信された。その次に送られる正規メッセージは約 3.5ms と非常に短い周期で送信されていることが確認できた。その次のメッセージ以降は約 10ms と、通常の送信周期に戻っている。

指標毎の侵入検知の精度は以下のとおりである。

(ⅰ) メッセージ単位での評価

メッセージ単位での評価結果を表 3.2c2-14 に示す。

表 3.2c2-14　DoS 攻撃（有効データの大量送信）に対するメッセージ単位での評価結果

パラメータ	値
True Positive	123
True Negative	16
Condition Positive	500
Condition Negative	17
TPR(検知率)	25%
TNR(正常メッセージ認識率)	94%
FPR(誤検知率)	6%
FNR(見逃し率)	75%

これらの結果からわかるように全ての DoS 攻撃のメッセージの一つ一つを検知することはできず、全ての DoS 攻撃のうち 25%程度を検知している。検知ユニットの応答性能の限界が原因と考えるが、DoS 攻撃においては大量のメッセージの塊を送信することで攻撃が成立するため、必ずしも全てのメッセージを検知する必要はないといえる。

そのため、複数のメッセージの送信状況に鑑みて DoS 攻撃が行われている状況をいち早く検知して、フェールセーフ機能などにつなげていくことが重要と考える。

(ⅱ) 単位時間での評価

単位時間での評価結果を表 3.2c2-15 に示す。正規メッセージの 1 周期分、3 周期分、6 周期分、10 周期分を 1 区画として評価を行った。

表 3.2c2-15　DoS 攻撃（有効データの大量送信）に対する単位時間での評価結果

パラメータ	値			
	1 周期	3 周期	6 周期	10 周期
True Positive	5	2	2	1
True Negative	15	4	1	0
Condition Positive	5	2	2	1
Condition Negative	15	4	1	0
TPR(検知率)	100%	100%	100%	100%
TNR(正常メッセージ認識率)	100%	100%	100%	―
FPR(誤検知率)	0%	0%	0%	―
FNR(見逃し率)	0%	0%	0%	0%

単位時間ごとの評価において、単位時間の長さを変化させることで Condition Negative の値も同様に変化するが、検知精度を示す値に影響はない。原因として精度評価に使う単位時間の範囲を正規メッセージの送信周期よりも大きい範囲で評価を行っていたためであると考える。単位時間を正規メッセージの周期よりも短い時間に設定するとメッセージ単位での精度評価で算出された値に近づいていくと考えられる。周期の取り方は検知メッセージの応答性能に依存する。

従ってメッセージ単位と単位時間での評価において、DoS 攻撃に対する侵入検知の評価精度は侵入検知ユニットの応答性能に依存する。

④ DoS 攻撃（エラーフレームの大量送信）

エラーフレームを 500 回、周期的に送信し続けたが、正規の周期メッセージに対して周期の乱れは観測されなかった。従って、侵入検知のエラーメッセージが送信されることもなかった。表 3.2c2-16 はホイールエンコーダのカウント値（CAN ID 0x161）と Main バッテリーの電流（CAN ID 0x126）の二つの周期メッセージに対してエラーフレームを大量送信する攻撃をした際の周期の揺らぎを示している。周期の揺らぎの値からわかるようにどちらのメッセージも大きく揺らいだところで 1.5～1.7%程度で大きな影響はないことが分かる。エラーフレームを送信することで、タイミングによっては正規メッセージの送信を打ち消してしまっていることが確認できた。しかし、その後の正規メッセージを再送するまでが早く、侵入検知での正常範囲内のうちに再送できているため、検知メッセージが送られていないことが分かる。

表 3.2c2-16　DoS 攻撃（エラーフレームの大量送信）時の正規メッセージの周期変化

ホイールエンコーダのカウント値 (CAN ID 0x161)			Main バッテリー電流 (CAN ID 0x126)		
タイム スタンプ[s]	間隔時間 [ms]	周期の 揺らぎ[%]	タイム スタンプ[s]	間隔時間 [ms]	周期の 揺らぎ[%]
38.93785			38.93468		
38.94784	9.998	0.02	38.94469	10.015	0.15
38.958	10.001	0.01	38.955	10.011	0.11
38.96784	9.994	0.06	38.96471	10.009	0.09
38.97784	10.000	0.00	38.97472	10.007	0.07
38.98799	10.154	1.54	38.98473	10.012	0.12
38.99798	9.984	0.16	38.99474	10.007	0.07
39.008	10.023	0.23	39.00475	10.015	0.15
39.01798	9.976	0.24	39.01476	10.007	0.07
39.028	10.021	0.21	39.02477	10.007	0.07
39.038	9.996	0.04	39.035	10.175	1.75
39.048	9.994	0.06	39.045	10.000	0.00
39.058	10.010	0.10	39.05494	9.999	0.01
39.06783	9.837	1.63	39.0648	9.862	1.38
39.07783	10.000	0.00	39.07481	10.010	0.10
39.08783	9.993	0.07	39.08482	10.012	0.12
39.09783	10.003	0.03	39.09483	10.007	0.07
39.108	9.999	0.01	39.105	10.010	0.10
39.11783	10.001	0.01	39.11485	10.010	0.10
39.12783	10.000	0.00	39.12486	10.009	0.09

指標毎の侵入検知の精度についてメッセージ単位での評価結果を表 3.2c2-17 に示す。

表 3.2c2-17　DoS 攻撃（エラーフレームの大量送信）に対するメッセージ単位での評価結果

パラメータ	値
True Positive	0
True Negative	20
Condition Positive	370
Condition Negative	20
TPR(検知率)	0%
TNR(正常メッセージ認識率)	100%
FPR(誤検知率)	0%
FNR(見逃し率)	100%

エラーフレームを大量送信する攻撃の場合、検知率は 0%という結果になった。エラーフレームを大量に送信したにも関わらず、正規メッセージの周期にはほぼ影響がなかったため検知できなかったと考える。これはエラーフレームを大量に送信しても正規メッセージと衝突せず、衝突しても正規メッセージが許容範囲内で即座に再送されるため異常と検知しないためである。

攻撃メッセージに対して検知メッセージは一度も送信されていないためにどのような精度評価手法を持っても結果は同じになることは明らかである。

今回の実験システムにおいてエラーフレームの大量送信は CAN メッセージに影響がないという結論になった。しかし、エラーフレームの送信周期をあげると正規メッセージとの衝突が増え､正規メッセージの再送にも衝突するようになれば影響が出る可能性はある。

攻撃を行っていても、正規メッセージに影響がない攻撃は許容するのか、それとも影響がなくとも攻撃とみなすメッセージは全て検知するべきなのか、侵入検知で実現するべきポリシーによって侵入検知のメカニズムは検討していく必要があると考える。

⑤ まとめ

評価指標による検知精度の特徴から、周期乱れの発生や検知メッセージ送信などの挙動に基づいて評価結果の優劣をチェックできたのは DoS 攻撃（エラーフレームの大量送信）を除いた三つの攻撃事例であった。

1) 単発攻撃
2) 周期攻撃
3) DoS 攻撃（有効データの大量送信）

各攻撃事例についてメッセージ単位と単位時間での評価を行い、攻撃毎の評価結果の特徴を以下に纏める。

(ｉ) メッセージ単位での評価の特徴

メッセージ単位での評価における検知結果の特徴を表 3.2c2-18 に示す。

表 3.2c2-18　メッセージ単位での評価における侵入検知結果の特徴

攻撃方法		特徴
単発攻撃		攻撃タイミングが許容範囲外であれば侵入検知は容易であり、TPR は高い検知率が期待できる
周期攻撃	許容範囲外	攻撃メッセージが正規メッセージの許容範囲外であれば、周期による侵入検知は容易であり、TPR は高くなる
	許容範囲内、同時タイミング	攻撃メッセージが正規メッセージの許容範囲内や同時タイミングであれば、メッセージの僅かな周期乱れで正規メッセージと攻撃メッセージの順序が入れ替わる恐れがある 先に攻撃メッセージが送信されると、後発の正規メッセージを誤検知するため、TPR、TNR が大きく下がる
DoS 攻撃		FNR は検知応答に依存し、大きく上がる 攻撃メッセージにより CAN Bus が占有されると正規メッセージの周期が乱れて TNR が下がる

メッセージ単位での評価の問題は、正規メッセージと周期的な攻撃メッセージの送信タイミングが近いと両者の区別がつかないことによって誤検知、見逃しにつながる。これを解決するためにはメッセージの中身の検証が必要となり、侵入検知ユニットに高性能のプロセッサが必要となる。これはコスト増加に繋がるため、システムに応じて適切な機能を選ぶ必要がある。

DoS 攻撃の検知率は検知メッセージの送信性能に依存するが、必ずしも全てのメッセージを通知できなくても検知の目的は達成できるといえる。

(ⅱ) 単位時間での評価の特徴

単位時間での評価における検知結果の特徴を表 3.2c2-19 に示す。

表 3.2c2-19　単位時間での評価における侵入検知結果の特徴

攻撃方法		特徴
単発攻撃		メッセージ単位での評価手法と同様で、TPR は高くなる メッセージ単位と単位時間での評価結果に違いは見られない
周期攻撃	許容範囲外	正規メッセージの許容範囲外における周期攻撃に対する単位時間での評価では、メッセージ単位での評価結果と違いは見られない
	許容範囲内、同時タイミング	正規メッセージの許容範囲内や同時タイミングにおける周期攻撃に対する単位時間での評価では、1 区間内で 1 回でも検知があれば攻撃を攻撃と判定できたとすることから、TPR は高くなる
DoS 攻撃		区間内で 1 回でも検知があれば攻撃されたと判定できることから TPR は高くなるが、単位時間を短くしていくとメッセージ単位での評価に近づき TPR は下がっていくと考えられる。

単位時間での評価は攻撃パターンに寄らず、良い評価が出る傾向にある。例えば 10 周期毎に評価を行えば検知率は全て 100%であったが、0.1s 後にしか評価結果が分からないため検知までに時間が掛かる。

周期の取り方は検知メッセージの応答性能に依存するためメッセージ単位での評価同様にシステムに応じて適切な機能を選ぶ必要がある。

3.2c.3　サイバー攻撃に対する情報共有に関する検討

毎年、IPA の情報セキュリティ 10 大脅威[23]で報告されている内容が移り変わっている事からも判るように、サイバー攻撃のトレンドは日々変化している。車両向けの情報セキュリティの研究を行う上で、研究内容が現実のサイバー攻撃のトレンドから外れたものにならないように、その方向性を確認・認識しながら進めることは重要である。

そのためには ICT における最新の情報セキュリティ関連情報を、信頼できる組織から提供を受けてプロジェクトメンバで共有し、その分析や対策について車載への適用および車載でしか発生しえないリスクなども議論していく必要がある。このような検討をする上で、本事業では一般社団法人 JPCERT コーディネーションセンター 以下、JPCERT/CC という）にアドバイスを求めつつ、情報共有に関する検討を行った。

この情報共有の取り組みの中で、通常は 1 対 1 の守秘義務契約に基づいて行われる情報のやりとりを会社や組織を跨がる場合に共通に議論できるようにするためにはどのように進めればいいのかについて調査・検討し、具体的な規約の例についてまとめたので、その結果を報告する。

(1) 考え方と対象

昨年度の本事業の一環で、組織間での情報セキュリティ共有について運用までを含めて調査した。

その結果と JPCERT/CC のコメントを踏まえて、IPA が情報ハブとなり民間組織とともに運営しているサイバー情報共有イニシアティブ（Initiative for Cyber Security Information sharing Partnership of Japan 以下、J-CSIP という）の体制[24]（図 3.2c.3-1）を参考に、今回のプロジェクトメンバおよび関係団体に適用することとした。

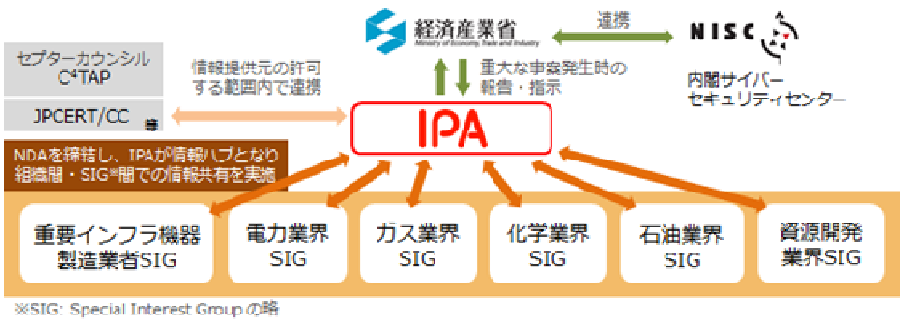


図 3.2c.3-1　J-CSIP の体制

情報共有については上流から下流というような単純な一本線の流れではなく、情報の提供側と受領側での立場で考える必要があり、以下この二つの視点で考察する。

(2) 情報提供側の情報管理

今回アドバイスをいただいた JPCERT/CC からは、Traffic Light Protocol（以下 TLP という）に基づき、情報の扱いにレベル付けを指定して共有する提案を受けた。

具体的には以下の四つの参照・取り扱いレベルである。

- RED：情報提供元からセキュリティ情報を受け取った人のみに限定する情報。受信者は高レベルの保護を行う必要がある。
- AMBER：知る必要がある人のみに限定する情報
- GREEN：各層における関係者と共有可能な情報
- WHITE：公共向けの情報

各情報レベルに対して、今回は以下の観点で取り扱いに規定が定められている。

a) 情報共有可能範囲
b) 情報伝達時の暗号化の必要性の有無
c) 共有範囲内の他者への口頭での伝達の可否
d) 共有範囲内の他者への電子データでの伝達の可否
e) 一般への情報開示／公開の可否

各参照レベルと観点ごとの取り扱いについて表 3.2c.3-1 にまとめる。

表 3.2c.3-1　共有範囲の TLP レベルとその情報の取り扱い

レベル / 要素	RED	AMBER	GREEN	WHITE
①情報共有可能範囲	受信者のみ	自組織及び関連組織内の必要最小限	コミュニティワイド	制限無し
②伝達時の暗号化	対象外(伝達自体不可)	必要に応じて実施	不要	不要
③他者への口頭での伝達	不可	可能	可能	可能
④共有範囲内の他者への電子データでの伝達	不可	可能	可能	可能
⑤一般への情報開示／公開	不可	不可	不可	可能

この TLP の考え方は情報の取り扱いに関するレベル付けの観点で判り易く、例えば内閣サイバーセキュリティセンター（以下 NISC という）により 2011 年 4 月（当時の名称は内閣官房情報セキュリティセンター）に発行されている「『重要インフラの情報セキュリティ

対策に係る行動計画』の情報連絡・情報提供に関する実施細目」[25]などでも使用されている。

(3) 情報受領側の情報管理

一般的に企業などの組織体では、自社内や契約などにより他社または顧客などの社外から得た情報の管理について規定する、いわゆる情報管理規定に類する内部規定を定めていることが多い。

今回提供される情報については、社外から得た情報に相当する。従って、TLP の各レベルの情報の扱いに対応できる内部規定が既存にあれば、各社の情報管理規定に従って管理すれば良い。 一方、不足している部分があれば TLP で規定する扱いに適合できるよう管理規定を作成する必要がある（図 3.2c.3-2 参照）。

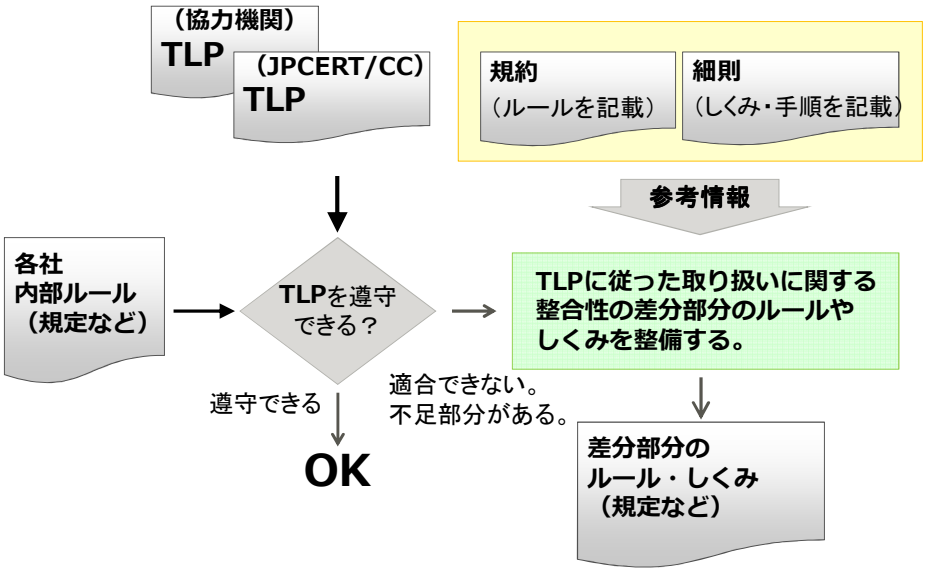


図 3.2c.3-2　規約作成の手順

今回はプロジェクトの参加メンバが誤解なく、かつこの取り組みを進めやすくするため、内部規定が無いことを仮定して、まず TLP で規定する取り扱いを遵守するための規約を作成し、次にその規約に対して、より具体的な取り扱い方法の例までを含めた細則案（プロジェクト参加メンバへ例示の位置付け）を作成した。

(4) 情報の取り扱いの検討方法

今回は NISC から 2011 年 4 月に発行されている『「情報の格付け及び取扱制限に関する規程」策定手引書』[26]をベースに情報の取り扱い事例を抽出し、過不足について精査した後に具体的な取り扱い方法、対応方法を検討した。

(5) 規約の検討要素

規約の作成にあたり検討した要素は以下のとおりである。

a) 目的（Why）

b) 適用範囲

b)-1. 組織（Who，Where）

b)-2. 情報（What）

b)-3. 期間（When）

c) 遵守事項の考え方

c)-1. セキュリティ情報を取扱う人の制限（Who）

c)-2. セキュリティ情報の分類（What）

d) 規定している取扱いシーン（How）

d)-1. 保管

d)-2. 利用

d)-3. 廃棄

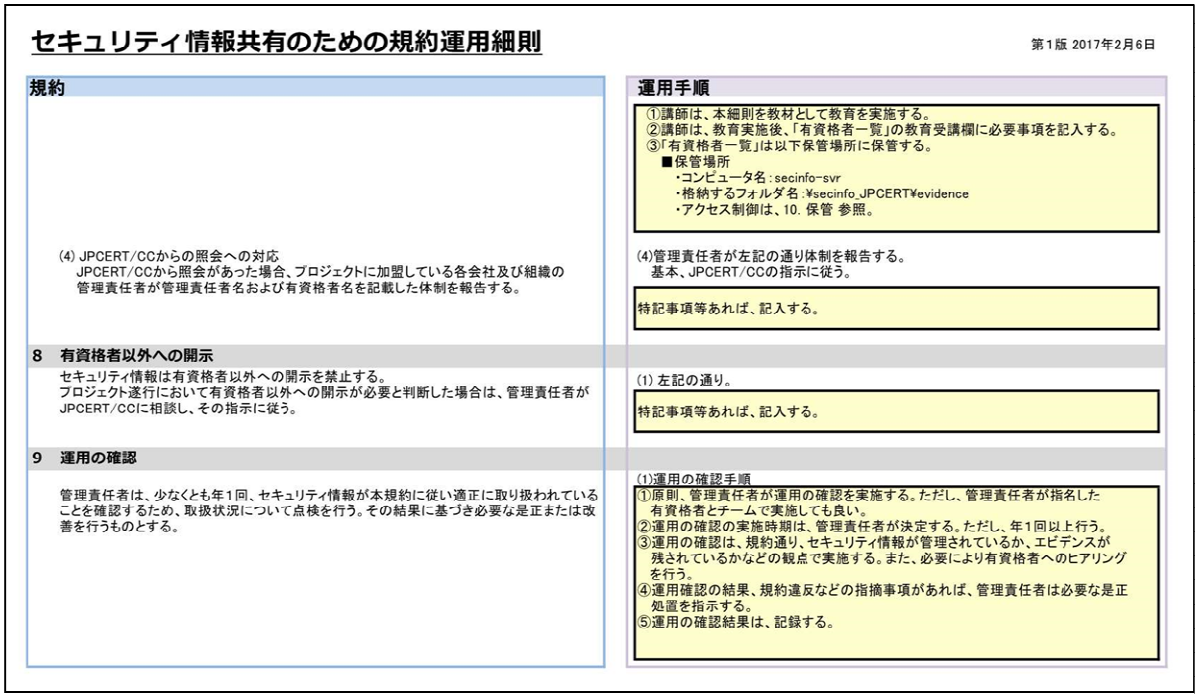
(6) 規約案と細則案

今回作成したセキュリティ情報取扱規約案の目次を図 3.2c.3-3 に示す。

目次

図 3.2c.3-3　セキュリティ情報取扱規約案の目次

この規約案の各項目について更に具体的な例示を加えたものが細則案である。今回作成した細則案の一部を図 3.2c.3-4 に例示する。



セキュリティ情報共有のための規約運用細則

第1版 2017年2月6日

規約	運用手順
	①講師は、本細則を教材として教育を実施する。 ②講師は、教育実施後、「有資格者一覧」の教育受講欄に必要事項を記入する。 ③「有資格者一覧」は以下保管場所に保管する。 ■保管場所 ・コンピュータ名：secinfo-svr ・格納するフォルダ名：¥secinfo_JPCERT¥evidence ・アクセス制御は、10. 保管 参照。
(4) JPCERT/CCからの照会への対応 JPCERT/CCから照会があった場合、プロジェクトに加盟している各会社及び組織の管理責任者が管理責任者名および有資格者名を記載した体制を報告する。	(4)管理責任者が左記の通り体制を報告する。 基本、JPCERT/CCの指示に従う。 特記事項等あれば、記入する。
8 有資格者以外への開示	
セキュリティ情報は有資格者以外への開示を禁止する。 プロジェクト遂行において有資格者以外への開示が必要と判断した場合は、管理責任者がJPCERT/CCに相談し、その指示に従う。	(1) 左記の通り。 特記事項等あれば、記入する。
9 運用の確認	
管理責任者は、少なくとも年1回、セキュリティ情報が本規約に従い適正に取り扱われていることを確認するため、取扱状況について点検を行う。その結果に基づき必要な是正または改善を行うものとする。	(1)運用の確認手順 ①原則、管理責任者が運用の確認を実施する。ただし、管理責任者が指名した有資格者とチームで実施しても良い。 ②運用の確認の実施時期は、管理責任者が決定する。ただし、年1回以上行う。 ③運用の確認は、規約通り、セキュリティ情報が管理されているか、エビデンスが残されているかなどの観点で実施する。また、必要により有資格者へのヒアリングを行う。 ④運用確認の結果、規約違反などの指摘事項があれば、管理責任者は必要な是正処置を指示する。 ⑤運用の確認結果は、記録する。

図 3.2c.3-4　細則案の例（一部抜粋）

参考文献

[1] SBD ジャパン： “サイバーセキュリティガイド” 2017.

[2] ISO 14229 “Road vehicles -- Unified diagnostic services (UDS)
ISO 14230 “Road vehicles -- Diagnostic systems -- Keyword Protocol 2000

[3] ISO 15765 “Road vehicles -- Diagnostic communication over Controller Area Network (DoCAN)

[4] http://www.nisc.go.jp/inquiry/pdf/so_honbun.pdf

[5] https://www.ipa.go.jp/security/controlsystem/press_1.html

[6] https://www.ipa.go.jp/security/technicalwatch/20170123.html

[7] http://www.trendmicro.co.jp/jp/about-us/press-releases/articles/20160727064652.html

[8] FROST & SULLIVAN： “Automotive Over-the-air Updates” 2016.（邦題：自動車向け OTA (over-the-air) の世界市場の最新動向）

[9] SBD ジャパン： “自動車 OTA 動向 – 要件、ソリューション、今後の課題” 2017

[10] https://www.ipa.go.jp/about/press/20110920.html

[11] SBD ジャパン： “CAN バスのセキュリティ対策” 2017.

[12] https://www.ipa.go.jp/security/fy18/reports/contents/remote/Chapter7/8.htm

[13] 「自動運転車社会における責任問題はどう解決すべきか？」一般社団法人 電子情報技術産業協会

[14] 日経 BP：“自動運転の未来 2016－2020” 2015.

[15] ISO/IEC9798 “Information technology -- Security techniques

[16] http://www.nhtsa.gov/staticfiles/nvs/pdf/812333_CybersecurityForModernVehicles.pdf

[17] https://www.nist.gov/sites/default/files/documents/cyberframework/Cybersecurity-Framework-for-FCSM-Jan-2016.pdf

[18] National institute of Standards and Technology, “Cybersecurity Framework”, https://www.nist.gov/cyberframework

[19] https://www.zmp.co.jp/products/robocar-mv

[20] https://www.ipa.go.jp/security/vuln/10threats2016.html

[21] http://www.nisc.go.jp/active/general/sbd_sakutei.html

[22] ISO/IEC15408 “Information technology -- Security techniques -- Evaluation criteria for IT security

[23] SBD ジャパン： “サイバーセキュリティガイド” 2017

[24] 情報セキュリティ 10 大脅威 2017 https://www.ipa.go.jp/security/vuln/10threats2017.html

[25] サイバー情報共有イニシアティブ https://www.ipa.go.jp/security/J-CSIP/

[26] http://www.nisc.go.jp/conference/seisaku/kihon/dai9/pdf/9siryou_ref04.pdf

[27] 「情報の格付け及び取扱制限に関する規程」策定手引書：
http://www.nisc.go.jp/active/general/pdf/dm3-01-101_manual.pdf

3.2d　車内通信プロトコルの仕様に基づく評価方法の検討

自動走行に向けて運転支援システム等が高度化することに伴い、これまで以上に車両内の ECU やセンサ等のコンポーネントが協調動作することの重要性が増してくる。この協調動作には車内のネットワークを通じたコンポーネント間の通信が不可欠であり、当該ネットワーク上の通信プロトコルがより重要な役割を担うようになると考えられる。そのため、自動車に対するサイバー攻撃により、車内ネットワークの通信プロトコルがどのような影響を受けるかを明らかにすることは、つながる車の車両のセキュリティ対策を考える上で非常に重要である。

しかしながら、従来の車内ネットワークのセキュリティに関する研究では、特定の通信プロトコルに対する攻撃方法のみが検討の対象となっており、その他の通信プロトコルについては、脆弱性の有無やセキュリティ対策に関する評価の方法や基準についての検討はほとんど行われておらず、明らかになっていなかった。また、既知の脆弱性や攻撃方法に関して評価することが可能な評価環境を構築する方法についても確立されていなかった。

そこで、自動走行が実現された際に主流になることが予測される車内ネットワークの通信プロトコルを主な対象とし、通信プロトコルの仕様およびマイコンのアプリケーションにおける処理方法のセキュリティ対策に関する評価方法や評価基準を明らかにし、また、それらに基づく評価環境の開発に必要となる要件を検討するために、平成 27 年度の成果をベースに以下の検討と検証を実施した。

・評価方法の検討（3.2d.1）

セキュリティ対策を評価する際に必要となる攻撃方法に関し、現時点における代表的な車内通信プロトコルである CAN を対象に、可能な限り網羅的で体系的な攻撃方法の整理・分類の検討を行った。また、それら攻撃方法について、自動走行車に利用されることが予測される他の通信プロトコル（CAN-FD、LIN、FlexRay）での適用可能性および拡張可能性を机上検討し、従来明確には指摘されていなかった新たなものを含む幅広い攻撃方法の可能性を明らかにすると共に、攻撃方法の分類を拡張し、網羅性の向上を図った。

・シミュレータによる評価方法の有効性検証（3.2d.2）

CAN を対象に、前記の攻撃方法の分類それぞれに対応する具体的な攻撃シナリオの選定および新規考案と、それらをシミュレータ上で再現するための詳細な手順の検討を行った。また、前記の手順に基づき、シミュレータ上での攻撃の再現性や被害の可観測性などを確認、評価し、セキュリティ対策をシミュレータにより評価する方法の有効性を検証した。

3.2d.1　評価方法の検討

車内ネットワークの通信プロトコル（以下「車内通信プロトコル」という）が備えるセキュリティ対策を評価する際に必要となる攻撃方法の検討結果を記す。以降では、まず本検討で対象とする攻撃に関する前提条件を整理する。次いで、主たる検討対象として選定した CAN に対する攻撃方法について可能な限り網羅的で体系的な整理・分類を行う。更に、それぞれの攻撃方法について既存の文献で指摘されている攻撃手順を示すと共に、シミュレーションによる安全性評価をより網羅的に実施するための攻撃手順のバリエーション拡充について検討する。最後に、CAN に対する攻撃方法の、その他の通信プロトコル（CAN-FD、LIN、FlexRay など）への適用可能性を検討し、その結果に基づく攻撃方法の拡充と分類の拡張を行う。

(1) 攻撃方法に関する前提

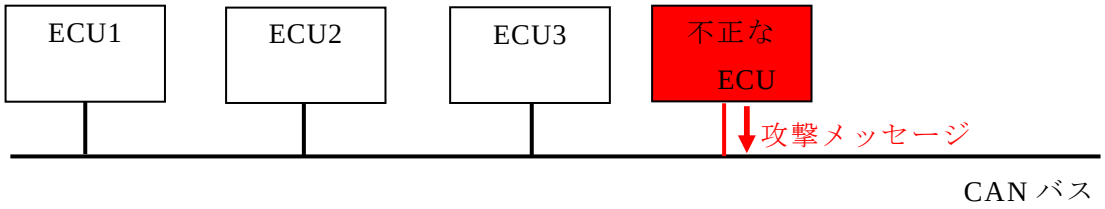
ここでは本検討で実施する攻撃方法の整理・分類、および、シミュレータ上での再現において対象とする攻撃に関する前提条件を記す。

①　攻撃者の環境

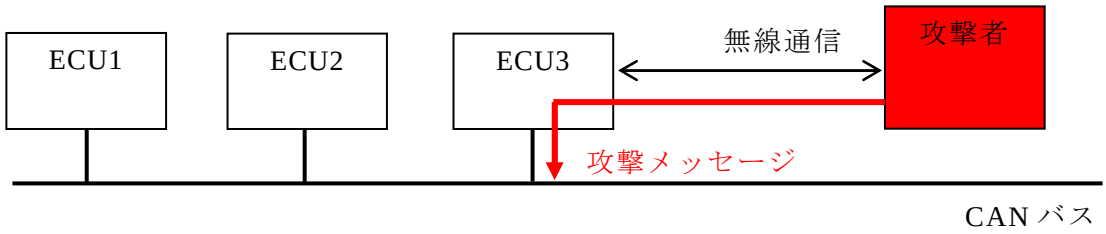
車内通信プロトコルに対して攻撃が行われる際、攻撃者の取り得る環境は図 3.2d.1-1 のように 3 通りに分けられる。1 つ目は、攻撃者が CAN バスに不正な機器を物理的に接続し、その不正な機器が他の ECU を攻撃するメッセージを送信する場合である。不正な機器の接続の仕方は OBD-II ポートを介して繋ぐ方法や、自動車の内外装を剥がし CAN バスに直接繋ぐ方法が考えられる。2 つ目は、攻撃者が外部ネットワークと通信する機能を有する正規の既設 ECU を介して CAN バスに攻撃メッセージを送信する場合である。正規の既設 ECU に脆弱性があり外部ネットワークから不正な CAN メッセージを直接送り込まれる場合や、メッセージ自体は正規の ECU が生成し送信するものではあるが外部ネットワークからの制御を許し不正なタイミングで送信させられてしまう場合を含む。3 つ目は、攻撃者が CAN バスに接続された正規の ECU に何らかの方法で不正な動作をするプログラムを書き込み、そのプログラムが他の ECU を攻撃するメッセージを送信する場合である。不正プログラムを書き込む方法は、OBD-II ポートから CAN バスに侵入して書き換えを実行する命令を送信する方法や、正規の遠隔アップデート機能の脆弱性を突いて書き換え機能を悪用する方法などがある。

本検討はつながる車における自動車に対するサイバー攻撃を主たる脅威として捉えていることから、攻撃は遠隔地から実施されることを想定しており、攻撃対象の自動車への物理的な接続や改造を伴う攻撃は想定しない。そのため、図 3.2d.1-1 の 2、および 3 の攻撃環境で実現可能な攻撃方法を検討の対象とし、1 つ目の物理的に不正な ECU を接続する必要がある攻撃方法は検討の対象外とした。具体的には以下の攻撃方法を対象外とした。

1. 攻撃者が CAN バスに不正な ECU を接続し攻撃を行う場合（検討対象外）



2. 攻撃者が正規の ECU を経由して遠隔地から攻撃する場合



3. 攻撃者が正規の ECU に不正なプログラムを書き込み攻撃する場合

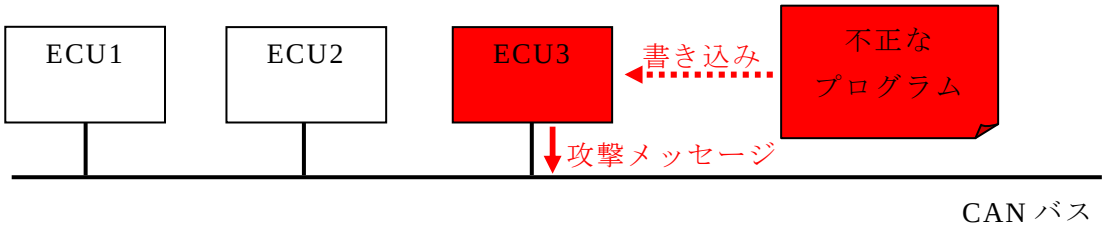


図 3.2d.1-1　攻撃者の環境

(i) エラーフレーム送信

CAN において、エラーフレームの送信は CAN トランシーバおよび CAN コントローラが制御している。任意のタイミングでエラーフレームを送信し、CAN バスを麻痺させたり、特定の ECU に誤動作を引き起こしたりする攻撃は、任意のタイミングでエラーフレームを送信できるよう改造された CAN トランシーバと CAN コントローラを接続しなければ実現できない。

ECU のアプリケーション層はファームウェアアップデートなどソフトウェアの書き換えが可能である。一方、CAN トランシーバと CAN コントローラは CAN の仕様通りに動作すればよいため、書き換えられるように設計されていることは希である。例えば、最も普及している CAN コントローラの 1 つ MCP2515[1]がアプリケーション層から受け取れる情報はデータフレームの送信に必要となる CAN-ID とデータフィールド（以下、DF）の値のみである（図 3.2d.1-2）。つまり ECU のアプリケーション層は MCP2515 に対して CAN エラーフレームの送信を命令できず、また MCP2515 に対して制御回路の書き換えを命令できない。よって、本攻撃を実施するには、攻撃者が改造を施した CAN トランシーバと CAN コントローラを内蔵した不正な ECU を CAN バスに直に物理的に接続する、上記 1 つ目の環境である必要がある。

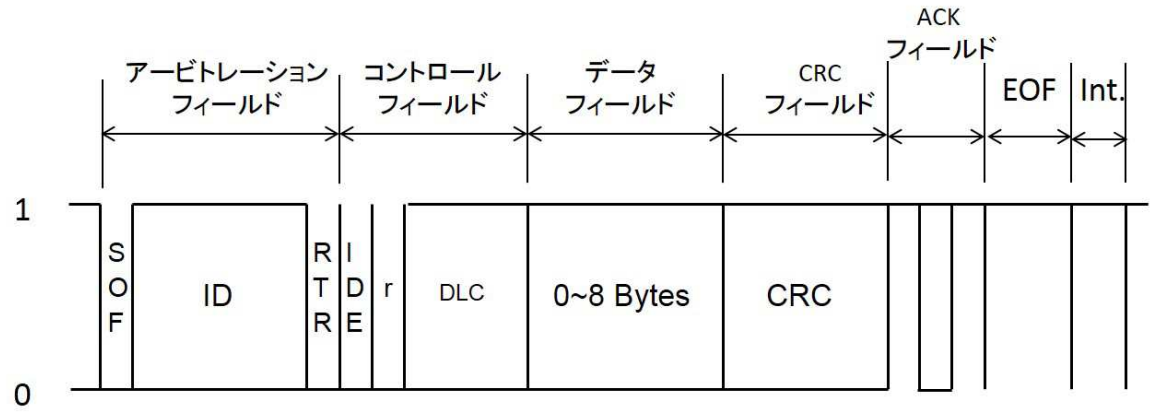


図 3.2d.1-2　CAN のデータフォーマット（11 ビットヘッダの場合）

(ⅱ) 規定外フォーマット送信

規定外フォーマットとは、例えばビット 0 とビット 1 を 1 ビットずつ交互に送信し続ける事や、CAN バスに他 ECU がメッセージを送信している最中にメッセージ送信を行うなどの CAN の仕様を無視したメッセージ送信を指す。通常 CAN トランシーバならびに CAN コントローラは前述の規定外フォーマットを送信できるようには設計されていない。よって、本攻撃を実施するには、前述のエラーフレーム送信と同様、攻撃者が改造を施した CAN トランシーバと CAN コントローラを内蔵した不正な ECU を CAN バスに直に物理的に接続する必要がある。

② プロトコル

本検討を効率良く進めるために、攻撃方法の整理・分類の主たる分析対象、かつ、シミュレータを用いた検証対象の車内通信プロトコルを選定する。現代の車内ネットワークは、複数のサブシステムとサブネットワークから構成されており、その目的や伝送データ量、伝送速度、信頼性、コストなどの観点から適切な車内通信プロトコルが選定されている。2010 年頃の典型的な車内ネットワークは、マイコンを搭載した Electronic Control Unit（以下「ECU」という）が 30 個弱搭載され車両制御系ネットワーク、ボディ系ネットワーク、情報系ネットワークの 3 系統のサブネットワークを構成し、3 系統は車内のゲートウェイ機器を介して相互に接続していた[2]。2017 年現在においては、1 台あたり 100 個以上の ECU を搭載した車両も存在し、機能拡充に伴い車内ネットワークの構成は前述の 3 系統に加えてテレマティクス系、ITS 機能系、保守診断系、安全快適機能系など複雑化、多層化している。

平成 27 年度に実施した脆弱性および攻撃方法の調査[I]で、脆弱性あるいは攻撃方法の指摘が確認された車内通信プロトコルがそれぞれどのように用いられるかを以下に述べる。

(ⅰ) CAN

サブネットワークの機能に依らず ECU 間の通信に広く用いられる。外部と無線通信できる ECU が接続された CAN を用いた車内ネットワークは、その ECU を介して車外ネットワ

ークと無線接続可能である。また故障診断用ポート On-Board Diagnostics-II（以下「OBD-II」という）が CAN に接続された車両は多く、OBD-II ポートを介して車外ネットワークと物理的な有線でも接続可能である。OBD-II ポートに取り付け可能な無線通信機器も市場で販売されており、後付けした通信機器を介して CAN が車外ネットワークと無線接続する可能性もある。

(ⅱ) CAN-FD

CAN の方式を踏襲し CAN よりも高速通信が可能であり、やはり ECU 間の通信に用いられる。ただし CAN-FD は 2015 年に標準化されたばかりであり、2017 年 3 月現在において CAN ほどには実車両に採用されていない。

(ⅲ) FlexRay

CAN と同様に ECU 間で利用することも想定された車内通信プロトコルであるが、主にブレーキやステアリングなど通信に高い信頼性が要求される制御系のサブネットワーク、いわゆる X-by-Wire において用いられている。そのため FlexRay による車内ネットワークが車外ネットワークと直接接続していることは、CAN と比較すると希である。

(ⅳ) LIN

LIN は主に ECU と制御末端のセンサ、アクチュエータ間の通信として、1 台あるいは少数台数の ECU と複数のセンサ、アクチュエータから成るサブネットワークに用いられる。LIN による車内ネットワークが車外ネットワークと接続するためには CAN などの ECU 間通信を経由する必要があり、LIN による車内ネットワークが車外ネットワークと直に接続していることは希である。

(ⅴ) MOST

主に音楽や動画といったマルチメディアを再生するインフォテイメント機器、スピーカやマイク、ディスプレイなどの通信に用いられる。LIN 同様 MOST による車内ネットワークが車外ネットワークと接続するためには、インフォテイメント機器が CAN などの ECU 間通信に接続している必要があり、MOST による車内ネットワークが車外ネットワークと直に接続していることは希である。

(ⅵ) SENT

主に ECU とセンサ間の通信に用いられる。LIN 同様 SENT による車内ネットワークが車外ネットワークと接続するためには CAN などの ECU 間通信に接続している必要があり、SENT による車内ネットワークが車外ネットワークと直に接続していることは希である。

以上のように、現代の車両の制御において最も利用されている車内通信プロトコルは CAN であり、また OBD-II 等を介して車外と接続し、脅威に晒されるサブネットで採用されているプロトコルも CAN である場合が多い。更に、表 3.2d.1-1 に示す通り、平成 27 年度に実施した脆弱性および攻撃方法の調査に依れば、文献などによる脆弱性あるいは攻撃

方法の指摘も CAN に対するものが圧倒的多数であると同時に、他の車内通信プロトコルに対する脆弱性あるいは攻撃方法の指摘は CAN にも当てはまる点が多かった。

以上の点から、本検討における攻撃方法の整理・分類の主たる分析対象、かつ、シミュレータを用いた検証対象の車内通信プロトコルとして CAN を選定した。

表 3.2d.1-1　脆弱性あるいは攻撃方法の指摘

通信プロトコル	脆弱性あるいは攻撃方法を指摘する文献等の件数
CAN	21 件 [II、IV、V、VI、VII、VIII、IX、X、XI、XII、XIII、XIV、XV、XVI、XVII、XVIII、XIX、XX、XXI、XXII、XXIII]
CAN-FD	1 件 [XXIII]
FlexRay	1 件 [II]
LIN	1 件 [II]
MOST	1 件 [II]
SENT	1 件 [III]

(2) 攻撃方法の分類

ここでは前述のように主たる検討対象として選定した CAN に対する攻撃方法を基に、可能な限り網羅的で体系的な攻撃方法の整理・分類を行う。まず、情報システムのセキュリティの三大要素である「機密性」「完全性」「可用性」に基づき大分類を定める。ついで、攻撃者が攻撃メッセージを作成・送信する際に故意に調整し得る要素である、作成するメッセージの内容とメッセージを送信する時間の二点に着目し、それらの観点から分類の細分化を図る。そして、平成 27 年度に調査した脆弱性あるいは攻撃方法の各指摘が本分類のいずれかに当てはまるかを点検することで、本分類の十分性を確認する。なお、その調査で抽出した既存の攻撃方法以外の攻撃方法の拡充、ならびに CAN 以外のプロトコルを含めて見た場合の分類の拡張および攻撃方法の拡充については、次項以降で検討する。

① 情報セキュリティの三大要素による分類

OECD の情報セキュリティガイドライン[1]での定義を規範とし、情報システムに関する多くの規程類やガイドラインにおいて、情報システムが有する情報資産について満たすべき情報セキュリティの三大要素として機密性、完全性、可用性が用いられている。車内ネットワークにおいて守るべき情報資産とは、送受信されるメッセージ、ならびに接続している ECU 等の機器である。これら情報資産に対しても前記三大要素の維持が重要であり、車内ネットワークに対する攻撃方法の基本分類として適切であろうと考える。

ISMS ファミリの用語･定義を記述した ISO/IEC 27000 ではそれぞれ次のように定めている。まず機密性とは「認可されていない個人、エンティティ又はプロセス対して、情報を使用させず、また、開示しない特性」と定義されている。CAN においてエンティティとは

主に CAN に接続する ECU を指し、機密性が維持されるとは、認可されていない攻撃者が CAN メッセージを参照できない、あるいは参照できたとしてもそれを使用できない（意味を理解できない）性質と言える。次に完全性は「正確さ及び完全さの特性」と定義されている。CAN では、CAN メッセージの欠落や重複、改竄、なりすまし、破壊などが起こらないよう保護する性質と言える。そして可用性は「認可されたエンティティが要求したときに、アクセス及び使用が可能である特性」と定義されている。CAN では CAN メッセージが必要な時に必要な ECU によって確実に送信・受信され、各 ECU が定められた処理を遅延あるいは停止せずに実行する性質と言える。

以下では、それぞれの要件に対応する脅威について説明する。

・盗聴

CAN メッセージが盗み見られたり奪取されたりする脅威である。前述のとおり CAN は全てのメッセージがブロードキャストで送信される。そして通信路上のメッセージを秘匿するための機能、例えば暗号化などは規定されていない。そのため、ひとたび攻撃者が CAN バスに接続できたなら、攻撃者はそこに流れる全てのメッセージを受信し取得できる。つまり CAN ではアプリケーション層でペイロードの暗号化など機密性を保持する機能が実装されない限り、攻撃者に接続された瞬間に直ちに機密性が満たされなくなる。

このように機密性が一切保たれない状況は、CAN 以外の車内通信プロトコル（CAN-FD、FlexRay、LIN など）でも同様であり、この観点からの評価は有意ではない。よって機密性に対する脅威としての盗聴は、本検討で対象とする攻撃方法からは除外することとした。

・なりすまし

ECU が本来受信すべきメッセージと異なる内容あるいはタイミングのメッセージを受信させられることにより、車内システムに予期せぬ制御や表示が引き起こされる脅威である。CAN はマルチマスタ方式であるため攻撃者は通信路に流れたメッセージを受信して記憶し、任意のタイミングで再送すればメッセージの重複が発生する。また受信して記憶したメッセージのペイロードを一部変更して再送すればなりすましメッセージを送信できる。ただし、攻撃が遠隔地から実施される前提の下ではメッセージの欠落と破壊は困難である。なお、CAN においてデータフレーム内に規定される CRC フィールドの機能はノイズによるビット反転などに対する誤り訂正符号であり、情報セキュリティ上の完全性を維持するための機能は有しない。

・DoS

一部の機能あるいは車輌全体の機能が阻害される脅威である。CAN はイベントトリガ方式であるため、攻撃者は連続して過剰にメッセージを送信することが可能である。過剰な送信によって CAN バスの帯域が埋め尽くされることにより、正常な通信が遅延するかあるいは完全に停止する。これにより正規の送信 ECU が送信したいメッセージが送信されず、また正規の受信 ECU が受信したいメッセージを受信できない被害を受ける。また、CAN バスの帯域を占有しない場合でも、受信 ECU が処理しきれない頻度でメッセージを送信することで受信 ECU の機能が遅延あるいは停止する被害を受けることが想定される。

② 攻撃メッセージの調整内容による分類の細分化

なりすまし攻撃と DoS 攻撃は通信路に何らかのメッセージを送信する能動的な攻撃である。CAN において、なりすまし攻撃および DoS 攻撃のための攻撃メッセージを作成・送信する際、攻撃者が故意に調整し得る要素としては、作成するメッセージの内容とメッセージを送信するタイミングの二点がある。一点目のメッセージ内容については、前述の通り、MCP2515[2]など通常の CAN コントローラがアプリケーション層から受け取れる情報は CAN-ID と DF の値のみである。ただし、攻撃者は意図した送信データ長（以下、DLC）の値となるようなデータを設定可能であること、CAN-ID は攻撃対象の ECU に応じて決定する必要があることから、DLC と DF の二点が調整可能な箇所と考えられる。また、現在の自動車では故障診断モードが実装されており、そこで用いられる CAN メッセージは ISO15765-2 で標準化されている。故障診断モードに移行した場合、ECU の機能は大きく制限されることから、診断用メッセージは他の通常の CAN メッセージとは分けて考えるのが妥当である。二点目のメッセージの送信のタイミングについては、周期・頻度を調整することが可能である。

完全性（なりすまし）、可用性（DoS）の大分類に対し、攻撃者が調整し得るメッセージの内容および送信タイミングの観点からの区分を適用し、有り得る組合せを整理した結果を表 3.2d.1-2 の左側、「攻撃方法」の列に示す。

同表の右側、「指摘している文献」の列に、平成 27 年度の調査で抽出された既存の CAN に対する攻撃方法のうち本検討の対象内のもの、すなわち、攻撃者による不正な機器の物理的な接続を要せず遠隔から実施可能な攻撃方法をマップした。空欄が存在しないことから、表 3.2d.1-2 の攻撃方法の分類は漏れの無い分類となっていることが確認できる。

表 3.2d.1-2　CAN における攻撃方法の分類と指摘している文献

攻撃方法			指摘している文献
なりすまし	リプレイ送信		[IV、V、VI、VII、IX、X、XI、XIII、XIV、XVI、XVII、XVIII、XX、XXII]
	不正メッセージ送信	不正 DLC	[XIV、XVII]
		不正 DF	[IV、V、VI、VII、VIII、IX、X、XI、XII、XIII、XIV、XVI、XVII、XX、XXII]
DoS	高頻度送信	通信路全体	[II、V、VI、VIII、IX、X、XI]
		特定 ECU	[IV、XV]
	正規メッセージとの衝突メッセージ送信		[XXI、XXIII]
	通常メッセージ以外のメッセージ送信	診断用メッセージ送信	[V、XXI、XXII]
	不正メッセージ送信	不正 DLC	[XXIII]
		不正 DF	[XXIII]

(3) 攻撃手順の概要

先に述べた攻撃方法それぞれについて、CAN における攻撃の手順の概要を示す。各攻撃方法について、平成 27 年度に調査した各既存の文献が指摘する攻撃手順を挙げたのち、その個々の文献の攻撃手順のみシミュレートすれば安全性評価として十分か、指摘以外のバリエーションを拡充する余地や必要性がないかについて検討し、それらの結果についても論じる。なお各文献の詳細な分析は[I]を参照されたい。

① なりすまし

(ⅰ) リプレイ送信

攻撃者は CAN バスを盗聴するなどして得た正規のメッセージを記憶する。そして記憶したとおりの CAN-ID と DLC、DF、送信周期で再送する攻撃である。各文献では記憶したメッセージが送信され得ないタイミングなどにただ再送すれば攻撃は実現できるとされているが、評価対象の被害状況を確認し、安全性を正しく評価する上で、以下のバリエーションを追加考慮すべきである。

a) 乗っ取った ECU と送信する CAN-ID の組み合わせによる違い

各文献は暗黙のうちに、攻撃者が乗っ取った ECU は他の ECU が送信する CAN-ID になりすますことを前提としている。しかし、乗っ取った ECU が元より送信する CAN-ID を攻撃者が利用する状態も考えられる。図 3.2d.1-3 を用いてそれぞれの状態における ECU の挙動を説明する。まず、ECU A は正常な送信 ECU で CAN-ID A のメッセージを送信する。次に、ECU B は攻撃者が乗っ取った ECU で、攻撃者に乗っ取られる前は CAN-ID B を送信していた。最後に、ECU C は正常な受信 ECU で CAN-ID A と CAN-ID B のメッセージを受信し何らかの制御を行う。①の状態は文献が暗黙のうちに前提としているものを表している。具体的には ECU B が ECU A になりすまそうとしている。他方、②の状態では攻撃 ECU である ECU B は、ECU B が元より送信するメッセージを攻撃として利用している。①と②では受信 ECU に引き起こされる被害は異なり得る(図 3.2d.1-4 を参照)。例えば、①では攻撃者 ECU B は CAN-ID A のメッセージをリプレイしているが、正常な CAN-ID A のメッセージを消すことはできない。一方、②では ECU B は CAN-ID B のメッセージを本来とは異なるタイミングで送信するだけでなく、例えば停止させるなどして、本来のタイミングでは送信しないように変更できる。

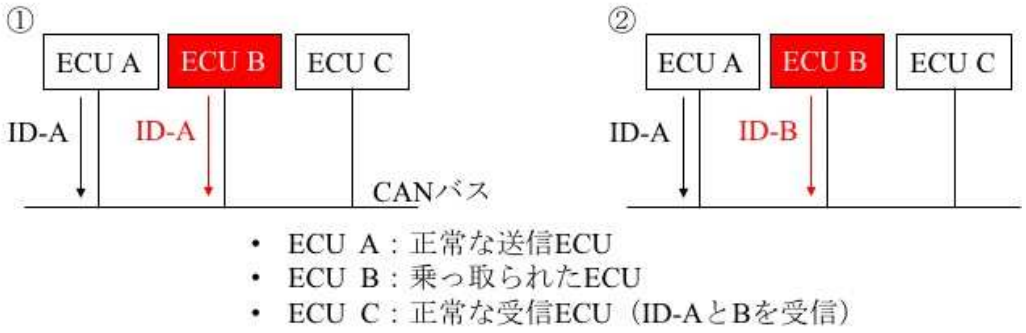


図 3.2d.1-3　乗っ取った ECU と送信する CAN-ID の組み合わせ

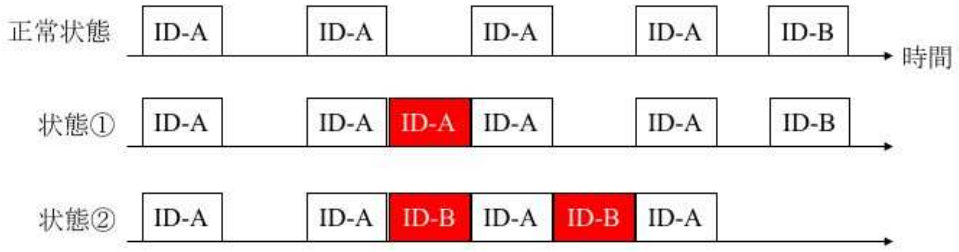


図 3.2d.1-4　CAN バスの時間変化、状態の違いによるリプレイ送信の違い

b) 被攻撃メッセージの周期的送信・イベント送信による違い

CAN で流れるメッセージは CAN-ID 毎に周期・頻度の性質で 2 通りに分けられる。何らかの周期性を有しほぼ定常的にバスに送信される周期的メッセージと、定常的には流れず、なんらかのイベントに反応して単発的、あるいは短い時間だけ周期的に送信されるイベントメッセージである。攻撃者がどちらのタイプのメッセージを記憶し再送するのかによって引き起こされる被害は異なり得る。

c) 攻撃メッセージの送信タイミング・周期による違い

b)にあげたイベントメッセージの場合に、正規のイベントメッセージが流れているほぼ同じタイミングに再送するのか、全く異なるタイミングに再送するのかによって引き起こされる被害は異なり得る。周期的メッセージの場合には、正規の周期的メッセージと同じ周期で再送するのか、周期を変更して再送するのかによって引き起こされる被害は異なり得る。

(ⅱ) 不正メッセージ送信

攻撃者がなりすましたい機能に関わるメッセージ（CAN-ID）において、CAN メッセージのデータフレームの規定に従い攻撃者が変更可能な DLC、DF の 1 つないし両方を不正な値に変更したメッセージを送信する攻撃である。変更を施す箇所により、更に以下の二種に区分される。

a) 不正 DLC メッセージ送信

DLC は DF の長さを指定するパラメータであり、0 バイトから 8 バイトまで指定できる。ただし 3.2d.1.(2)で述べたとおり、本検討が想定する攻撃者が任意に指定できるのは CAN-ID と DF の値のみであるため、攻撃者は DLC が攻撃者の狙った値になるように DF の値を設定する。例えば、DF に 0x00 00 を指定すれば DLC は 2 となり、DF に 0x00 を指定すれば DLC は 1 となる。不正 DLC メッセージの例を以下に示す。攻撃者が盗聴して得たメッセージの DLC が 7 であったとき、攻撃者が DF を任意のバイト数を切り詰めることで DLC を 0〜6 に変更した不正なメッセージを送信するか、あるいは DF の 8 バイト目に不正な値を追加して DLC を 8 に変更したメッセージを送信する。本来の DLC より短くしたメッセージを送信することで、受信者は処理すべきデータの一部が欠損したメッセージを受信し、機能が正常に完了しないおそれがある。同様に本来の DLC より長いメッセージを送信することで、受信者はオーバフロー等を引き起こし不正な機能を実行するおそれがある。

b) 不正 DF メッセージ送信

DF の一部または全てを変更して送信する攻撃である。例えば車速や操舵角といった制御に関わるデータを変更したメッセージを送信することで受信者に不正な制御を実行させる。本来取り得る値域内の異なるデータに変更する場合や、値域外に変更する場合が考えられる。例えば、120km/h で走行中に 0km/h や 900km/h という不正な DF メッセージを送信する攻撃である。

a)、b)双方の場合においてリプレイ送信で述べた以下のバリエーションが同様に当てはまる。攻撃に対する安全性を正しく評価する上で、これらのバリエーションを追加考慮すべきである。

1) 乗っ取った ECU と送信する CAN-ID の組み合わせによる違い
2) 被攻撃メッセージの周期的送信・イベント送信による違い
3) 攻撃メッセージの送信タイミング・周期による違い

② DoS

(ⅰ) 高頻度送信

攻撃者はなんらかの CAN メッセージを高頻度に送信することで一部の ECU あるいはバス全体の機能を阻害する攻撃である。各文献の指摘のとおり機能を阻害する方法は 2 通りに分類できる。

a) 通信路全体に対する高頻度送信

1 つ目は送信側を阻害する方法である。具体的には、攻撃者は高頻度なメッセージ送信によって通信路を占有し正規のメッセージが送信できなくする。文献でも通信路全体の機能を阻害する攻撃方法は CAN バスの帯域を埋め尽くすことで実現できると指摘されている。

しかし、具体的に何個の ECU を乗っ取ればよいか、どういった CAN-ID や DL、DF の値を用いてどれだけの間隔で攻撃メッセージを送信すれば帯域を埋め尽くせるかといった事は具体的に示されていない。評価対象の安全性を正しく評価するためには、それらのパラメータを振って、それぞれの条件における被害状況を確認すべきである。

また、攻撃メッセージに用いる CAN-ID の値によって攻撃の被害が通信路全体に及ぶのかあるいは一部の機能に及ぶのかが異なる。なぜならば CAN では前述のとおり同時送信メッセージ間の調停機能が定められており、CAN-ID の値が小さいほど送信が優先される。つまり攻撃者が高頻度に送信するメッセージの CAN-ID よりも番号が小さい CAN-ID を有する正規のメッセージは、攻撃メッセージが通信路を占有していようとも攻撃メッセージとの調停に打ち勝ち送信が可能である。評価対象の安全性を正しく評価するためには、それらのバリエーションについて追加考慮すべきである。

b) 特定 ECU に対する高頻度送信

高頻度送信の 2 つ目は受信側を阻害する方法である。具体的には、攻撃者は受信者の処理能力を上回る量のメッセージを送信し、これを受信させて受信者の機能の実行を遅延あるいは停止させる。最も受信者に負荷を掛けられる方法は、メッセージを最短頻度で送ることであり、前述の通信路を占有する高頻度送信と同じ攻撃手順になる。通信路を占有するメッセージ群に対して受信者が遅延あるいは停止せずに処理を続ける事ができる場合、本攻撃に対する耐性を有すると言える。しかし通信路を占有するほどではないが高送信頻度の攻撃メッセージ群によって受信者が遅延あるいは停止した場合は、本攻撃に対して脆弱だと言える。評価対象の安全性を正しく評価するためには、送信頻度のパラメータを振って攻撃してそれぞれの場合の被害状況を確認し、どの程度の受信頻度にまで耐えられるのかを確認すべきである。

リプレイ送信で述べたバリエーションのうち、以下に示す 2 つが a、b 双方の場合において同様に当てはまる。攻撃に対する安全性を正しく評価する上で、これらのバリエーションを追加考慮すべきである。

1) 乗っ取った ECU と送信する CAN-ID の組み合わせによる違い

通信路全体に対する高頻度送信は、攻撃メッセージの CAN-ID によって 3 通りに大別できる。まず、①車両内で実際に使われている CAN-ID の中で最も小さい番号の CAN-ID を攻撃メッセージとして用いる場合である。次に、②乗っ取った ECU 自身が元より送信する CAN-ID の中で最も小さい番号の CAN-ID を用いる場合である。最後に、③車両で実際に使われている最も小さい番号の CAN-ID よりもさらに小さい CAN-ID を用いる場合である。

攻撃に用いる CAN-ID の大小によっては調停機能により攻撃メッセージがバスを占有できないことがあるが、その際は DLC や DF の値を優先されるように小さい値に設定することでバスを占有できるようになる場合がある。

一方、特定 ECU に対する高頻度送信では、攻撃メッセージの CAN-ID によって 2 通りに大別できる。まず、①特定 ECU が受信して処理する CAN-ID を用いる場合と、②特定

ECU が受信して処理する CAN-ID よりも小さい番号の CAN-ID を用いる場合である。①の場合、特定 ECU は攻撃メッセージを受信し何らかの処理を行う。攻撃メッセージの送信頻度が設計仕様上の上限を上回る場合、不正な処理あるいは処理の遅滞や停止が起こり得る。②の場合、攻撃メッセージを高頻度に送信することで特定 ECU が受信するメッセージに調停で打ち勝ち、特定 ECU が受信するメッセージを遅滞あるいは停止させることが起こり得る

2) 攻撃メッセージの送信タイミング・周期による違い

通信路全体に対する DoS か特定 ECU に対する DoS かによって攻撃メッセージの周期に違いがある。まず、CAN の通信路全体に対する高頻度送信で他のメッセージの送信を停止させるためには、連続する攻撃メッセージ間の隙間 ITM が CAN の仕様で定められている最短の 3 ビットである必要がある。ITM を 4 ビット以上空けてしまうと、送信待ちをしている他の ECU が送信したメッセージが通信路に送信されてしまうため、攻撃メッセージで通信路を占有できない。他方、特定の ECU に対する高頻度送信では、必ずしも ITM を 3 ビットで送信し続ける必要はない。特定 ECU の処理能力が低い場合は、ある程度高頻度の送信であれば遅滞あるいは停止が起こり得る。

なお、リプレイ送信で述べたうちの残りの一つである被攻撃メッセージの周期的送信・イベント送信による違いから発生するバリエーションもあり得るものの、攻撃手順や攻撃した結果に区別すべき差が発生するとは考えにくいため、詳しい検討は省略する。

(ⅱ) 正規メッセージとの衝突メッセージ送信（バスオフ誘発）

攻撃者は正規メッセージと同タイミングで攻撃メッセージを送信することでメッセージの衝突を引き起こし、CAN のエラーハンドリング機能を発動させて正規メッセージの送信遅延を繰り返し、正規メッセージを送信する ECU をバスオフ状態に遷移させて一時的に停止させる攻撃である。

リプレイ送信で述べたバリエーションのうち、以下に示す 2 つが、本攻撃の場合においても同様に当てはまる。攻撃に対する安全性を正しく評価する上で、これらのバリエーションを追加考慮すべきである。

1) 攻撃メッセージの送信タイミング・周期による違い

文献では攻撃メッセージの送信周期を衝突させたい正規メッセージの送信周期と同調させる必要があるとしているが、ECU をバスオフ状態に陥らせるためには同調させて送信しなくてもよい。正規メッセージの送信周期を考慮せず、攻撃メッセージを高頻度に送信することで同様にバスオフ状態に移行させられる場合がある。

2) 被攻撃メッセージの周期的送信・イベント送信による違い

前述の攻撃メッセージの送信周期を衝突させたい正規メッセージの送信周期と同調させる攻撃手順の場合、攻撃対象の正規メッセージがイベント型送信だとそもそも同調することが難しい。一方、高頻度に送信する攻撃手順であれば、イベント型送信であった

としてもバスオフ状態に移行させられる場合がある。

なお、リプレイ送信で述べたうちの残りの一つである乗っ取った ECU と送信する CAN-ID の組み合わせの違いによるバリエーションもあり得るものの、攻撃手順や攻撃した結果に区別すべき差が発生するとは考えにくいため、詳しい検討は省略する。

(ⅲ) 通常メッセージ以外のメッセージ送信

a) 診断用メッセージフレーム送信

現在の自動車では ECU に故障診断モードという機能が実装されていることが多い。OBD-II ポートに接続した診断機器と ECU が故障診断用の CAN メッセージを送受信することで、テスタは各 ECU の機能を診断する。故障診断モードに移行すると診断テスタからのテストメッセージに対してのみ反応をする場合が多い。そのため不正なタイミングで故障診断モードへの移行を命令するメッセージを送信すると、特定の ECU あるいは特定の機能を有する複数の ECU の機能が停止する。ECU の診断モードで行う処理の内容は CAN の仕様では規定されていないが、診断に用いる CAN 通信の形式は ISO15765-2 で標準化されている。

リプレイ送信で述べたバリエーションのうち、以下に示す二つは、本攻撃の場合においてもあり得るものの、攻撃手順や攻撃した結果に区別すべき差が発生するとは考えにくいため、詳しい検討は省略する。

1) 攻撃メッセージの送信タイミング・周期による違い
2) 攻撃メッセージの単一 CAN-ID 送信・複数 CAN-ID 送信による違い

また、診断用メッセージはイベント送信であるため、リプレイ送信で述べたうちの残りの一つである以下のバリエーションは本攻撃の場合において想定できない。

3) 被攻撃メッセージの周期的送信・イベント送信による違い

(ⅳ) 不正メッセージ送信

a) 不正 DLC メッセージ送信

攻撃者は正規のメッセージと同時に不正なメッセージを送信し、DLC でメッセージの衝突を引き起こすことにより、正規メッセージの送信中止とエラーフレームの送信、その後の正規メッセージの再送を引き起こし、正規のメッセージの送信完了を遅延させる攻撃である。本攻撃による衝突を連続して繰り返す事で正規のメッセージを送信した ECU はエラーカウンタが増大し、エラーカウンタが上限に到達するとバスオフ状態に遷移する。すなわち前述の(ⅱ)正規メッセージとの衝突メッセージ送信（バスオフ誘発）を引き起こせる。すなわち、本攻撃はバスオフ誘発の過程で発生する事象として観測され得るものである。本攻撃の詳細は(ⅱ)を参照されたい。

b) 不正 DF メッセージ送信

攻撃者は正規のメッセージと同時に不正なメッセージを送信し、DF でメッセージの衝突を引き起こすことにより、正規メッセージの送信中止とエラーフレームの送信、そ

の後の正規メッセージの再送を引き起こし、正規のメッセージの送信完了を遅延させる攻撃である。本攻撃による衝突を連続して繰り返す事で不正 DLC メッセージ送信の場合と同様に、前述の(ⅱ)正規メッセージとの衝突メッセージ送信（バスオフ誘発）を引き起こせる。すなわち、本攻撃はバスオフ誘発の過程で発生する事象として観測され得るものである。本攻撃の詳細は(ⅱ)を参照されたい。

(4) 攻撃方法の他通信プロトコルへの適用および拡張可能性の検討

自動走行車には、これまでフォーカスをあてて来た CAN 以外にも、CAN-FD、FlexRay、LIN といった車内プロトコルが各々の特徴に応じた領域で利用されると考えられる。よって、それらに対して、攻撃方法を明確にすることが重要であると考え、これまでに述べた CAN に対する攻撃方法を参考に、CAN-FD、FlexRay、LIN 通信上の攻撃方法について検討し、机上で検証した結果を述べる。ここでは、攻撃方法を網羅的に検討するため、CAN で検討の範囲外とした攻撃方法も含めて検討を行った。本検討結果は、車内プロトコルへの攻撃方法の整理・分類における網羅性を向上させると共に、各プロトコルの方式や構成といった特徴から、各通信プロトコル上において攻撃が可能であるかを検証することに利用することができると考える。

表 3.2d.1-3 に各通信プロトコルの概要を記載する[2]。

表 3.2d.1-3　各通信プロトコルの概要

プロトコル名	CAN	CAN-FD	FlexRay	LIN
カテゴリ	イベントトリガ	イベントトリガ	タイムトリガ	サブバス
構成	マルチマスタ	マルチマスタ	マルチマスタ	シングルマスタ
ネットワークトポロジ	バス型	バス型	バス型 スター型 バス・スター混合型	バス型
アクセスコントロール	CSMA/CA	CSMA/CA	TDMA FTDMA	ポーリング
データレート	1 Mbit/s	8Mbit/s	10 Mbit/s	20 kBit/s
エラー検知	CRC パリティビット	CRC パリティビット	CRC バス・ガーディアン	チェックサム パリティビット
主なアプリケーション	パワートレイン（エンジン、トランスミッション等）	パワートレイン（エンジン、トランスミッション等）	高性能パワートレイン、安全機能（ドライブバイワイヤ、クルーズコントロール等）	ミラー、電動シート、ステアリング、各種アクセサリ等

① CAN-FD [9, 10]

(ⅰ) CAN-FD の概要

CAN-FD は CAN のプロトコルを拡張して、より多くのデータを高速に転送するための方式である[9, 10]。CAN-FD は 1 メッセージあたりのペイロードのサイズを最大 512 ビットに拡張し、通信速度は可変ビットレート方式を導入している。図 3.2d.1-5 に CAN-FD の基本データフォーマット（11 ビットのヘッダの場合）を示す。ここでは ISO11898[10]で規定されている領域の名称を利用する。図 3.2d.1-2 に示した CAN のデータフォーマットと比べ CAN-FD のコントロールフィールドでは FDF ビット、BRS ビット、ESI ビットの 3 つのビットが新たに追加されている。各々の意味は以下の通りである。

- FDF : CAN と CAN-FD のデータフォーマットを区別するためのビット。値はリセッシブ(1)を示す。
- BRS : ビットレートを変更するか決めるビット。値がリセッシブ(1)の場合、送信ノードが BRS のサンプリングポイントで、通常のビットレートから、あらかじめ決められた高速なクロック速度のクロックモードに切り替える。ドミナント(0)の場合、ビットレートの切り替えは行わない。
- ESI : 送信ノードのエラーの状態を示すビット。エラーパッシブの場合はリセッシブ(1)になり、エラーアクティブの場合はドミナント(0)となる。

他のフィールドの詳細内容に関しては仕様[10]を参照のこと。

CAN-FD は、CAN の拡張方式であるため、CAN の攻撃方法を CAN-FD に適用可能な場合が多いと考える。以下に CAN-FD 通信上における攻撃方法を示す。なお、ここでは網羅的に攻撃方法を検討するため、CAN 通信上における攻撃方法で検討の範囲外としたもの含め、検討を行った結果を記載する。

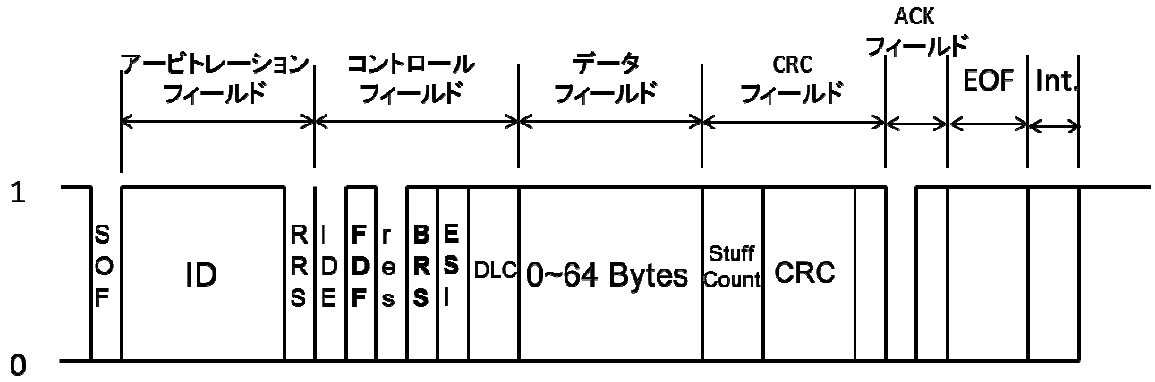


図 3.2d.1-5　CAN-FD の基本データフォーマット（11 ビットのヘッダの場合）

(ⅱ) CAN-FD 通信上での DoS

以下に、CAN 通信上の DoS 攻撃を CAN-FD 通信に適用・拡張した場合の攻撃方法に関して述べる。

a) バスへの高頻度送信により、正規レスポンス送信を抑制

・任意のノードに影響を与える：

CAN はバス上にメッセージが送信されている場合は、他ノードの送信を抑制する。よって、攻撃者が不正メッセージでバス上を占有することが可能であった。CAN-FD の場合も、CAN と同様、バス上にフレームが流れている場合は他ノードの送信が抑制されるため、同様の方法で、攻撃が可能である。

ここで CAN と CAN-FD における攻撃の容易さを比較する。両者のビットレートが同等の場合、CAN-FD は CAN に比べてデータ長を長く設定できるため、データ長が長い場合は 1 つの不正フレームでバスを占有する時間が CAN よりも長くなる。よって不正フレームの送信周期が CAN よりも長い場合でも、CAN に対する攻撃と同等の効果を得ることができる可能性がある。

・特定ノードに影響を与える：

CAN では、バス上に流れるメッセージ ID の優先度により、特定ノードの動作に影響を及ぼすことが可能であった。CAN-FD の場合も同様に、フレーム ID の優先度により特定ノードに影響を及ぼすことが可能である。

・他ノードに影響を与える：

CAN では、バス上に流れるメッセージの ID の優先度により、他ノードの動作に影響を及ぼすことが可能であった。CAN-FD の場合も同様に、フレーム ID の優先度により他ノードに影響を及ぼすことが可能である。

b) 正規フレームとの衝突により、正規ノードをバスから論理的に切り離し、正規レスポンス送信を抑制

CAN では、正規メッセージと不正メッセージを意図的に衝突させることにより、正規ノードをバスから切り離し、正規レスポンスを抑制することが可能であった。

CAN-FD も、CAN と同様のアービトレーション方式およびエラーハンドリング処理を採用しているため同等の攻撃が可能である。更に、CAN-FD の場合は概要で記載した通り、コントロールフィールドにおいて 3 つの新たなビットが追加されている。よって、CAN に比べて衝突を誘発できるビットポイントが増えており、これらのビットに衝突を繰り返し起こし、エラーカウントを増大させることにより、正規ノードのバスオフを起こすことが可能である[7]。

c) 正規レスポンス以外のフレーム送信により、正規レスポンス送信を抑制

CAN では診断用メッセージを送信することにより、特定ノードの正規メッセージを抑制することが可能であった。CAN-FD の場合も、CAN と同様の診断用メッセージの利用により、ノードを診断モードに移行させ、メッセージの送信を抑制することが可能

である。

d) ヘッダの変更により、正規フレームの送信を抑制

ここで、ヘッダ（SOF、アービトレーションフィールド、コントロールフィールド）のうち、制御に関係すると想定されるアービトレーションフィールドおよびコントロールフィールドの変更による DoS 攻撃に関して述べる。CAN における攻撃方法の拡張として、正規・不正フレームをタイミング良く衝突させること等により、各フィールドの値を変更することによって攻撃を行う方法が可能であるかを理論的に検討した結果を示す。

・アービトレーションフィールドの変更

－ID の変更

攻撃対象の ID を含むフレームが送信されるのと同時に、攻撃者が不正フレームを送信することにより、各ノードの受信・解釈用 ID に割り当てられていない ID に変更することが可能であれば、該当フレームによる挙動の誘発を抑制することが可能である。

－RRS（予約ビット）の変更

バス上では電気的にドミナントが優先されるため、衝突により RRS ビット（ドミナント(0)）をリセッシブ(1)に変更させることは難しい。よって RRS の変更による DoS 攻撃は難しい。

・コントロールフィールドの変更

コントロールフィールドのうち、DLC を変更することにより、上記と同様にビットエラーを誘発させ、該当フレームのビットエラーを誘発した領域以降の発信を抑制することが可能である。

コントロールフィールドのうち、IDE ビット、res ビットは、上記記載の RRS ビットにおける理由と同様に、衝突による値の変更および DoS 攻撃を誘発することは難しい。同様に、BRS、ESI を衝突により 0 から 1 に変更することは難しいため、以下では BRS、ESI が 1 の場合を検討する。

FDF、BRS（=“1”の時）、ESI（=“1”の時）に関して、これらを衝突により変更することによって、ビットエラー（送信ノードが発するデータとバス上のデータが異なる時に送信ノードが検知するエラー）を誘発することができれば、エラーフレームが送信ノードから発信されるため、該当フレームのビットエラーを誘発した領域以降の発信を抑制することが可能である。

e) データフィールドの変更により、正規フレームの送信を抑制

CAN では、正規・不正メッセージをタイミング良く送信し、衝突させることにより正規メッセージの発信を抑制することが可能であった。

CAN-FD でも同様に、ビットエラーにより、エラーメッセージが送信ノードから発信されるため、該当フレームの発信を抑制することが可能である。

f) CRC フィールドの変更により、正規フレームの送信を抑制

CRC フィールドのうち、制御に関係すると想定される CRC 領域の変更に関して述べる。

上記 e)と同様に、CRC の値を衝突等により変更することにより、フレーム内に含まれる CRC 値と受信ノードで計算する CRC 値が異なるため、受信ノードで該当フレームを破棄させることにより、該当フレームの受信を抑制することが可能である。

(ⅲ) CAN-FD 通信上でのなりすまし

以下に、CAN 通信上のなりすまし攻撃を CAN-FD 通信に適用した場合の攻撃方法に関して検討した結果を述べる。

a) フレームのリプレイ攻撃

CAN では、攻撃者がバス上のメッセージを傍受した後、同等のメッセージを送信することによりリプレイ攻撃が可能であった。CAN-FD 通信においても、フレームは暗号化されていないため傍受は可能である。また CAN-FD においても ID の値でメッセージ送信の優先順位が決まるため、リプレイ攻撃を行うフレームの ID の値によっては優先的にフレームがバス上に送信されるため、CAN と同様の方法でリプレイ攻撃が可能である。

b) ヘッダの変更による不正フレームの送信と不正挙動の誘発

DoS 攻撃と同様に、ヘッダのうち制御に関係すると想定されるアービトレーションフィールドおよびコントロールフィールドの変更によるなりすまし攻撃に関して述べる。また、ここでは正規・不正フレームとの衝突により、ヘッダを変更する攻撃方法を述べる。

・アービトレーションフィールドの変更

－ID の変更

衝突等を使い ID を変更することにより、あたかも正規送信ノードになりすましてフレームを送信し、受信ノードに正しいフレームとして解釈させることは可能である。ただし ID を変更することにより CRC 値が本来の値から変わるため、受信ノードに正しくフレームを受信させるためには、CRC 値の変更も行う必要がある。

－RRS の変更

DoS 攻撃で述べた通り、RRS ビットを変更させることは難しい。よって、RRS の変更によるなりすまし攻撃は難しい。

・コントロールフィールドの変更

DoS 攻撃と同様に、コントロールフィールドのうち、IDE ビット、res ビットは、衝突による値の変更となりすまし攻撃の実施は難しい。同様に、BRS、ESI を衝突により“0”から“1”に変更することは難しいため、以下では正規フレーム（攻撃対象のフレーム）の BRS および ESI がリセッシブ(1)の場合を検討する。

－FDF の変更

FDF を衝突等によりドミナント（0）に変更することにより、送信ノードはビット

エラーを検知する。この時、送信ノードがエラーパッシブの場合、エラーフラグ（6ビットの連続したリセッシブ）を送信する。よって、エラーフラグはリセッシブであることから、攻撃者が送信する不正フレームには影響を及ぼさないため、不正フレームをバス上に送信することが可能であり、受信ノードにあたかも正しいものとして不正フレームを受信させ、不正挙動を誘発することは可能である。

送信ノードがエラーアクティブの場合、エラーフラグ（6 ビットの連続したドミナント）を送信する。この時、不正フレームの res 以降の 6 ビットにリセッシブを含む場合、エラーフラグが優先されるため不正フレームもビットエラーを起こしてしまい、不正フレームをバス上に送信・受信者に受信させることは難しい。不正フレームの res 以降の 6 ビットにリセッシブを含まない場合、不正フレームが優先されるため、受信ノードにあたかも正しいものとして不正フレームを受信させ、不正挙動を誘発することは可能である。

－BRS（=“1”の時）の変更

BRS を“0”に変更することにより、ビットエラーを誘発する。この時、送信ノードの状態がエラーパッシブの場合は、FDF の変更と同様で攻撃可能である。エラーアクティブの場合は、6 ビットのエラーフレームで ESI（1 ビット）と DLC（4 ビット）が“0”に上書きされるが、DLC が 0 という場合はほとんど取り得ないため、高い確率で不正フレームもビットエラーを発生し、なりすまし攻撃は難しい。

－ESI（=“1”の時）の変更

ESI を“0”に変更することにより、ビットエラーを誘発する。この時、ESI=1 であるため送信ノードはエラーパッシブである。よって、送信ノードは 6 ビットすべてがリセッシブのエラーフラグを送信する。従って、不正フレームには影響を及ぼさないため、受信ノードにあたかも正しいものとして不正フレームを受信させ、不正挙動を誘発することは可能である。

－DLC の変更

DLC の値が実際のデータの長さが異なる不正フレームを正規フレームとは独立に送信した場合は、受信ノードで該当フレームを破棄することが行われるため、なりすまし攻撃を行うことは難しい。

更に、正規・不正フレームの衝突により DLC を変更する場合を検討した。この時、DLC の変更により送信ノードはビットエラーを誘発する。この時、送信ノードがエラーパッシブの場合は、上述の通り、受信ノードにあたかも正しいものとして不正フレームを受信させ、不正挙動を誘発することは可能である。送信ノードがエラーアクティブの場合は、6 ビットドミナントのエラーフレームが送信ノードから送信される。そのため、不正フレーム内でビットエラーが発生した場合は、なりすまし攻撃は難しい。

c) データ変更による不正フレーム送信と不正挙動の誘発

CAN では任意のタイミングで攻撃者が正規メッセージとは独立になりすましメッセージを送信することにより、不正挙動の誘発が可能であった。

CAN-FD の場合も、同様の方法でなりすましフレームを送信することにより、不正挙

動を誘発することは可能である。この時、CRC エラーを誘発しないために、不正フレームにヘッダやデータから計算される正しい CRC を付与する必要がある。

d) CRC の変更による不正フレーム送信と不正挙動の誘発

正規フレームの CRC を変更した不正フレームを正規フレームとは独立に攻撃者が送信する。この時、CRC エラーにより受信ノードで該当フレームは破棄されるため、なりすまし攻撃により不正挙動を誘発することは難しい。

正規・不正フレームの衝突により、CRC を変更した場合、不正フレームに含まれる CRC と受信ノードがヘッダおよびデータから計算する CRC が異なるため、受信ノードで該当フレームが破棄される。よって、なりすまし攻撃により不正挙動を誘発することは難しい。

② FlexRay [5]

(ⅰ) FlexRay の概要

FlexRay は、高性能パワートレイン、安全機能といったアプリケーション向けに開発された通信プロトコルである。メディアアクセス方式は、同期通信が可能なタイムトリガ方式を採用しており、あらかじめメッセージ送信に関してスケジューリングされている。また、CAN と同様、非同期のイベントトリガ方式も同時に行えるようになっている。ネットワーク構成は、バス型、スター型、それらのハイブリッド型をとることができる。図 3.2d.1-6 に例としてバス型（単一チャネル構成）を示す。

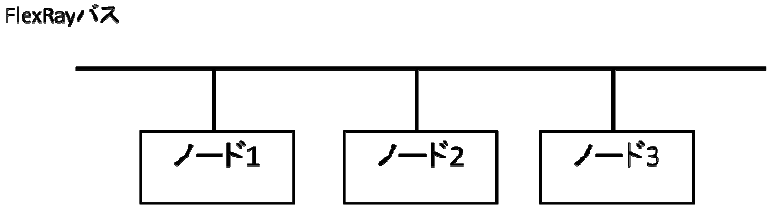


図 3.2d.1-6　FlexRay のネットワーク構成例（バス型、単一チャネル）

FlexRay ネットワークのアクセス処理の基盤となる通信サイクルは、「静的セグメント（必須）」、「動的セグメント（オプション）」、「シンボルウィンドウ（オプション）」、「ネットワークアイドル時間（必須）」の 4 つのセグメントで構成される（図 3.2d.1-7）。各セグメントは以下の通りの意味を持つ、

・静的セグメント：タイムトリガに基づき固定周期でデータを送信するためのセグメント（静的スロットと呼ぶ固定長のスロット群で構成される）
・動的セグメント：イベントトリガに基づき非同期でデータを送信するために用意されたセグメント（ミニスロットと呼ぶ可変長のスロット群で構成される）
・シンボルウィンドウ：ネットワークのメンテナンス等で利用するセグメント
・ネットワークアイドル時間：ノード・クロック間の同期を維持するためのセグメント

図 3.2d.1-7　FlexRay の通信サイクル

図 3.2d.1-8 に静的セグメントと動的セグメントを利用した通信例（単一チャネルの場合）を示す。図 3.2d.1-8 は、4 つの静的スロット及び 7 つのミニスロットに分割されている。通常、スロット番号とフレーム ID の番号は一致する。

静的セグメントに関して、最初に ID1 のフレームが送信され、スロット番号 2 は空きであり、次に ID3 および ID4 のフレームが送信さる。各フレームはバス上に接続された全てのノードが受信し、あらかじめ実装された動作を行う。動的セグメントに関して、ID5、6、8 のフレームが順に送信される。図 3.2d.1-9 にフレームのデータフォーマットを示す。FlexRay は CAN とは異なり、データフォーマットは 1 種類のみである。詳細な内容は仕様[5]を参照のこと。

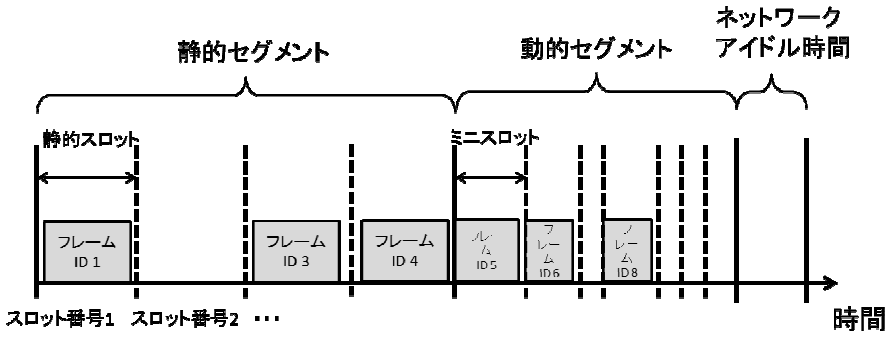


図 3.2d.1-8　静的セグメントと動的セグメントにおけるフレーム送信

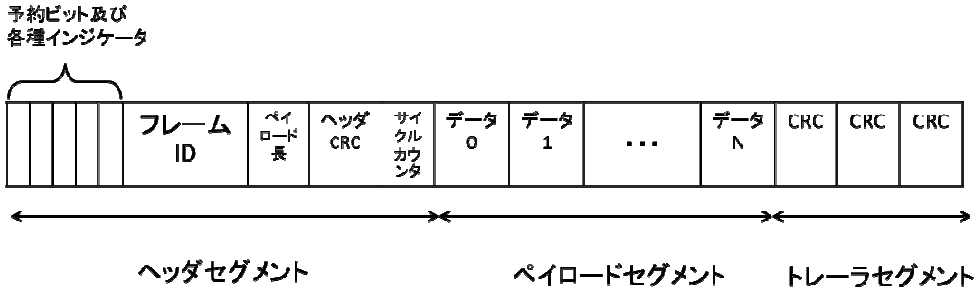


図 3.2d.1-9　FlexRay のフレーム構造

(ⅱ) FlexRay 通信上での DoS

以下に、CAN 通信上の DoS 攻撃を FlexRay 通信に適用・拡張した場合の攻撃方法に関して述べる。

a) バスへの高頻度送信により、正規レスポンス送信を抑制

・任意のノードに影響を与える

CAN はバス上にメッセージが送信されている場合は、他のノードは送信を抑制する。よって、攻撃者が不正メッセージでバス上を占有することが可能であった。

FlexRay は、CAN とは異なり、バス上の状態に関わらず、あらかじめ決められた時間内でスケジュールに基づきフレームが送信されるため、CAN と同じの方法で正規レスポンス送信を抑制することは難しい。

しかし、受信ノード内のエラー発生後の実装方法に依存するが、以下の方法により攻撃が可能であると考える。フレームを高頻度に送信することにより、受信ノードに以下のいずれかのエラーを発生させる。この時いずれかのエラーが発生した場合に、アプリケーション側で受信フレームを破棄するという実装がなされている場合は、各ノードにフレームを受信させることができなくなるため、動作に影響を及ぼすことが可能である。

－1 つのスロット内で 2 つ以上のフレームが受信されたとき

－静的セグメントまたは動的セグメント上で受信したフレームに格納されている ID と、スロット番号と一致しないとき

－動的セグメント上で ID が 0 となったとき

・特定ノードに影響を与える：

CAN では、バス上に流れるメッセージの ID の優先度により、特定ノードに影響を与えることが可能であった。

上記、任意のノードに影響を与える攻撃、で記載した理由により、CAN の攻撃方法をそのまま適用することは難しいが、前述の通り、該当するフレーム ID のみを攻撃対象とすることにより、特定ノードに影響を与えることが可能であると考える。

また、FlexRay 特有の攻撃として、動的セグメントにおいては、優先順位の低いフレーム送信（大きいスロット番号に割り当てられているフレーム）がこのセグメントに割り当てられた時間を超えた場合、該当サイクル内においてフレームが送信されないという特性がある。これを攻撃に利用して、スロット番号の小さい開放されたミニスロットがあれば、それを不正フレームで埋めることにより、優先順位の低いフレーム送信(大きいスロット番号に該当するフレームの送信)を抑制することが可能である。

・他ノードに影響を与える

FlexRay のフレームの利用方法によっては、特定ノードを攻撃対象とすることで、他ノードに影響を与えることができると考える。例えば、図 3.2d.1-10 のように、静的セグメントが開始されてから 4 セグメント（1～4）まではある車両データを含むフレームが送信され、スロット番号 6 でそのデータを読み取った上で処理を行った結果を反映したフレームを送信することを考える。

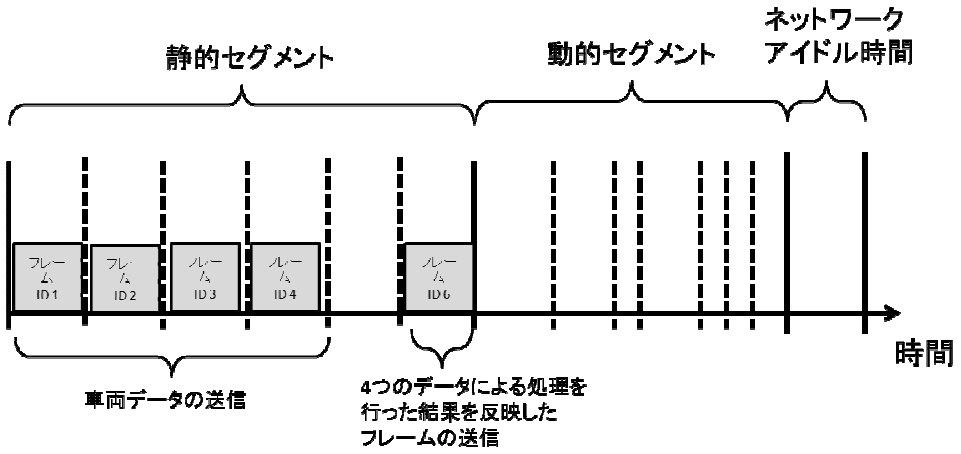


図 3.2d.1-10　サイクル内の静的セグメントにおける処理例

ここで、攻撃者がスロット番号 2 のフレーム ID を変更し、該当フレームを無効にした場合、正常とは異なるデータ値を読み込むことになるため、フレーム（ID6）に影響を与えることが可能であり、結果、動作に影響を及ぼすことが可能である。

b) 正規および不正フレームとの衝突により、正規ノードをバスから論理的に切り離し、正規フレーム送信を抑制

CAN では正規メッセージと不正メッセージを意図的に衝突させることにより、正規ノードをバスから切り離し、正規レスポンスを抑制することが可能であった。

FlexRay の場合は、衝突等により何らかのエラーが発生した場合、ノーマル・パッシブ（フレーム送信を停止、受信は継続）に遷移する。また、致命的なエラーが発生した場合は停止状態（フレームの送受信を停止）に遷移する。FlexRay では、基本的には多少のエラーが発生しても通信を継続するため[6]、衝突によりノーマル・パッシブに遷移させることは難しいと考える。また、例えノーマル・パッシブに移行できたとしても、本状態ではフレームの送信を行うことはできないため、文献[7]で提案されている衝突による攻撃により、停止状態まで遷移させて、論理的に正規ノードをバスから切り離すことは難しいと考える。

c) 正規フレーム以外のフレーム送信により、正規フレーム送信を抑制

CAN では診断用メッセージを送信することにより、特定ノードの正規メッセージ送信を抑制することが可能であった。

FlexRay の場合も同様に、攻撃者が不正のスリープフレームを送信することにより、FlexRay のコントローラのパワーセーブを不能にし、正規ノードのフレーム送信を抑制することが可能である[2]。ただし、バス上に接続された全てのノードに影響を及ぼすため、特定のノードのみのフレームを抑制することは難しい。

d) ヘッダセグメントの変更により、正規フレームの送信を抑制

ここでは、ヘッダセグメントのうち、制御に関係すると想定される、フレーム ID、ペ

イロード長およびヘッダ CRC の変更による正規フレームの送信の抑制が可能であるかを検討する。

・フレーム ID の変更

(1) ②の攻撃で記載した通り、攻撃者が意図的にフレーム ID をスロット番号と異なるものに変更した場合エラーが発生し受信ノードのエラー発生後の実装に依存するが、エラー発生後に受信フレームを破棄する場合は、正規フレームの受信を抑制することが可能である。また、FlexRay のプロトコル仕様によりフレーム ID が“0”の場合は無効となるため、衝突等により意図的に“0”に変更することにより該当フレームを無効にすることが可能である。

また、フレーム ID を変更することによりヘッダ CRC が変わるため、エラー発生後にアプリケーション側で受信フレームを破棄する場合は、CRC の不一致により該当フレームを無効にすることも可能である。

・ペイロード長の変更

ペイロード長を変更することにより、ヘッダセグメントに格納されたペイロード長と受信した際のペイロードの長さが異なることからエラーが発生する。受信ノードのエラー発生後の実装に依存するが、エラー発生後に受信フレームを破棄する場合は、正規フレームの受信を抑制することが可能である。

また上記と同様に、ペイロード長を変更することによりヘッダ CRC が変わるため、エラー発生後にアプリケーション側で受信フレームを破棄する場合は、CRC の不一致により該当フレームを無効にすることも可能である。

・ヘッダ CRC の変更

エラー発生後にアプリケーション側で受信フレームを破棄する場合は、CRC の不一致により該当フレームを無効にすることが可能である。

e) ペイロードセグメントの変更により、正規フレームの送信を抑制

上記 b.で述べた通り、FlexRay では多少のエラーが発生した場合でも通信を継続する[6]。また、プロトコル仕様では、ビットエラーのようなバスを監視して送信データとバス上のデータが異なる場合にエラーを検知する仕組みは規定されていない。そのため、アプリケーション側でビットエラーに対する何らかの対処が行われていない限り、ペイロードセグメント内のデータを変更することによる DoS 攻撃は難しい。

f) トレーラセグメントの変更により、正規フレームの送信を抑制

FlexRay のプロトコル仕様により、トレーラセグメントはフレーム全体でエラーが発生しているかどうかをチェックするための機能を備える。よってフレーム内の CRC 値が攻撃者によって変更された場合、ヘッダセグメントとペイロードセグメントから計算されたそれと異なるためエラーが発生する。エラー発生後にアプリケーション側で受信フレームを破棄する場合は該当フレームを無効化することが可能である。

(ⅲ) FlexRay 通信上でのなりすまし

以下に、CAN 通信上のなりすまし攻撃を FlexRay 通信に適用・拡張した場合の攻撃方法に関して検討した結果を述べる。

a) フレームのリプレイ攻撃

CAN では、攻撃者がバス上のメッセージを傍受した後、同等のメッセージを送信することによりリプレイ攻撃が可能であった。FlexRay 通信においても、フレームは暗号化されていないため、攻撃者はそれらの傍受は可能である。

FlexRay では、スケジュールに基づき、フレームの送信タイミングが決まっているため、正規フレームとリプレイ攻撃用の不正フレームとが衝突する可能性がある。そのため、CAN 通信上のリプレイ攻撃に比べ、各サイクル間で不正フレームを送信するなどといった送信タイミングを工夫する必要があり、サイクル間の時間によってリプレイ攻撃が難しい場合がある。

b) ヘッダセグメントの変更による不正フレーム送信と不正挙動の誘発

・フレーム ID の変更

フレーム ID が送信されるタイミングで衝突等を起こすことにより ID の変更は可能である。しかし ID とスロット番号が異なるためエラーが発生し、受信ノードでフレームを破棄する実装がなされている場合はなりすましフレームにより不正挙動を誘発することは難しいと考える。

・ペイロード長の変更

ペイロード長が送信されるタイミングで衝突等を起こすことによりその値の変更は可能である。しかし、フレームに含まれるペイロードと実際にノードが受信したデータの長さが異なる時であっても、フレームを破棄する実装がなされている場合はなりすましフレームにより不正挙動を誘発することは難しいと考える。

・ヘッダ CRC の変更

ヘッダ CRC が送信されるタイミングで衝突等を起こすことにより、その値の変更は可能である。しかし、フレームに含まれる CRC 値と受信ノードで計算した CRC 値が異なる時、フレームを破棄する実装がなされている場合は、なりすましフレームにより不正挙動を誘発することは難しいと考える。

c) ペイロードセグメント内のデータの変更による不正フレームの送信と不正挙動の誘発

CAN では任意のタイミングで攻撃者がなりすましデータを送信することにより、不正挙動の誘発が可能であった。

FlexRay においては、送信タイミングがあらかじめ決まっているため、同様の方法では正規レスポンスと衝突する可能性がある。よって、なりすましフレームの送信方法を工夫する必要がある。また、文献[8]によると、FlexRay 上においても単純になりすましフレームをバス上に送信し、受信ノードにそれを受信させることにより不正挙動を誘発することは可能とある。しかし、あるペイロードセグメント内のデータを変更する場合は

衝突のタイミング等を工夫する必要があり、単純に不正フレームをバス上に送信するのみではデータを任意の値に変更できない可能性がある。

ヘッダセグメントと同様、データが送信されるタイミングで衝突等によりペイロードセグメント内のデータを変更することは可能であり、そのなりすましデータ値によって受信ノードの不正挙動を誘発することは可能である。但し、不正挙動を誘発するためには、なりすましフレームのトレーラセグメント内の値も、正規フレームの値からなりすましフレームから計算される値に変更する必要がある。

また、FlexRay 特有の攻撃方法として、静的及び動的セグメント内に空きスロットがある場合、該当スロットのタイミングでスロット番号と一致した ID を持つなりすましフレームを送信することにより受信ノードに不正挙動を誘発することが可能であると考える。ただし、該当 ID を受信した後に何らかの動作を行うノードがバス上に存在する場合のみ、本攻撃が有効である。

d) トレーラセグメント内の CRC の変更による不正フレームの送信と不正挙動の誘発

CRC 値が送信されるタイミングで、衝突等により CRC 値を変更することは可能である。しかし、フレーム内の CRC 値と受信ノードで計算された値が異なる時、フレームを破棄する実装がなされている場合は、なりすましフレームにより不正挙動を誘発することは難しいと考える。

③ LIN[1]

(ⅰ) LIN の概要

LIN は CAN 等のサブバスとして位置付けられており、シングルマスタ方式を採用する。図 3.2d.1-11 に LIN の構造例を示す。図 3.2d.1-11 のように単一の LIN バス上に単一のマスタノードと単一または複数のスレーブノードが接続される。

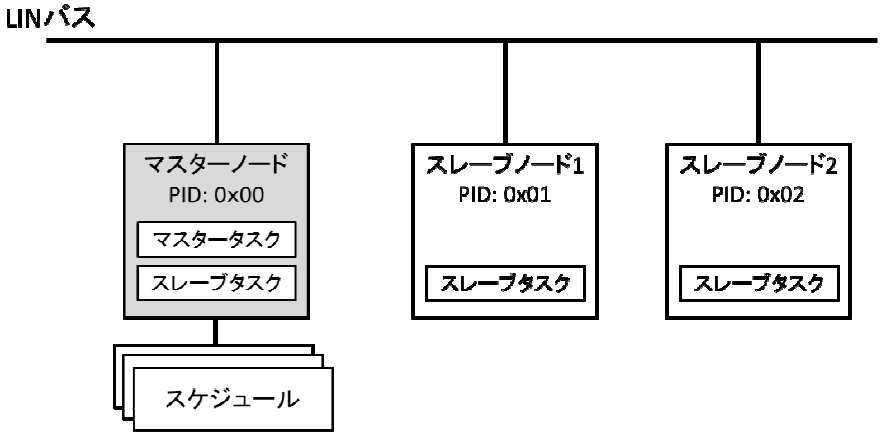


図 3.2d.1-11　LIN のクラスタ（単一のマスタノードと複数のスレーブノードから成る）

LIN 通信はマスタノードのヘッダ送信を合図に開始される。図 3.2d.1-12 に示す通り、あらかじめ定められたスケジュールに基づき、マスタノードがヘッダを送信し（図 3.2d.1-12（ア））、該当 ID を持つスレーブノードがレスポンスを送信する（図 3.2d.1-12（イ））。レスポンスはバス上にブロードキャストされるが、該当 ID と受信用 ID が一致するノード（この場合はスレーブノード 2））のみがレスポンスを受信・解釈し（図 3.2d.1-12（ウ））、あらかじめ実装された動作を行う。LIN 通信では、ヘッダとレスポンスを合わせてフレームと呼ぶ。図 3.2d.1-13 に各データフォーマットを示す。詳細な内容は仕様[1]を参照のこと。

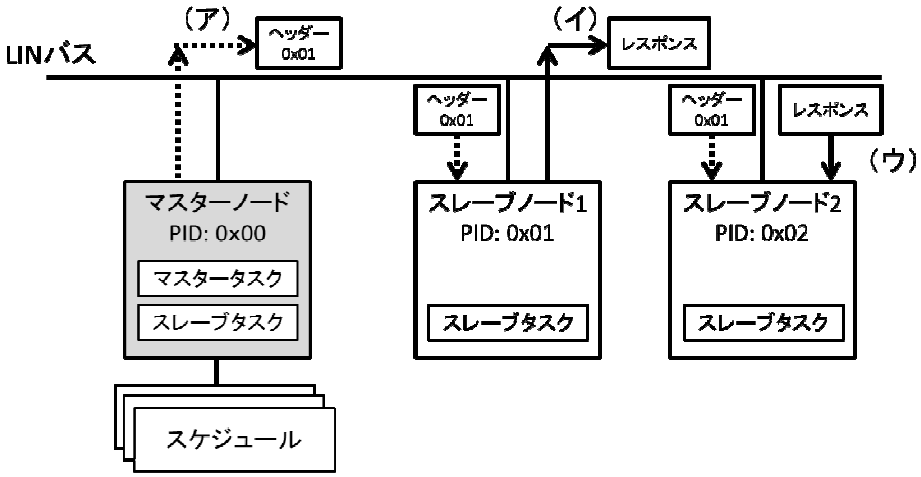


図 3.2d.1-12　メッセージフレームの送信および受信の方法

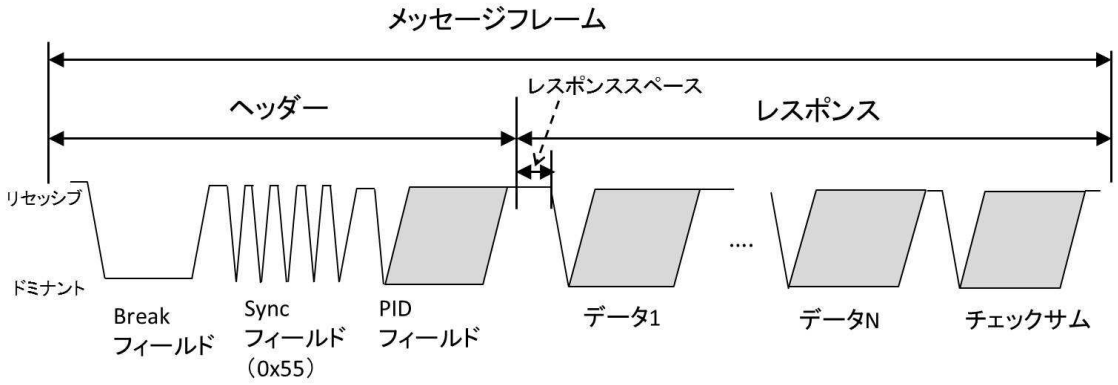


図 3.2d.1-13　LIN のデータフォーマット

(ⅱ) LIN 通信上での DoS

以下に、CAN 通信上の DoS 攻撃を LIN 通信に適用・拡張した場合の攻撃方法に関して述べる。

a) バスへの高頻度送信により、正規レスポンス送信を抑制

・任意のノードに影響を与える

CAN はバス上にメッセージが送信されている場合は他のノードは送信を抑制する。よって、攻撃者が不正メッセージでバス上を占有することが可能であった。LIN の場合

も同様に、攻撃者が不正ヘッダ（または不正レスポンス）を高頻度で送信し、バスを占有することは可能であると考える。ただし、LIN は、CAN とは異なり、マスタノードは攻撃者からの不正メッセージの送信の有無に関わらずスケジュールに基づきヘッダを送信するため、そのヘッダと攻撃者が送信する不正ヘッダ（または不正レスポンス）との衝突が発生する可能性がある。

・特定ノードに影響を与える

CAN では、バス上に流れるメッセージ ID の優先度により特定ノードの動作に影響を及ぼすことが可能であった。LIN の場合は、タスクの優先順位はマスタノードのスケジュールで決まる。そのため、特定ノードのみを妨害するためには、CAN とは異なり攻撃対象のノードの動作を誘発する ID を含む正規ヘッダがマスタノードから送信された場合に、同時に攻撃者が不正ヘッダ（または不正レスポンス）を同タイミングで送信することにより、両者の衝突を起こす必要がある。よって、CAN に比べて攻撃の難易度はあがるが、衝突を起こすことができれば特定ノードの動作を妨害することが可能である。

・他ノードに影響を与える

CAN では、バス上に流れるメッセージの ID の優先度により他ノードの動作に影響を及ぼすことが可能であった。LIN の場合は前述の通り、タスクの優先順位はマスタノードのスケジュールで決まる。よって、他ノードに影響を及ぼすには、前述の通り他ノードに何らかの動作を要求する ID を含むヘッダを衝突等により妨害することができれば、他ノードに影響を及ぼすことが可能である。

b) 正規レスポンスとの衝突により、正規レスポンス送信を抑制

CAN では、正規メッセージと不正メッセージを意図的に衝突させることにより、正規ノードをバスから切り離し正規レスポンスを抑制することが可能であった。

LIN の場合は、バス上での衝突が許容されていないフレーム送信時（例えば無条件フレーム送信時）に正規レスポンスと攻撃者が送信するレスポンスによる衝突が発生することにより、各ノードがエラーと判断した場合には limp home mode に移行する場合がある（limp home mode とは通信に最低限必要な機能以外を停止するモードである）。よって、CAN のように完全にバスから論理的に切り離すことは難しいが、攻撃者が衝突を頻繁に発生させることにより攻撃対象のノードを limp home mode に移行させ、本来のノードの動作を抑制することは可能であると考える。

c) 正規レスポンス以外のフレーム送信により、正規レスポンス送信を抑制

CAN では診断用メッセージを送信することにより、特定ノードの正規メッセージ送信を抑制することが可能であった。LIN の場合も、マスタノードが送信する強制スリープモードの診断要求フレーム（go-to-sleep コマンド）を攻撃者がマスタノードになりすまして送信することで、バス上に接続された全てのスレーブノードをスリープモード（スリープモードとは、レスポンスの送受信を行わないモードである）に移行させることが可能である[2][3]。文献[3]では、バスがアイドル状態の場合と正規ヘッダが送信される場合の攻撃方法を示している。

ただし、本フレームを利用することによりバス上の全てのノードがスリープモードに移行するため、全てのノードのレスポンス送信を抑制することは可能であるが、LIN の場合は、攻撃対象とする特定ノードのみのレスポンス送信を抑制することは難しいと考える。

d) ヘッダの変更により、正規レスポンスの送信を抑制

以下に、正規・不正フレームをタイミング良く衝突させること等によりヘッダを変更することによって DoS 攻撃を行う攻撃方法を示す。

・PID の変更

LIN では、ヘッダの PID フィールド値によりレスポンスの送信者が決まる。よって、特定のノードの送信を抑制したい場合、該当する PID を含むヘッダがマスタノードから送信されるのと同時に、攻撃者が不正ヘッダを送信することにより、各ノードに割り当てられていない ID に変更することが可能であれば、該当 ID を持つノードの発信を抑制することが可能である。

・Break Field の変更

LIN のヘッダでは、図 3.2d.1-12 に示す通り、フレームの開始を示す Break Field が設定されており、13 ビット以上のドミナント（Break と呼ぶ）と、1 ビット以上のリセッシブ（Break-delimiter と呼ぶ）から成る。バスの電気的特性により、バス上のドミナント値をリセッシブに変更することを遠隔から実施することは難しいため、最初の 13 ビットをリセッシブに変更することは困難である。しかし Break-delimiter を衝突により 0 に変更し Break と思わせ、本来であれば Sync byte の「1」を Break-delimiter であると各ノードが錯覚すれば、Sync byte に影響を及ぼし、バス全体に影響を及ぼすことが可能である。

・Sync Field の変更

LIN のヘッダでは、各ノード間で同期取得に利用する Sync Byte が設定されており、0x55 で定義されている。Sync Byte を変更することにより、バス上の通信レートの同期に不具合を起こすことが可能であるため、通信を不能にすることが可能である[2]。

e) レスポンス内のデータの変更により、正規レスポンスの送信を抑制

ここで、正規・不正レスポンスをタイミング良く衝突させること等によりレスポンス内のデータを変更することによって DoS 攻撃を行う攻撃方法を示す。

文献[4]に記載がある通り、正規ノードがレスポンスを発するタイミングと同時に攻撃者が不正レスポンスを送信・衝突させ、ビットエラーによるエラーハンドリング機構の発動により、正規レスポンスの送信の発信を止めることが可能である。

f) レスポンス内のチェックサムの変更により、正規レスポンスの送信を抑制

e) と同様に、レスポンス内のチェックサムが送信されるタイミングで、衝突等によりチェックサムを変更することは可能である。この時、レスポンスに含まれるチェックサムと受信ノードで計算したチェックサムが異なる時、レスポンスを破棄することが受信ノードで実装されている場合、該当フレームは破棄されるため、正規レスポンスの受信

を抑制することが可能である。

(ⅲ) LIN 通信上でのなりすまし

以下に、CAN 通信上のなりすまし攻撃を LIN 通信に適用・拡張した場合の攻撃方法に関して検討した結果を述べる。

a) フレームのリプレイ攻撃

CAN では、攻撃者がバス上のメッセージを傍受した後、同等のメッセージを送信することにより、リプレイ攻撃が可能であった。LIN 通信においても、ヘッダとレスポンスは暗号化されていないため、攻撃者はそれらの傍受は可能である。

LIN の場合は、受信者に正しくレスポンスを受信させるためにはヘッダの送受信が必要であるため、リプレイ攻撃を成功させるためには、攻撃者はヘッダとレスポンスの両方をバス上に送信する必要がある。また、リプレイ攻撃を実施するタイミングを考慮する必要があり、マスタノードからの正規ヘッダとスレーブノードからの正規レスポンスと、攻撃者が送信する不正ヘッダとフレームとの衝突を回避するために、攻撃者は、正規フレーム送信間に不正ヘッダと不正レスポンスを送信してリプレイ攻撃を実施する必要がある。そのため、CAN 通信上のリプレイ攻撃に比べ、不正メッセージの送信タイミングを工夫する必要があり、フレームの周期によってはリプレイ攻撃が難しい場合があると考える。

b) ヘッダの変更による不正レスポンスの送信と不正挙動の誘発

・PID フィールドの変更

LIN では、マスタノードがヘッダを発するタイミングと同時に攻撃者が不正ヘッダを送信・衝突させることで、バス上に流れるヘッダの値を変更することにより、本来とは異なる送信ノードにレスポンスを送信させることが可能である[4]。これにより、マスタノードに実装されたスケジュールを物理的に操作することなく、レスポンス送信の順番を変更することが可能である。その後、レスポンスの改ざんによる不正レスポンス送信の攻撃方法と同様に、レスポンスの衝突を誘発することにより、不正レスポンスをあたかも正しいものとして､受信ノードに受信･解釈させることが可能である。

・Break Field の変更

正規ヘッダと不正ヘッダの衝突のタイミングにより、最後の 1 ビットをドミナントに変更することは理論的には可能であるが、それを変更することでエラーを誘発することは可能であるものの、なりすまし攻撃により直接的に不正挙動を誘発することは難しいと考える。

・Sync Byte の変更

DoS 攻撃で記載した通り、Sync Byte を変更することにより、バス上の通信レートの同期に不具合を起こすことが可能である[2]。このことにより、バス上の通信を不能にさせることは可能であるが、なりすまし攻撃により不正挙動を直接的に起こすことは難しいと考える。

c) レスポンス内のデータ値の変更による不正レスポンス送信と不正挙動の誘発

CAN では任意のタイミングで攻撃者がなりすましメッセージを送信することにより、不正挙動の誘発が可能であった。LIN は、CAN とは異なり、レスポンスを有効にするためにはヘッダの送信が必要であるため、攻撃者は不正レスポンスを正規レスポンスよりも短い周期でバス上に送信することにより不正挙動を誘発することは難しいと考える。よって以下に示す方法により、不正メッセージの送信を行う必要がある。

すなわち、受信ノードに不正レスポンスをあたかも正しい送信ノードが送信したものとして受信・解釈させることにより不正挙動を誘発することが可能である[4]。

正規ノードがレスポンスを発するタイミングと同時に攻撃者が不正レスポンスを送信・衝突させ、ビットエラーによるエラーハンドリング機構の発動により、正規レスポンスの送信の発信は止まる。直後に攻撃者は不正レスポンスを送信することにより、不正レスポンスをあたかも正しいものとして受信ノードに受信・解釈させることが可能である。但し、本方法では、衝突を発生させたバイト値を任意の値に改ざんすることは不可能である。そのため、全バイト値を任意の値に改ざんするためには、正規レスポンスの発生を抑制し、その間に不正レスポンスを送信する方法が必要である。

d) レスポンス内のチェックサムの変更による不正レスポンス送信と不正挙動の誘発

送信ノードから正規レスポンスが送信され､チェックサムが送信されるタイミングで、攻撃者が不正レスポンスを送信することにより、チェックサムの値を変更することは可能であると考える。但し、データ値から計算されるチェックサムの値と、変更されたチェックサムとの値は異なり、受信ノードで該当レスポンスは破棄されるため、不正挙動を誘発することは難しい。

④ 他通信プロトコルへの適用可能性のまとめ

ここでは、上記①から③で述べた、CAN-FD、FlexRay、LIN 通信において、CAN 通信における攻撃方法を参考に適用可能であるかを机上で検討した結果をまとめる。

(ⅰ) CAN-FD 通信上での攻撃方法

表 3.2d.1-4 に、上記①で記載した CAN-FD 通信上での DoS 攻撃の攻撃可能性を示す。なお、各項目の意味は以下の通りである。

・攻撃の可否
- 可：攻撃が可能
- 難：攻撃が難しい

・CAN 攻撃の適用
- 適用可：CAN 上の攻撃方法をそのまま適用できる
- 工夫要：攻撃方法を工夫する必要がある
- 難：適用は難しい
- 該当なし：該当する攻撃方法が提案されていない
- ―：CAN 通信プロトコルの仕様では規定されていないパラメータを攻撃に利用

・新規性

- 新規：本研究による新規の知見（攻撃としては可能であると考えられるが、発表事例がない場合。）
- 既存[文献番号]：既存文献に記載がある場合

表 3.2d.1-4　CAN-FD 通信上における DoS 攻撃の可能性

攻撃の方法		攻撃の可否	CAN 攻撃の適用	新規性	備考
（1）高頻度送信	①任意のノードに影響	可	適用可	新規	
	②特定ノードに影響	可	適用可	新規	
	③他ノードへの影響	可	適用可	新規	
（2）正規フレームとの衝突		可	適用可	既存[5]	
（3）正規フレーム以外のフレーム送信		可	適用可	新規	
（4）ヘッダの変更	フレーム ID	可	該当なし	新規	変更後の ID が、各ノードの受信・解釈用の ID に割り当てられていない場合
	各種コントロールビット	可	該当なし	新規	ただし、BRS=1、ESI=1 の時のみ
	DLC	可	適用可	新規	
（5）データフィールドの変更		可	適用可	新規	
（6）CRC の変更		可	該当なし	新規	

表 3.2d.1-5 に、上記①で記載した CAN-FD 通信上でのなりすまし攻撃の攻撃可能性を示す。

表 3.2d.1-5　CAN-FD 通信上におけるなりすまし攻撃の可能性

攻撃の方法		攻撃の可否	CAN 攻撃の適用	新規性	備考
(1) フレームのリプレイ攻撃		可	適用可	新規	
(2) ヘッダの変更	フレーム ID	可	該当なし	新規	
	各種コントロールビット	可	該当なし	新規	ただし、攻撃の成功はコントロールビットの種類、送信ノードのエラー状態に依存する。
	DLC	可	該当なし	新規	正規・不正フレームいとの衝突を誘発する場合。攻撃成功は、送信ノードのエラー状態に依存する。
（3）データの変更		可	適用可	新規	
（4）CRC の変更		難	該当なし	新規	

(ⅱ) FlexRay 通信上での攻撃方法

表 3.2d.1-6 に、上記②で記載した FlexRay 通信上での DoS 攻撃の攻撃可能性を示す。各項目の意味は、表 3.2d.1-4 と同様である。

表 3.2d.1-6　FlexRay 通信上における DoS 攻撃の可能性

攻撃の方法		攻撃の可否	CAN 攻撃の適用	新規性	備考
(1)高頻度送信	①任意のノードに影響	可	工夫要	新規	受信ノードのエラー処理に関する条件あり
	②特定ノードに影響	可	工夫要	新規	(1)①と同様
	③他ノードへの影響	可	該当なし	新規	
(2)正規フレームとの衝突		難	難	既存[5]	
(3)正規フレーム以外のフレーム送信		可	適用可	既存[2]	特定ノードのみに影響を与えることは難しい。
(4)ヘッダセグメントの変更	フレーム ID	可	該当なし	新規	(1)①と同様
	ペイロード長	可	該当なし	新規	(1)①と同様
	ヘッダ CRC	可	―	新規	(1)①と同様
(5)ペイロードセグメントの変更		難	該当なし	新規	
（6）トレーラセグメントの変更		可	該当なし	新規	

表 3.2d.1-7 に、上記②で記載した FlexRay 通信上でのなりすまし攻撃の攻撃可能性を示す。各項目の意味は、表 3.2d.1-5 と同様である。

表 3.2d.1-7　FlexRay 通信上におけるなりすまし攻撃の可能性

攻撃の方法		攻撃の可否	CAN 攻撃の適用	新規性	備考
(1) フレームのリプレイ攻撃		可	工夫要	新規	攻撃の成功可否はサイクル間の時間に依存する。
(2) ヘッダセグメントの変更	フレーム ID	難	難	新規	エラー検知により、受信ノードでフレームを破棄する場合は不可。
	ペイロード長	難	難	新規	（2）フレーム ID と同様。
	ヘッダ CRC	難	―	新規	（2）フレーム ID と同様。
(3) ペイロードセグメント（データ）の変更		可	工夫要	新規 既存[8]	空きスロットを利用した攻撃方法は新規。
(4) トレーラセグメントの変更		難	該当なし	新規	（2）フレーム ID と同様。

(ⅲ) LIN 通信上での攻撃方法

表 3.2d.1-8 に、上記③で記載した LIN 通信上での DoS 攻撃の攻撃可能性を示す。各項目の意味は、表 3.2d.1-4 と同様である。

表 3.2d.1-8　LIN 通信上における DoS 攻撃の可能性

攻撃の方法		攻撃の可否	CAN 攻撃の適用	新規性	備考
(1)高頻度送信	①任意のノードに影響	可	適用可	新規	
	②特定ノードに影響	可	工夫要	新規	正規ヘッダと不正ヘッダの衝突等を起こすことが可能な場合、可能。
	③他ノードに影響	可	該当なし	新規	正規ヘッダと不正ヘッダの衝突等を起こすことが可能な場合、可能。
(2) 正規レスポンスとの衝突		可	適用可	新規	バスオフを発生させることは難しいが、limp home mode に移行させることは可。
(3) 正規フレーム以外のフレーム送信		可	適用可	既存[2,3]	
(4) ヘッダの変更	PID	可	該当なし	新規	各ノードに割り当てられていない PID に変更できる場合。
	Sync Field	可	—	既存[2]	
	Break Field	可	—	新規	Break-delimiter を衝突により 0 に変更できる場合。
(5)データの変更		可	該当なし	既存[4]	
(6) チェックサムの変更		可	該当なし	新規	

表 3.2d.1-9 に、上記③で記載した LIN 通信上でのなりすまし攻撃の攻撃可能性を示す。各項目の意味は、表 3.2d.1-5 と同様である。

表 3.2d.1-9　LIN 通信上におけるなりすまし攻撃の可能性

攻撃の方法		攻撃の可否	CAN 攻撃の適用	新規性	備考
(1) フレームのリプレイ攻撃		可	工夫要	新規	攻撃の成功可否は正規フレームの周期に依存する。
(2) ヘッダの変更	PID	可	該当なし	既存[4]	
	Sync Byte	難	―	新規	
	Break Field	難	該当なし	新規	
(3)データの変更		可	該当なし	既存[4]	衝突を発生させたバイト値を任意の値に変更することは不可。
(4)チェックサムの変更		難	該当なし	新規	エラー処理により、レスポンスが破棄される場合。

(ⅳ)考察

上記検討結果より、CAN 通信上の攻撃方法を他通信プロトコルへ適用した場合や拡張した場合の攻撃可能性に関して考察する。特にプロトコル構成やアクセス方式の違いが各攻撃方法にどのように影響をするかを述べる。

a) DoS 攻撃に関して

CAN や CAN-FD といったマルチマスタ構成、かつイベントトリガ型のアクセス方式を採用するプロトコルでは、攻撃者が DoS 攻撃用の不正メッセージを送信中は、正規ノードによるメッセージの送信が抑制される、また、攻撃者は任意のタイミングで不正メッセージを送信することが可能であることから、比較的簡単に攻撃を成功させることが可能である。また、これらのプロトコルはメッセージ調停が行われ、ID の値によってバス上でのメッセージ送信の優劣が決まるため、この特性を利用することによっても DoS 攻撃が可能である。

一方で、LIN のシングルマスタ構成や、FlexRay のタイムトリガ型のアクセス方式を採用するプロトコルの場合は、あらかじめメッセージの送信タイミングが管理されており、メッセージ調停の概念がないため、単純な DoS 攻撃により正規メッセージの送信を抑制することは難しい。そのため、不正メッセージの送信タイミングの工夫や、エラー検知後のアプリケーションによる実装処理の特性を利用するなどの工夫をすることにより、攻撃が可能となる。

また、ノードを休止状態に移行させるための特別なメッセージは全てのプロトコル仕様で規定されていることが多いため、プロトコル構成やアクセス方式の違いに依らず、これを利用した DoS 攻撃は可能である。

b) なりすまし攻撃に関して

マルチマスタ・プロトコル、かつイベントトリガ型のアクセス方式を採用するプロトコルでは、上記、DoS 攻撃で記載した理由により、単純に攻撃者がリプレイ攻撃用のメッセージを送信することにより、本攻撃が可能である。

一方で、シングルマスタ・プロトコルやタイムトリガ型のアクセス方式を採用するプロトコルにおいては、正規フレームの送信が常時行われることにより、これと衝突を回避するようにリプレイ攻撃用の不正フレームを送信するために送信タイミングを工夫すれば攻撃が可能である。

また、シングルマスタ・プロトコルやタイムトリガ型のアクセス方式を採用するプロトコルにおいては、フレームの変更によるなりすまし攻撃に関して、攻撃者が任意の値に変更した不正フレームを送信するためには、空きスロットを利用する、衝突を起こすタイミングを工夫して攻撃を実施する、エラー検知後のアプリケーションによる実装処理の特性を利用するといった工夫を行うことにより攻撃が可能となる。

以上から、シングルマスタ・プロトコルやタイムトリガ型のアクセス方式の通信プロトコルの攻撃有効性評価を行う場合は、マルチマスタ・プロトコル、イベントトリガ型のアクセス方式を採用するプロトコルにおける攻撃方法をそのまま適用できる場合もあるが、その多くは上記で述べた工夫を行った攻撃方法により、有効性評価を実施する必要がある。

3.2d.2　シミュレータによる評価方法の有効性検証

ここでは、一般的に ECU や車内ネットワークの開発等に用いられる市販のソフトウェアツール（以下「車内通信プロトコルシミュレータ」という）を用いて、3.2d.1.(2)にて分類した CAN 通信上における攻撃の再現が可能であるか、および、シミュレーションによる攻撃方法の評価が有効であるかを検証した結果を述べる。まず、シミュレーション環境として、利用した機器の仕様と、シミュレートした車内ネットワークの構成の概要を述べる。次に、シミュレータを用いた安全性評価方法の有効性を評価する観点について述べる。

① シミュレーション環境

シミュレーションに用いた機器一式を表 3.2d.2-1 に示す。市販の車内通信プロトコルシミュレータとして、CANoe[1]を用いた。CAN に対して脆弱性あるいは攻撃方法を指摘している文献のうちシミュレーションによる有効性評価を行っている文献は全て CANoe を利用している。そのため本研究でも CANoe を利用することとした。

CANoe は ECU 単体あるいは車両のネットワーク全体の開発を目的とした Windows OS マシン用ソフトウェアである。攻撃方法の有効性評価を実施する上で必要な、CANoe に特徴的な機能を以下に述べる。各 ECU が制御する機能、例えばエンジンや運転席の表示パネルなどを制御する各 ECU の動作をシミュレートできる。そしてエンジンや表示パネルなどの制御結果をグラフィカルにシミュレートできる。そして CANoe 上に設計した仮想的な ECU 間の CAN 通信をシミュレートできる。さらに Windows OS マシンと CAN インタフェ

ース機器を接続することで、仮想的な CAN 通信と実際の CAN 通信を混在させてシミュレートできる。

以降では、CANoe 上の仮想的な CAN 通信と実際の CAN 通信を区別するために以下の用語を用いる。

・仮想バス：CANoe 上でシミュレートされる仮想的な CAN バス。
・実バス　：実際に電気信号がワイヤ内に流れる CAN バス。具体的には、実車両に用いられる CAN バスや表 3.2d.2-1 のジャンパケーブル等によって構成された CAN バスを指す。

CANoe 上の仮想バスでは、脆弱性あるいは攻撃方法が指摘されている CAN の仕様の内、一部の仕様を再現することが難しい。具体的には、複数のメッセージが同時に送信された際に起こる、調停部以外でのメッセージの衝突に伴うエラーフレームが送信ノードから送信されない。またエラーフレームの送受信による各ノードのエラーカウンタが仕様通りに変化しない。よって、エラーフレーム送信によるエラーカウンタの増大を繰り返すことによってバスオフを誘発する攻撃（②DoS（正規メッセージとの衝突メッセージ送信（バスオフを誘発））の評価を、CANoe 上のみで再現することは難しいと考え、市販の CAN の仕様に則したマイコンを組み合わせることにより評価を実施した。

表 3.2d.2-1　シミュレーション機器

No.	機器名	説明
1	ノートパソコン 1	● 型版：DELL 株式会社製 LATITUDE E6530 ● CPU：Intel Core i7-3520M ● メモリ（RAM）：8.00GB ● オペレーティングシステム：Windows 7（64 ビット） ● CANoe 9.0.65（SP2）の実行環境 ● 仮想バスによるシミュレーション用機器として使用 ● 実バスに接続し通信ログ取得用機器として使用
2	ノートパソコン 2	● 型版：パナソニック株式会社製： Let’s note CF-SX3 ● CPU：Intel Core i5-4300U ● メモリ（RAM）：4.00GB ● オペレーティングシステム：Windows 7（32 ビット） ● ECU 擬似端末 Arduino UNO の動作設定機器、給電機器として使用
3	CAN インタフェース	● 型版：ベクター・ジャパン株式会社製 VN1630A ● CANoe（ノートパソコン 1）と実バスの接続に使用
4	ECU 擬似端末	● 型版：Arduino 社製 Arduino UNO ● ECU のアプリケーション層のシミュレーション用機器として使用
5	CAN 通信用モジュール	● 型版：HiLetgo MCP2515 CAN バス モジュール TJA1050 レシーバーSPI モジュール for Arduino AVR ● CAN コントローラ：MCP2515 ● CAN トランシーバ：TJA1050 ● ECU の CAN コントローラと CAN トランシーバのシミュレーション用機器として使用
6	抵抗	● 実バスの終端抵抗（120Ω）として使用
7	ジャンパケーブル	● ECU 擬似端末と CAN 通信用モジュールとブレッドボードの各接続に使用 ● 実バスの一部として使用
8	ブレッドボード	● 実バスの一部として使用
9	USB ケーブル	● ノートパソコン 2 と ECU 擬似端末の接続に使用

次に、シミュレートした車内ネットワークの構成の概要を示す。各攻撃方法の車内ネットワーク構成は、CANoe の利用サンプルの 1 つとして提供される「Easy.cfg」をベースに構築した。車内ネットワークには、正規の送信 ECU、正規の受信 ECU、正規の ECU が乗っ取られて攻撃者と化した攻撃 ECU がそれぞれを少なくとも 1 台以上接続している。図 3.2d.2-1 に構成を例示する。図 3.2d.2-1 は仮想バスのみで構成される車内ネットワークを示しており、正規の送信 ECU が 2 台、正規の受信 ECU が 1 台、正規の ECU を乗っ取った攻撃 ECU が 1 台、同一の仮想バスに接続している。攻撃方法の違いに応じて「Easy.cfg」の各々の ECU の挙動を修正することにより、各シミュレーション環境を構築した。各攻撃方法の車内ネットワーク構成は本項(1)攻撃再現手順で述べる。

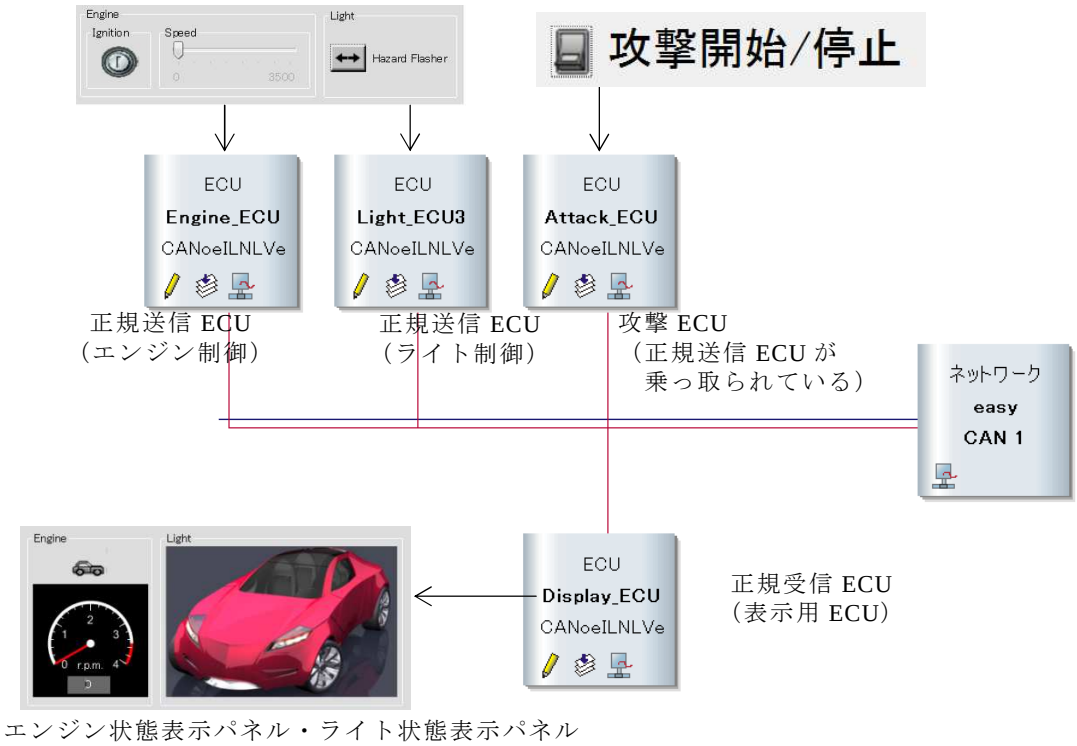


図 3.2d.2-1　仮想バスを用いた車内ネットワーク構成例

② 有効性評価の観点

シミュレータを用いた車内通信プロトコルの評価方法が有効であるかを、以下の観点から評価する。

- 通信プロトコルの再現性：車内通信プロトコルの仕様どおりシミュレーションが行われているか、特に脆弱性が指摘されている箇所が再現されているか、また、それを確認できるかどうか。
- 攻撃方法の再現性：攻撃に関わる各 ECU（正規の送受信 ECU、攻撃 ECU）をシミュレーションできるかどうか、特に攻撃 ECU の攻撃手順が再現できるかどうか。
- 攻撃結果の再現性：攻撃の成功有無、つまり評価対象の被害の有無、および被害がどの程度の範囲に及んでいるかを確認できるかどうか。
- 実施の効率性：シミュレーションの実行時間が実車輌で攻撃を行った場合と同等あるいはそれ以下か。また、評価対象のネットワーク構成や攻撃者の設定（攻撃メッセージの値や周期・頻度など）の変更が容易かどうか。

③ 有効性評価の手順

シミュレータによる評価方法の有効性を検証する手順を図 3.2d.2-1 を例として用いて説明する。仮に、図 3.2d.2-1 は通信路全体に対する高頻度送信を実施する環境とする。

まず、評価を行う者は攻撃を実行する前に CANoe が脆弱性を確認するべき対象の動作、すなわちこの例の場合高頻度送信に関わる CAN の仕様を再現しているか確認しておく。

具体的には、CAN-ID 部におけるメッセージの調停機能が仕様通り再現されるか確認しておく。次に、再現しようとする攻撃の手順、すなわちこの例の場合攻撃メッセージを高頻度に送信する攻撃 ECU を設計する。そして、攻撃を実行したのち想定した被害が再現されているか確認する。この例の場合、被害が再現されているとはエンジンやライトの表示更新が遅延または停止したことを評価者が目視で確認でき、通信ログからも被害の有無および被害の規模を確認できる状態を指す。最後に本シミュレーションに要した時間を確認する。

上記の各観点を満たしていれば、車内通信プロトコルのセキュリティ対策に対して、通信路全体に対する高頻度送信のシミュレーションによる評価方法は有効だと判断する。それぞれの攻撃方法において上記の手順を実施して、シミュレータを用いた評価方法の有効性を確認する。

(1) 攻撃再現手順

ここでは、シミュレータで各攻撃方法を実現するための具体的な攻撃手順を述べる。車内通信プロトコルの完全性を毀損するなりすまし攻撃、並びに可用性を毀損する DoS 攻撃のそれぞれの攻撃方法について、シミュレーション環境と攻撃手順の概要、そして実際に行った攻撃手順を示す。

なお、本項では「攻撃 ECU」と「攻撃者」を以下のとおり区別する。攻撃 ECU とは、シミュレーション環境内に存在し、攻撃を実行するプログラムとする。攻撃者とは、シミュレーション環境の外側に存在し、環境全体の挙動（正常な送信 ECU と受信 ECU、攻撃 ECU、通信路の挙動）を確認できる人間あるいはプログラムとする。

① なりすまし（リプレイ送信）

図 3.2d.2-2 にリプレイ送信を再現するためのシミュレーション環境を示す。

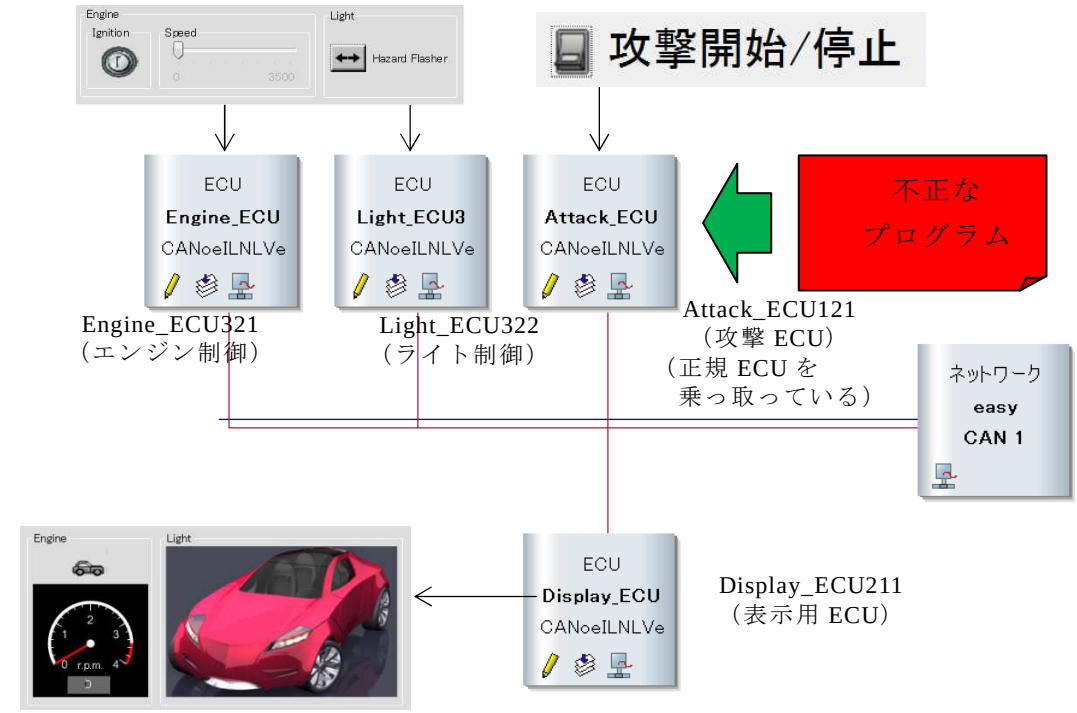


図 3.2d.2-2　リプレイ送信のシミュレーション環境

図 3.2d.2-2 の各 ECU の機能を説明する。

Engine_ECU321 は、表 3.2d.2-2 に示す CAN メッセージを一定の周期で送信、またはイベント発生時に送信する。Ignition の切り替え操作はエンジン操作パネルから攻撃者が行う。

Light_ECU322 は表 3.2d.2-3 に示す CAN メッセージを一定の周期で送信する。ライトの操作はライト操作パネルから攻撃者が行う。

Display_ECU211 は Engine_ECU321 からの CAN メッセージの受信を起点として、以下の通りエンジン状態表示パネルの更新を行う。

・CAN-ID 0x110 のデータに合せてエンジンの ON/OFF 表示を切り替える。
・Ignition=ON の時、CAN-ID 0x120 のデータに合わせて r.p.m.値の表示を更新する。
・Ignition=OFF の時、CAN-ID 0x120 のデータに関わらず、r.p.m.値の表示は常にゼロ表示とする。

Display_ECU211 は Light_ECU322 からの CAN メッセージの受信を起点として、以下の通りライト状態表示パネルの更新を行う。

・CAN-ID 0x210 のデータに合せてハザードランプの表示を点灯/消灯させる。

Attack_ECU121（攻撃 ECU）は、正規の ECU が乗っ取られ攻撃メッセージを送信するように書き換えられている。以下の(ⅰ)と(ⅱ)の攻撃手順のとおり、正規メッセージの傍受および不正メッセージの作成と送信を行う。

表 3.2d.2-2　Engine_ECU321 が送信する CAN メッセージ

CAN-ID	DF	送信周期	機能
0x110	00 00 00 00 00 00 xx 00	イベント	● エンジン操作パネルにて Ignition の ON/OFF を切り替えた場合に送信する。 ● Ignition=OFF の場合は xx=01、Ignition=ON の場合は xx=02 とする。
0x120	00 00 00 00 00 00 yy zz	500msec	● r.p.m.値を送信する。 ● r.p.m.値は送信する毎に 250（10 進数）ずつカウントアップする。具体的には最小値 00 00（10 進数で 0）、最大値 0D AC（10 進数で 3500）の範囲で yy zz がカウントアップする。

表 3.2d.2-3　Light_ECU322 が送信する CAN メッセージ

CAN-ID	DF	送信周期	機能
0x210	00 00 00 00 00 00 00 0x	100msec	● ハザードランプの状態（点灯の場合は x=1、消灯の場合は x=0）を送信する。 ● ライト操作パネルにて Hazard Flasher が押下された状態の場合、ECU322 はハザードランプの状態（点灯・消灯）を 500msec 周期で切り替える。 ● Hazard Flasher が押下されていない場合は、ハザードランプの状態は常に消灯となる。

(ｉ) バス上に流れる全ての CAN-ID のメッセージを傍受・再送する場合

まず、攻撃手順の概要は以下の通りである。

1) シミュレーション開始後、任意の時間が経過してから、攻撃 ECU は CAN バスに流れた全ての CAN-ID のメッセージを一定時間傍受し記憶する。
2) 攻撃 ECU は記憶したメッセージ群のデータの内容、順序、間隔を変更すること無く再送する。送信開始のタイミングは任意とする。
3) 攻撃 ECU は再送してから任意の時間が経過した後、傍受と記憶、再送を繰り返す。
4) 攻撃者は受信 ECU の挙動を確認する。

次に、上記をシミュレーション上で実現するための、具体的な手順を説明する。

1) シミュレーション開始後、Attack_ECU121 は正規 ECU が送信する CAN メッセージ（CAN-ID 0x110、0x120、0x210）を傍受する。

2) 傍受から任意の時間（500msec と 550msec の 2 通りとする）が経過した後、Attack_ECU121 は CAN-ID、DLC、DF を変更することなく傍受したメッセージを CAN バスに送信する。
3) メッセージ送信後は、次のメッセージの傍受し、また再送するというサイクルを繰り返す。
4) 正規 ECU の挙動を確認する。

(ⅱ) バス上に流れる特定の CAN-ID のメッセージだけを傍受・再送する場合

攻撃手順の概要は以下の通りである。

1) シミュレーション開始後、任意の時間が経過してから、攻撃 ECU は CAN バスに流れた周期型の特定の CAN-ID のメッセージを一定時間傍受し記憶する。
2) 攻撃 ECU は記憶したメッセージ群のデータの内容、順序、間隔を変更すること無く再送する。送信開始のタイミングは任意とする。
3) 攻撃 ECU は再送してから任意の時間が経過した後、傍受と記憶、再送を繰り返す。
4) 攻撃者は受信 ECU の挙動を確認する。

次に、上記をシミュレーション上で実現するための具体的な手順を説明する。

1) シミュレーション開始後、Attack_ECU121 は正規 ECU が送信する周期的な CAN メッセージ（CAN-ID 0x120）を傍受する。
2) 傍受から任意の時間（500msec と 550msec の 2 通りとする）が経過した後、Attack_ECU121 は CAN-ID、DLC、DF を変更することなく傍受したメッセージを CAN バスに送信する。
3) メッセージ送信後は、次のメッセージの傍受し、また再送するというサイクルを繰り返す。
4) 正規 ECU の挙動を確認する。

(ⅰ)、(ⅱ)の攻撃手順を実行することで、3.2d.2.(1)で述べた攻撃のバリエーションの内、以下を確認できる。まず(ⅰ)の攻撃手順にて表 3.2d.2-2 に示すイベント型の正規メッセージと周期型の正規メッセージの両方を攻撃 ECU が傍受し再送する場合を実施することで「被攻撃メッセージの周期的送信・イベント送信による違い」を確認できる。次に、上記(ⅰ)、(ⅱ)それぞれの具体的な攻撃手順 2 を行うことで、「攻撃メッセージの送信タイミング・周期による違い」のうち送信タイミングによる違いを確認できる。

なお、「乗っ取った ECU と送信する CAN-ID の組み合わせによる違い」について今回実施した攻撃手順では攻撃結果に差が現れなかった。

② なりすまし（不正メッセージ送信）

攻撃 ECU が他 ECU の送信するメッセージになりすます場合における、不正メッセージ送信を再現するためのシミュレーション環境を述べる。仮想 ECU の組み合わせは図 3.2d.2-2 と同一である。図 3.2d.2-2 の構成でなりすまし（不正メッセージ送信）のシミュレーションを行う際の各 ECU の機能を説明する。

Engine_ECU321 は、表 3.2d.2-4 に示す CAN メッセージを一定周期、またはイベント発生時に送信する。Ignition の切り替え操作はエンジン操作パネルから攻撃者が行う。

Light_ECU322 は、表 3.2d.2-5 に示す CAN メッセージを一定周期で送信する。ライトの操作はライト操作パネルから攻撃者が行う。

Display_ECU211 は Engine_ECU321 からの CAN メッセージの受信を起点として、以下の通りエンジン状態表示パネルの更新を行う。

・CAN-ID 0x110 のデータに合せてエンジンの ON/OFF 表示を切り替える。
・Ignition=ON の時、CAN-ID 0x120 のデータに合わせて r.p.m.値の表示を更新する。
・Ignition=OFF の時、CAN-ID 0x120 のデータに関わらず、r.p.m.値の表示は常にゼロ表示とする。

Display_ECU211 は Light_ECU322 からの CAN メッセージの受信を起点として、以下の通りライト状態表示パネルの更新を行う。

・CAN-ID 0x210 のデータに合せてハザードランプの表示を点灯/消灯させる。

Attack_ECU121（攻撃 ECU）は、正規の ECU が乗っ取られ攻撃メッセージを送信するように書き換えられている。そして、正規メッセージの傍受および不正メッセージの作成と送信を行う。Attack_ECU121 の動作の詳細は以下の攻撃手順の通りである。

攻撃手順の概要を示す。なお手順 2-1 と 2-2 はそれぞれ別のシミュレーションとして行う。

1) 攻撃 ECU は CAN バスを十分な時間傍受し、攻撃対象の CAN メッセージを解析し、ID、DLC、DF を特定する。
2-1) 攻撃 ECU は特定した正規の CAN-ID の各メッセージに対して、DF の変数部分を任意の不正な値に変更したメッセージを任意のタイミングで送信する。
2-2) 攻撃 ECU は正規の CAN-ID の各メッセージに対して、DF の長さを短く切り詰めるように変更することで DLC の値を変更したメッセージを任意のタイミングで送信する。
3) 攻撃 ECU は受信 ECU の挙動を確認する。

次に、図 3.2d.2-3 に攻撃者が乗っ取った ECU が元より送信する CAN-ID のメッセージを利用してなりすます場合における、不正メッセージ送信を再現するためのシミュレーション環境を示す。本環境では乗っ取られた特定の ECU は Engine_ECU321（エンジン制御 ECU）とする。図 3.2d.2-3 の環境では不正なプログラムは不正プログラム制御パネルから発動タイミングを制御可能とする。制御パネルから不正プログラムの発動指示がない場合、Engine_ECU321 は正規の ECU として振る舞う。不正プログラムは、Engine_ECU321 が本来送信するメッセージ（CAN-ID 0x110 と CAN-ID 0x120）に対し、DLC、DF の変更および送信周期、頻度の変更を行い、攻撃メッセージとして送信する。

攻撃手順の概要は以下の通りである。

1) 攻撃 ECU は CAN バスを十分な時間傍受し、攻撃対象の CAN メッセージを解析し、ID、DLC、DF を特定する。ここでは仮に CAN-ID D とする。

2-1)攻撃 ECU は乗っ取った ECU が元より送信する CAN-ID のメッセージに対して、DF の変数部分を任意の不正な値に変更したメッセージを任意のタイミングで送信する。

2-2)攻撃 ECU は乗っ取った ECU が元より送信する CAN-ID のメッセージに対して、DF の長さを短く切り詰めるように変更することで DLC の値を変更したメッセージを任意のタイミングで送信する。

3) 攻撃 ECU は受信 ECU の挙動を確認する。

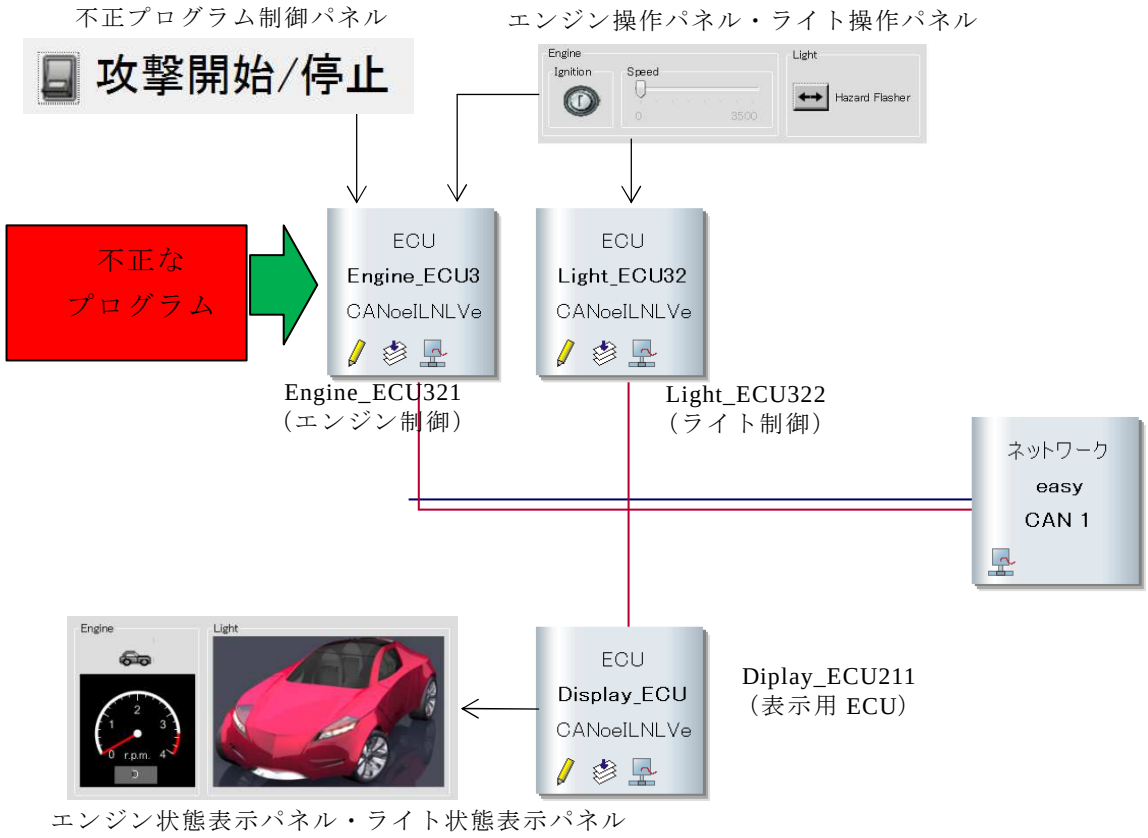


図 3.2d.2-3　不正メッセージ送信のシミュレーション環境
（攻撃 ECU が、乗っ取った ECU の CAN-ID を用いて不正送信する場合）

次に、図 3.2d.2-2 と図 3.2d.2-3 の両方の環境において上記をシミュレーション上で実現するための具体的な手順をまとめて説明する。なお、CANoe 上における仮想 ECU の実装形態が CAN の DLC に関する仕様を完全には再現していないため、一部の不正メッセージの送信および受信ノードに不正挙動を誘発させることは本シミュレーション上で完全に再現できなかった。

具体的には、①DLC だけを変更したなりすましメッセージの送信及び不正挙動の誘発に関して、DLC の値を 0 または 9 以上の値に変更することによる攻撃と被害を再現することは難しい。②DLC と DF を変更したなりすましメッセージの送信および不正挙動の誘発に関して、同様に DLC の値によって攻撃と被害を再現する事は難しい。詳細は 3.2d.2.(3)にて述べる。そのため、ここでは DF の変更のみについて、上記の攻撃手順をシミュレーシ

ョン上で実現するための手順を述べる。

1)手順 1 に関して、今回は、攻撃者は攻撃対象のメッセージ内容は既知であると仮定する。

2-1) 攻撃 ECU は表 3.2d.2-4 および表 3.2d.2-5 の各 CAN-ID のメッセージに対して、DF の変数部分（x、y、z と表記）を任意の不正な値に変更したメッセージを任意のタイミングで送信する。具体的には、CAN-ID 0x110 のメッセージについては攻撃者が xx を 0x01 と 0x02 のいずれかに設定して送信する。CAN-ID 0x120 のメッセージについては攻撃者が yy zz を 0x00 00 から 0xFF FF までのいずれかに設定して送信する。CAN-ID 0x210 のメッセージについては攻撃者が x を 0x0 か 0x1 のいずれかに設定して送信する。送信周期・タイミングは①各 CAN-ID の元の送信周期・タイミングと同等とする、②各 CAN-ID の元の送信周期より 1/2 以下に短くする、の 2 通りで実施する。

2-2)また､攻撃 ECU は表 3.2d.2-4 および表 3.2d.2-5 の各 CAN-ID のメッセージに対して、DF の長さを短く切り詰めるように変更することで DLC の値を変更したメッセージを任意のタイミングで送信する。具体的には 8 バイトメッセージ中最後のバイトを削除して 7 バイトのメッセージとして送信する。送信周期・タイミングは手順 3 と同様とする。

3) 攻撃者は、受信 ECU の挙動を確認する。

表 3.2d.2-4　Engine_ECU321 が送信する CAN メッセージ

CAN-ID	DF	送信周期	機能
0x110	00 00 00 00 00 00 xx 00	イベント	● エンジン操作パネルにて Ignition の ON/OFF を切り替えた場合に送信する。 ● Ignition=OFF の場合は xx=01、Ignition=ON の場合は xx=02 とする。
0x120	00 00 00 00 00 00 yy zz	500msec	● r.p.m.値を送信する。 ● r.p.m.値は送信する毎に 250（10 進数）カウントアップする。ただし、yy zz= 00 00～0D AC（10 進数で 0～3500）とする。

表 3.2d.2-5　Light_ECU322 が送信する CAN メッセージ

CAN-ID	DF	送信周期	機能
0x210	00 00 00 00 00 00 00 0x	100msec	● ハザードランプの状態（点灯の場合は x=1、消灯の場合は x=0）を送信する。 ● ライト操作パネルにて Hazard Flasher が押下された状態の場合、ECU322 はハザードランプの状態（点灯・消灯）を 500msec 周期で切り替える。 ● Hazard Flasher が押下されていない場合は、ハザードランプの状態は常に消灯となる。

上述の攻撃手順を実行することにより、3.2d.2.(1)で述べた攻撃のバリエーションの内、以下を確認できる。まず、図 3.2d.2-2 と図 3.2d.2-3 の両環境においてシミュレーションを実施することで、「乗っ取った ECU と送信する CAN-ID の組み合わせによる違い」を確認できる。次に、イベント型の正規メッセージと周期型の正規メッセージの双方を攻撃 ECU が不正送信することで「被攻撃メッセージの周期的送信・イベント送信による違い」を確認できる。さらに、それらの不正送信するメッセージの周期・タイミングを変更することで、「攻撃メッセージの送信タイミング・周期による違い」のうち送信タイミングによる違いを確認できる。

③ DoS（高頻度送信）

(ⅰ) 通信路全体への影響

通信路全体に対する高頻度送信を再現するためのシミュレーション環境について、正規のネットワーク構成は図 3.2d.2-2 および表 3.2d.2-2、表 3.2d.2-3 と同様である。攻撃 ECU121 は、バス上の ECU のいずれかを乗っ取ったと仮定し、DoS 攻撃を発生させるために表 3.2d.2-6 に示す攻撃用の CAN メッセージを送信するとする。送信開始・停止のタイミング、攻撃用 CAN メッセージの種類と送信周期の指定は、攻撃 ECU 制御パネルから行う。

表 3.2d.2-6　ECU121 が送信する CAN メッセージ

CAN-ID	DF	送信周期	機能
0x100	00 00 00 00 00 00 00 00	1msec ～500msec	● 指定された周期で送信を行う。 ● 各 ECU は本メッセージを受信しても何も処理を行わないが、最も若い番号の正規 CAN-ID よりも値が小さいため通信路上の調停では常に優先して送信される。

攻撃手順の概要は以下の通りである。

1) 攻撃 ECU は CAN バスを十分な時間傍受し、最も優先順位が高い CAN-ID のメッセージを特定する。ここでは仮に CAN-ID A とする。
2) 攻撃 ECU は CAN-ID 0x000 から、手順 1 で特定した CAN-ID A より小さい番号までの範囲内の CAN-ID を持つメッセージを作成する。この時、DLC の値は手順 1 で特定したメッセージと同等、DF は変更したい値とする。
3) 攻撃 ECU は攻撃メッセージを任意の周期（a_{init}）で送信する。
4) 攻撃者は、正規の ECU からのメッセージ送信が抑制されているかを確認する。また、攻撃 ECU 自身が送信する CAN-ID と異なる ID のメッセージがバス上に流れているかを確認する。
5) 攻撃 ECU 自身が送信する ID と異なる ID のメッセージが流れている場合、攻撃 ECU は攻撃メッセージの周期を徐々に小さくしていく。（最終的には、攻撃 ECU が送信可能な最短周期 a_{min} で送信する。）
6) 周期 a_{min} で送信してもなお、異なる ID のメッセージが流れている場合は、攻撃者は攻撃 ECU の数を増やして攻撃手順を繰り返す。

次に、上記をシミュレーション上で実現するための具体的な攻撃手順を説明する。

1)手順 1 に関して、今回は、攻撃者はバス上に流れるメッセージの中で最も優先度の低いメッセージの内容は既知であると仮定する。
2)攻撃 ECU は、表 3.2d.2-6 に記載のあるメッセージを攻撃用メッセージとして定める。
3)攻撃 ECU は、上記攻撃用メッセージを 500msec（= a_{init}）間隔で送信する。
4)攻撃者は Engine ECU、Light ECU、Display ECU の挙動を確認し、それぞれの ECU からのメッセージが遅延またはメッセージ送信が抑制されているかを確認する。
5)通信路に CAN-ID 0x100 以外のメッセージが流れている場合は、攻撃メッセージの送信周期を徐々に短くする。CANoe では ECU が送信可能な最短周期は 1msec（= a_{min}）であるため、遅延や停止が確認できなければ最終的に周期を 1msec として送信する。
6)周期を 1msec としてもメッセージ送信が抑制されない場合は攻撃 ECU の数を増やして攻撃手順を繰り返す。

(ⅱ) 特定 ECU への影響

特定の ECU に対する高頻度送信を再現するためのシミュレーション環境について、正規のネットワーク構成は図 3.2d.2-2 および表 3.2d.2-2、表 3.2d.2-3 と同様である。攻撃 ECU121 は、バス上の ECU のどれかが乗っ取られたと仮定し、DoS 攻撃を発生させるために、表 3.2d.2-7 に示す攻撃用の CAN メッセージを送信するものとする。送信開始・停止、送信する CAN メッセージの種類と送信周期の指定は、攻撃 ECU 制御パネルから行う。

表 3.2d.2-7　攻撃 ECU（ECU121）が送信する CAN メッセージ

CAN-ID	DF	送信周期	機能
0x120	00 00 00 00 00 00 00 00	1msec ～500msec	● 指定された周期で送信を行う。 ● 不正な r.p.m.値を送信することで、受信 ECU（ECU221）に影響を与える。

攻撃手順の概要は以下の通りである。

1) 攻撃 ECU は CAN バスを十分な時間傍受し、攻撃対象のメッセージの CAN-ID、DLC,DF を特定する。

2)攻撃用メッセージの CAN-ID B とする。DLC と DF は元のメッセージの中から利用された値を任意に設定する。

3)攻撃 ECU は攻撃用メッセージを任意の周期（b_{init}）で送信する。

4)攻撃者は攻撃対象の ID によるメッセージの正規の送信 ECU からの発信が抑制されているかどうか確認する。

5)正規メッセージの送信が抑制されていない場合は、周期を徐々に小さくしていき、最終的に最短周期 b_{min} で送信する。

6)周期 b_{min} で送信してもなお正規メッセージの発信が抑制されていない場合は、攻撃者は攻撃 ECU の数を増やして攻撃手順を繰り返す。

次に、当該シミュレーション環境下における具体的な攻撃手順を説明する。

1)手順 1 に関して、攻撃者は攻撃対象のメッセージの CAN-ID、DLC、DF が既知であると仮定する。

2)攻撃 ECU は、表 3.2.d.2-6 に記載のあるメッセージを攻撃用メッセージとして定める。

3)攻撃 ECU は、上記攻撃用メッセージを 500msec（=b_{init}）間隔で送信する。

4)攻撃者は、攻撃対象の ID によって動作する特定 ECU の挙動を確認し、特定 ECU からのメッセージが遅延またはメッセージが抑制されているかを確認する。

5)通信路に攻撃対象の ID のメッセージが流れている場合は、攻撃メッセージの送信周期を徐々に短くする。CANoe では ECU が送信可能な最短周期は 1msec（=b_{min}）間隔であるため、遅延や停止が確認できなければ最終的に周期を 1msec 間隔として送信する。

6)周期を 1msec としてもメッセージ送信が抑制されない場合は、攻撃 ECU の数を増やして上記の攻撃手順を繰り返す。

上述の攻撃手順を実行することにより、(ⅰ)と(ⅱ)のそれぞれに対して、3.2d.2.(1)で述べた攻撃のバリエーションの内、以下を確認できる。まず「乗っ取った ECU と送信する CAN-ID の組み合わせによる違い」について、使われている CAN-ID よりも小さい CAN-ID 0x100 と使われている中で最も小さい CAN-ID 0x120 を攻撃メッセージとして用いたことで CAN-ID の違いを確認できる。また、上記手順の通り攻撃 ECU の送信周期を徐々に変化させることで「攻撃メッセージの送信タイミング・周期による違い」のうち送信タイミン

グによる違いを確認できる。

④ 正規メッセージとの衝突メッセージ送信（バスオフを誘発）

図 3.2d.2-4 に正規メッセージとの衝突メッセージ送信を再現するためのシミュレーション環境を示す。当該環境の実バス部分の機器構成を示している。本構成に CANoe の実バスインタフェース（VN1630）を接続し CAN バスに流れた通信ログも取得してシミュレーションを実施した。

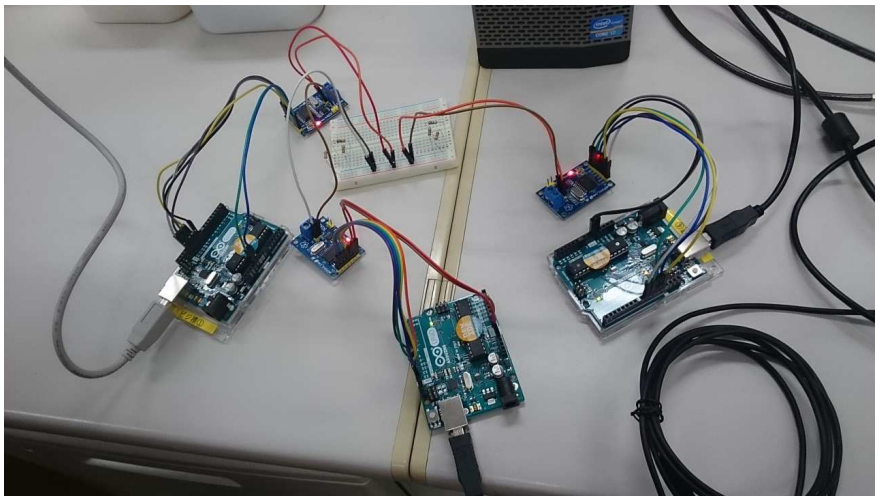


図 3.2d.2-4　正規メッセージとの衝突メッセージ送信における実バス環境

ECU A は送信 ECU とする。10msec 周期で表 3.2.d.2-8 に示す ID 0x007 と ID 0x009 のメッセージを連続して送信する。

ECU B は送信 ECU かつ被攻撃者とする。ECU A が送信した ID 0x007 と 0x009 のメッセージの直後に表 3.2.d.2-9 に示すメッセージを送信する。ID 0x009 と本メッセージの間は CAN バスがアイドル状態とならないようにする。

ECU C は攻撃 ECU とする。ECU A が送信した ID 0x007 と 0x009 のメッセージの直後に表 3.2.d.2-10 に示すメッセージを送信する。ID 0x009 と本メッセージの間は CAN バスがアイドル状態とならないようにする。

表 3.2.d.2-8　ECU A が送信する CAN メッセージ

CAN-ID	DF	送信周期	機能
0x007	00 00 00 00 00 00 00 00 （DLC=8）	10msec	● ID 0x009 のメッセージの直前に送信され、2 つのメッセージが CAN バス上を連続して流れる状態を一定周期で再現する。
0x009	00 00 00 00 00 00 00 00 （DLC=8）	10msec	● ID 0x007 のメッセージの直後に送信され、2 つのメッセージが CAN バス上を連続して流れる状態を一定周期で再現する。

表 3.2.d.2-9　ECU B が送信する CAN メッセージ

CAN-ID	DF	送信周期	機能
0x011	00 00 00 00 00 00 00 00 （DLC=8）	10msec	● 被攻撃メッセージ。ノード A が送信する ID 0x007 と 0x009 のメッセージの直後に CAN バス上に送信される。

表 3.2.d.2-10　ECU C が送信する CAN メッセージ

CAN-ID	DF	送信周期	機能
0x011	なし （DLC=0）	10msec	● 攻撃メッセージ。ノード A が送信する ID 0x007 と 0x009 のメッセージの直後に CAN バス上に送信される。

3.2d.1.(3)で言及した通り、攻撃メッセージの送信タイミング・周期の違いから以下の 2 通りの攻撃手順を示す。攻撃手順(ⅰ)は正規メッセージの送信周期に合わせて攻撃メッセージを送信する場合の手順であり、攻撃手順(ⅱ)は正規メッセージの送信周期に依らず、攻撃メッセージを高頻度に送信する場合の手順である。

(ⅰ) 正規メッセージとの同期送信による、正規メッセージとの衝突メッセージ送信（バスオフを誘発）

攻撃手順の概要は以下の通りである。

1) 攻撃 ECU はバスを傍受して、攻撃対象の ECU の CAN-ID や DLC、DF、そして送信周期や送信間隔を割り出す。なおここでは仮に攻撃対象のメッセージの CAN-ID を E とし、E の前に送信される特定のメッセージの CAN-ID を E-、E--、...とする。また、CAN-ID E のメッセージ送信周期、あるいは特定のメッセージを受信してから送信するまでに決められた時間 t 秒とする。
2) 攻撃 ECU は CAN-ID E を攻撃に使うメッセージの CAN-ID とする。DLC の値は 0 から使われた最小の値（ここでは DLC_{max} とする）までのいずれかに設定する。DF は DLC の設定値に依存して変わる。
   ・DLC を 0 とする場合は、DF は null（値無し）とする。
   ・DLC を 1 から DLC_{max}-1 の範囲のいずれかとする場合は、DF は任意の値とする。
   ・DLC を DLC_{max} とする場合は、DF は DLC_{max} バイト連続する 0 か、傍受した最小の値以下とする。
3) 攻撃 ECU は CAN-ID E を攻撃に使うメッセージの CAN-ID とする。
4) 攻撃 ECU は i 番目の CAN-ID E のメッセージを受信したら、t 秒毎に攻撃メッセージを送信する。
5) 攻撃者は i+1 番目の CAN-ID E のメッセージと攻撃メッセージとが衝突したかどうかを、CAN-ID E を送信する正規の ECU の送信エラーカウンタが上昇しているかによって逐次確認する。
6) 攻撃者は正規の ECU の送信エラーカウンタが上限値 255 に到達するか、あるいは十

分な時間が経過するまで、i+2、i+3、...番目のメッセージに対して攻撃手順を繰り返す。

次に、当該シミュレーション環境下における具体的な攻撃手順を説明する。

1) 今回実施したシミュレーションでは傍受と割り出しの手順は省略し、攻撃 ECU には既知の情報とした。
2) 攻撃 ECU は、表 3.2.d.2-10 に記載のあるメッセージを攻撃メッセージとして定める。
3) 攻撃 ECU は、表 3.2.d.2-8 に記載のある CAN-ID のメッセージを受信した直後に、10msec 間隔で攻撃メッセージを送信する。
4) 攻撃者は、攻撃メッセージが正規メッセージと衝突したことを、ECU A の送信エラーカウンタが上昇していることで確認する。
5) 攻撃者は、ECU A の送信エラーカウンタが上限値 255 に到達するまで、攻撃手順を繰り返す。

(i) (ⅱ)正規メッセージとの非同期送信による、正規メッセージとの衝突メッセージ送信（バスオフを誘発）

ECU A は送信 ECU かつ被攻撃者とする。10msec 周期で、表 3.2.d.2-11 に示す ID 0x011 のメッセージを連続して送信する。

ECU B は攻撃 ECU とする。ECU A が送信した ID 0x011 のメッセージの直後に表 3.2.d.2-12 に示すメッセージを送信する。ID 0x011 と本メッセージの間には CAN バスがアイドル状態とならないようにする。

表 3.2.d.2-11　ECU A が送信する CAN メッセージ

CAN-ID	DF	送信周期	機能
0x011	00 00 00 00 00 00 00 01 （DLC=8）	10msec	● 被攻撃メッセージ。

表 3.2.d.2-12　ECU B が送信する CAN メッセージ

CAN-ID	DF	送信周期	機能
0x011	00 00 00 00 00 00 00 00 （DLC=8）	<1msec	● 攻撃メッセージ。

攻撃手順の概要は以下の通りである。

1) 攻撃 ECU はバスを傍受して、攻撃対象の ECU の CAN-ID や DLC、DF、そして送信周期や送信間隔を割り出す。
2) 攻撃 ECU は割り出した CAN-ID を攻撃に使う送信メッセージの CAN-ID とする。
3) 攻撃 ECU は割り出した CAN-ID の i 番目のメッセージを受信したら、攻撃メッセージを連続して
   t 秒経過まで隙間無く送信し続ける。
4) 攻撃者は割り出した CAN-ID の i+1 番目のメッセージと攻撃メッセージとが衝突した

かどうかを、攻撃対象のの ECU の送信エラーカウンタが上昇したかどうかから逐次確認する。

5) 攻撃者は正規の ECU エラーカウンタが上限値 255 に到達するか、あるいは十分な時間が経過するまで、i+2、i+3、…番目のメッセージに対して攻撃手順を繰り返す。

次に、当該シミュレーション環境下における具体的な攻撃手順を説明する。

1) 今回実施したシミュレーションでは傍受と割り出しの手順は省略し、攻撃 ECU には既知の情報とした。
2) 攻撃 ECU は、表 3.2.d.2-12 に記載のあるメッセージを攻撃メッセージとして定める。
3) 攻撃 ECU は、表 3.2.d.2-11 に記載のある CAN-ID のメッセージを受信した直後に、マイコンおよび CAN コントローラの性能の範囲内で可能な限り短周期で 1000msec 後まで送信し続ける。
4) 攻撃者は、攻撃メッセージが正規メッセージと衝突したことを、ECU A のエラーカウンタが上昇していることから確認する。
5) 攻撃者は、ECU A のエラーカウンタが上限値 255 に到達するまで、攻撃手順を繰り返す。

⑤ 診断用メッセージ送信

図 3.2.d.2-5 に診断用メッセージ送信を再現するためのシミュレーション環境を示す。

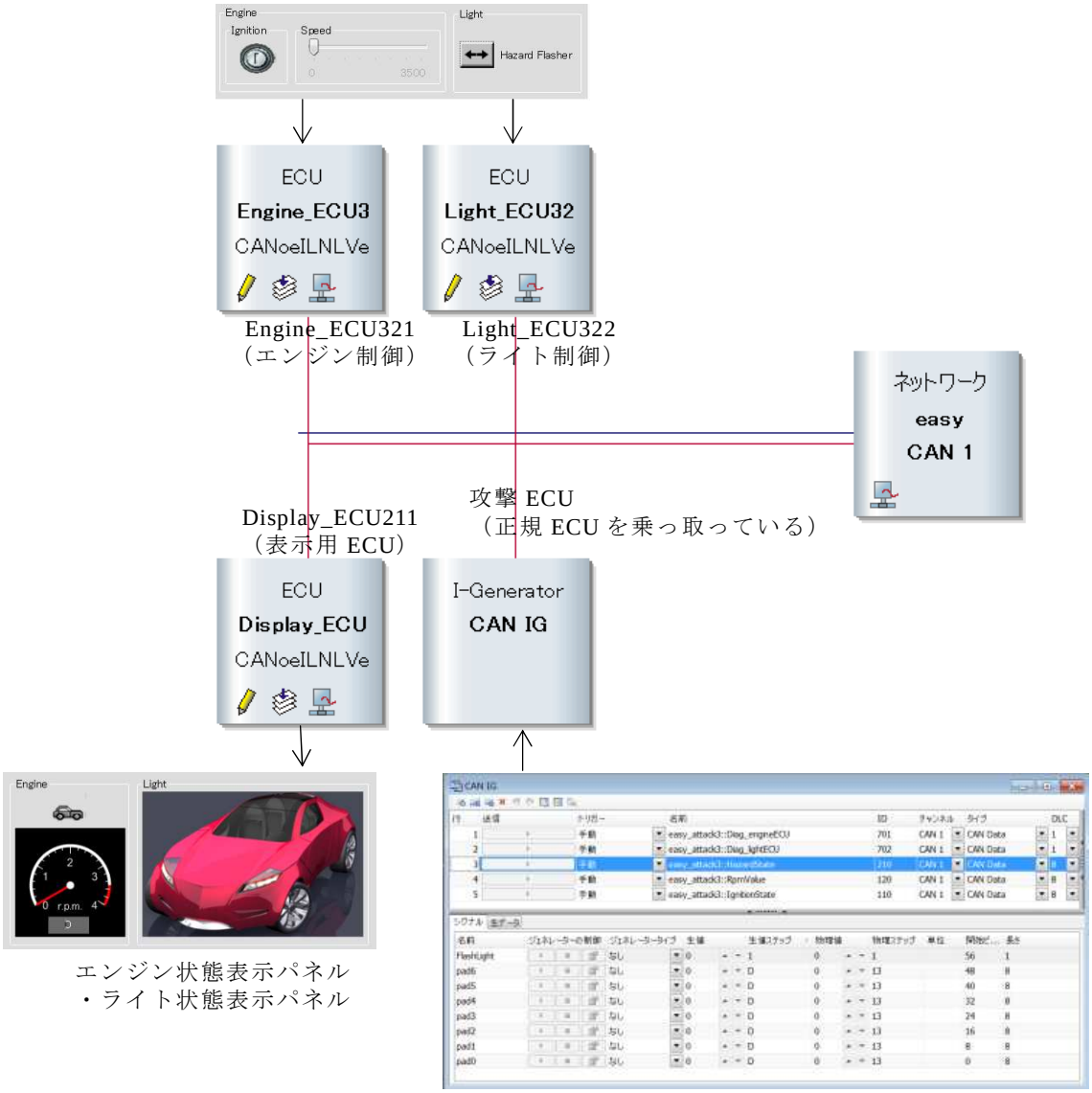


図 3.2.d.2-5　診断用メッセージ送信のシミュレーション環境

それぞれの ECU の機能について説明する。

Engine_ECU321 は表 3.2.d.2-13 に示す CAN メッセージを一定の送信周期、またはイベント発生時に送信する。Ignition の切り替え操作はエンジン操作パネルから攻撃者が行う。Engine_ECU321 は、CAN-ID 0x701 のメッセージを受信した場合、その DF の値に応じて自身の状態を通常モード/診断モードに切り替える。診断モードの場合、Engine_ECU321 は表 3.2.d.2-13 に示すメッセージの送信を行わない。

Light_ECU322 は表 3.2.d.2-14 に示す CAN メッセージを一定の周期で送信する。ライトの操作はライト操作パネルから攻撃者が行う。Light_ECU322 は、CAN-ID 0x702 のメッセージを受信した場合、その DF の値に応じて自身の状態を通常モード/診断モードに切り替える。診断モードの場合、Light_ECU322 は表 3.2.d.2-14 に示すメッセージの送信を行わない。

Display_ECU211 は Engine_ECU321 からの CAN メッセージの受信をトリガーとして、エンジン状態表示パネルの更新を行う。

・CAN-ID 0x110 のデータに合せてエンジンの ON/OFF 表示を切り替える。

・Ignition=OFF の時、CAN-ID 0x120 のデータに関わらず、r.p.m.値の表示は常にゼロ表示とする。

・Ignition=ON の時、CAN-ID 0x120 のデータに合わせて r.p.m.値の表示を更新する。

攻撃 ECU は表 3.2.d.2-15 に示すメッセージをイベント送信する。送信するメッセージとタイミングの指定は、攻撃者が攻撃 ECU の制御ウィンドウから行う。

表 3.2.d.2-13　Engine_ECU321 が送信する CAN メッセージ

CAN-ID	DF	送信周期	機能
0x110	00 00 00 00 00 00 xx 00	イベント	● エンジン操作パネルにて Ignition の ON/OFF を切り替えた場合に送信する。 ● Ignition=OFF の場合は xx=01、Ignition=ON の場合は xx=02 とする。
0x120	00 00 00 00 00 00 yy zz	500msec	● r.p.m.値を送信する。 ● r.p.m.値は送信する毎に 250（10 進数）カウントアップする。ただし、yy zz= 00 00～0D AC（10 進数で 0～3500）とする。

表 3.2.d.2-14　Light_ECU322 が送信する CAN メッセージ

CAN-ID	DF	送信周期	機能
0x210	00 00 00 00 00 00 00 0x	100msec	● ハザードランプの状態（点灯の場合は x=1、消灯の場合は x=0）を送信する。 ● ライト操作パネルにて Hazard Flasher が押下された状態の場合、ECU322 はハザードランプの状態（点灯・消灯）を 500msec 周期で切り替える。 ● Hazard Flasher が押下されていない場合は、ハザードランプの状態は常に消灯となる。

表 3.2.d.2-15　攻撃 ECU が送信する CAN メッセージ

CAN-ID	DF	送信周期	機能
0x701	xx	イベント	● Engine_ECU321 の通常モード/診断モードを切り替える。通常モードにしたい場合は xx=00、診断モードにしたい場合は xx=01 を指定する。
0x702	xx	イベント	● Light_ECU322 の通常モード/診断モードを切り替える。通常モードにしたい場合は xx=00、診断モードにしたい場合は xx=01 を指定する。
0x120	00 00 00 00 00 00 yy zz	イベント	● Display_ECU211 に表示する r.p.m. 値を指示する。r.p.m.値は yy zz= 00 00～0D AC（10 進数で 0～3500）の範囲で指定する。
0x210	00 00 00 00 00 00 00 0x	イベント	● Display_ECU211 にハザードランプの点灯・消灯を指示する。点灯の場合は x=1、消灯の場合は x=0 を指定する。

攻撃手順の概要は以下の通りである。

1) 攻撃 ECU は CAN バスを十分な時間傍受し、CAN バスに流れた診断用メッセージの CAN-ID、DLC および DF を特定する。ここでは仮に CAN-ID G とする。CAN-ID G のメッセージが複数種類の DLC または DF を有していた場合は各メッセージの送信間隔も割り出す。
2) 攻撃 ECU は CAN-ID G を攻撃用メッセージの ID とする。DLC と DF は、正規メッセージで利用された値をそのまま設定する。CAN-ID G のメッセージに複数の DLC または DF が存在する場合はそのすべてを攻撃メッセージとして設定する。
3) 攻撃 ECU は攻撃メッセージを前述の送信間隔に従い順に送信する。攻撃メッセージ送信の開始タイミングは任意とする。
4) 正規 ECU 正常な ECU である ECU321 または ECU322 が CAN-ID G の診断用メッセージを受信し、診断モードに移行し、メッセージの発信を抑制したかどうかを攻撃者は確認する。

次に、上記をシミュレーション上で実現するための具体的な攻撃手順を説明する。

また､本シミュレーションでは正規の ECU が診断モードに移行したかどうかを以下の方法で確認することとした。具体的には、診断モードに移行したかどうかをチェックするメッセージを送信後、攻撃者は正規 ECU になりすましたメッセージ（CAN-ID は同じで、DF の値が異なる）を送信して、攻撃 ECU が送信したメッセージだけが受信 ECU に処理されていることを確認した。

1) 今回実施したシミュレーションでは傍受と診断モードに移行する特定 CAN-ID の割り出しの手順は省略し、攻撃 ECU には既知の情報とした。
2) 現時点で診断モードに移行していないことを確認するために、攻撃 ECU 操作パネル

からCAN-ID 0x120のメッセージを不正送信し挙動を確認する。このとき、DFのr.p.m.値は任意の値とする。

3) 攻撃ECU操作パネルからCAN-ID 0x701のメッセージ（DFはxx=01）をイベント送信する。
4) エンジン状態表示パネルの更新が停止した後、再度攻撃ECU操作パネルから任意のr.p.m.値を設定したCAN-ID 0x120のメッセージを不正送信し、エンジン状態表示パネルの挙動を確認する。

上記手順をCAN-ID 0x702と0x210の組み合わせについても同様に実施する。

⑥ 不正メッセージ送信

(ⅰ) 不正DLCメッセージ送信

本攻撃方法によるシミュレーションは、上述の④(ⅰ)特定ECUへの高頻度送信によるシミュレーションと同一環境、同一手順で実施可能であるため、④(ⅰ)を参照のこと。

(ⅱ) 不正DFメッセージ送信

本攻撃方法によるシミュレーションは、上述の④(ⅰ)特定ECUへの高頻度送信によるシミュレーションと同一環境、同一手順で実施可能であるため、④(ⅰ)を参照のこと。

(2) シミュレーション結果

それぞれの攻撃方法の攻撃手順に従い、CANoeを用いて攻撃方法の再現を試みた結果を示す。

① なりすまし（リプレイ送信）

全CAN-IDのリプレイ送信と特定CAN-IDのリプレイ送信のシミュレーションに要した時間とCANメッセージ総数をそれぞれ表3.2.d.2-16と表3.2.d.2-17に示す。CANメッセージ総数のうち、表3.2.d.2-16の場合は正規の送信メッセージは約50%、攻撃メッセージは約50%であった。他方、表3.2.d.2-17の場合は正規の送信メッセージは約78%、攻撃メッセージは約22%であった。

表3.2.d.2-16　リプレイ送信のシミュレーション時間CANメッセージ総数

（全CAN-IDの再送）

No.	項目	通信速度	シミュレーション結果
1	シミュレーション時間	500kbps	107秒
		125kbps	79秒
2	CANメッセージ数（総数）	500kbps	1452フレーム
		125kbps	1158フレーム

表 3.2.d.2-17　リプレイ送信のシミュレーション時間と CAN メッセージ総数

（単一 CAN-ID の再送）

No.	項目	通信速度	シミュレーション結果
1	シミュレーション時間	500kbps	103 秒
		125kbps	82 秒
2	CAN メッセージ数（総数）	500kbps	927 フレーム
		125kbps	662 フレーム

攻撃 ECU が CAN バスを流れる 3 種類のメッセージ（CAN-ID 0x110 と 0x120 と 0x210）に対してリプレイ送信を行った結果、以下の現象を確認した。

1) エンジン状態表示パネルの r.p.m.値の更新周期が不規則になった。これは、エンジン制御 ECU が周期的に送信している r.p.m.値更新用のメッセージ（CAN-ID 0x120）を攻撃 ECU がリプレイ送信することにより、攻撃を受けた表示用 ECU が古いメッセージのデータを表示したためである。

2)ライト状態表示パネルのハザードライトの点滅周期が不規則になった。これは、ライト制御 ECU が周期的に送信しているハザードランプ更新用のメッセージ（CAN-ID 0x210）を攻撃 ECU がリプレイ攻撃することにより、攻撃を受けた表示用 ECU が古いメッセージのデータ（点灯、または消灯）を表示したためである。

3)エンジン操作パネルの Ignition を ON の状態から素早く OFF→ON（または OFF の状態から素早く ON→OFF）することで、エンジン操作パネルの Ignition の状態とエンジン状態表示パネルの走行状態アイコンに不整合が生じた。これは、エンジン操作パネルの Ignition を ON の状態から素早く OFF→ON した場合、一度は操作パネルの Ignition が ON、表示パネルの走行状態アイコンが走行中（Running）となり整合が取れるが、その後、攻撃 ECU が Ignition を OFF にした時のメッセージ（CAN-ID 0x110）でリプレイ攻撃を行うため、表示パネルの走行状態アイコンが停止中（表示なし）となり不整合が発生したためである。

以上のとおり、正規メッセージが散発的あるいは周期的に流れ続けている状態において、リプレイ送信による攻撃方法を再現できた。

② なりすまし（不正メッセージ送信）

(ⅰ) 不正 DLC 送信

DLC を 0（本来の DLC とは異なる値）に変更した場合：

本シミュレーションに要した時間と CAN メッセージ総数を表 3.2.d.2-18 に示す。CAN メッセージ総数のうち、正規の送信メッセージは約 50%、攻撃メッセージは約 50%であった。

表 3.2.d.2-18　不正 DLC 送信のシミュレーション時間と CAN メッセージ総数

（DLC=0x0 の場合）

No.	項目	通信速度	シミュレーション結果
1	シミュレーション時間	500kbps	59 秒
		125kbps	55 秒
2	CAN メッセージ数（総数）	500kbps	872 フレーム
		125kbps	796 フレーム

CAN バスを流れる 3 種類のメッセージ（CAN-ID 0x110 と 0x120 と 0x210）に対して不正な DLC に変更したメッセージの送信 （DLC を 0x0 に変更）を行った結果、以下の現象が確認できた。

1) CANoe の通信ログ上では DLC が 0x0 と出力された。DLC が 0x0 のため DF は出力されなかった。
2) r.p.m.値の更新周期およびハザードランプの点滅周期が不規則になった。また、エンジン操作パネルの Ignition を ON の状態から素早く OFF→ON（または OFF の状態から素早く ON→OFF）することで、エンジン操作パネルの Ignition の状態とエンジン状態表示パネルの走行状態アイコンに不整合が生じた。これは、リプレイ送信の結果と同様の結果であり、表示用 ECU が DLC を 0x0 に変更されたメッセージを受信したにも関わらず、受信 ECU が不正メッセージの DF を正しく解釈できていることを意味している。

・DLC を 0xF（規定外の値）にした場合：

本シミュレーションに要した時間と CAN メッセージ総数を表 3.2.d.2-19 に示す。CAN メッセージ総数のうち、正規の送信メッセージは約 50%、攻撃メッセージは約 50%であった。

表 3.2.d.2-19　不正 DLC 送信のシミュレーション時間と CAN メッセージ総数

（DLC=0xF の場合）

No.	項目	通信速度	シミュレーション結果
1	シミュレーション時間	500kbps	56 秒
		125kbps	41 秒
2	CAN メッセージ数（総数）	500kbps	803 フレーム
		125kbps	571 フレーム

CAN バスを流れる 3 種類のメッセージ（CAN-ID 0x110 と 0x120 と 0x210）に対して不正な DLC に変更したメッセージ送信 （DLC を 0xF に変更）を行った結果、以下の現象が確認できた。

1) CANoe の通信ログ上では DLC が 0xF と出力された。この時、DLC は 0xF と出力されているが、DF は正規の 8 バイト分のみ出力された。
2) r.p.m.値の更新周期およびハザードランプの点滅周期が不規則になった。また、エンジン操作パネルの Ignition を ON の状態から素早く OFF→ON（または OFF の状態から素早く ON→OFF）することで、エンジン操作パネルの Ignition の状態とエンジン状態表示パネルの走行状態アイコンに不整合が生じた。これは、リプレイ送信の結果と同様の結果であり、表示用 ECU が DLC を 0xF に変更されたメッセージを受信したにも関わらず、受信 ECU が不正メッセージの DF を正しく解釈できていることを意味している。

上記の結果から、DLC が 0x0 に変更されていても、シミュレータ上の仮想 ECU は 8 バイトの DF であると解釈していると推測される。一方で、CANoe の挙動を確認するため DLC を 0x1 から 0x8 の間に設定したところ、仮想 ECU は DF の長さを 1 バイトから 8 バイトの間で正しく解釈して送受信した。このように、CANoe では DLC を任意の値に設定しても、受信 ECU（仮想 ECU）がそれを正しいメッセージとして解釈している現象が見られた。一方、実車に搭載されている ECU（実 ECU）では、DLC と実際の DF の長さが異なる場合には、該当メッセージを破棄する処理が CAN コントローラで行われると推測する。よって、本シミュレーション環境では実 ECU の動作を完全に模擬することは難しく、特に DLC の最小値（0x0）への変更による攻撃の被害をシミュレーションにより完全に再現することは難しいと考える。

(ⅱ) 不正 DF 送信

正規メッセージの DF を変更した不正 DF 送信を再現した。

攻撃 ECU が他 ECU の送信する ID を含むメッセージを傍受し、該当メッセージの DF を変更して送信する場合のシミュレーションに要した時間と CAN メッセージ総数を表 3.2.d.2-20 に示す。CAN メッセージ総数のうち、正規の送信メッセージは約 50%、攻撃メッセージは約 50%であった。

表 3.2.d.2-20　不正 DF 送信のシミュレーション時間と CAN メッセージ総数
（他 ECU の CAN-ID メッセージのなりすまし）

No.	項目	通信速度	シミュレーション結果
1	シミュレーション時間	500kbps	48 秒
		125kbps	72 秒
2	CAN メッセージ数（総数）	500kbps	711 フレーム
		125kbps	1043 フレーム

他方、攻撃 ECU が自 ECU の送信する ID のメッセージの DF を変更して送信する場合のシミュレーションに要した時間と CAN メッセージ総数を表 3.2.d.2-21 に示す。CAN メッセージ総数のうち、正規の送信メッセージは約 27%、攻撃メッセージは約 73%であった。

表 3.2.d.2-21　不正 DF 送信のシミュレーション時間と CAN メッセージ総数
（自 ECU の CAN-ID メッセージのなりすまし）

No.	項目	通信速度	シミュレーション結果
1	シミュレーション時間	500kbps	43 秒
		125kbps	58 秒
2	CAN メッセージ数（総数）	500kbps	504 フレーム
		125kbps	697 フレーム

3 種類のメッセージ（CAN-ID 0x110 と 0x120 と 0x210）に対して不正な DF に変更したメッセージの送信を行った結果、以下の現象が確認できた。

1) エンジン状態表示パネルの r.p.m.値が正常値と 9999 を交互に示した(エラー処理として、表示用 ECU が不正値を受信した場合に r.p.m.値として 9999 かつメータの上限を表示するように設定した)。これは、エンジン制御 ECU が周期的に送信している r.p.m.値更新用のメッセージ（CAN-ID 0x120）の DF を全て 0xFF に変更したメッセージを攻撃 ECU が送信することにより、それを受信した表示用 ECU がエラー処理を行ったためである。
2) ライト状態表示パネルのハザードライト表示の点滅周期が不規則になった。これは、ライト制御 ECU が周期的に送信しているハザードランプ更新用のメッセージ（CAN-ID 0x210）の DF を全て 0xFF に変更したメッセージを攻撃 ECU が送信することにより発生している。CAN-ID 0x210 のメッセージの DF を 0xFF に変更するということは、ハザードライトの点灯/消灯を示すビットが 1（点灯）となることを示す。よって、ハザードライトが消灯しているタイミングで、表示用 ECU が変更されたメッセージを受信すると、通常の周期を逸脱してハザードライトを点灯させてしまう。
3) エンジン操作パネルの Ignition を操作すると、エンジン状態表示パネルの走行状態アイコンは正しい表示をしたが、それとは別にエラーログが出力された。これは、Ignition 操作時にエンジン制御 ECU が送信するメッセージ（CAN-ID 0x110）の DF を全て 0xFF に変更したものを攻撃 ECU が送信したためである。CAN-ID 0x110 のメッセージの DF を 0xFF に変更するということは、Ignition 状態（ON/OFF）を示す 2bit のデータが 11（10 進数で 3）となり不正値となる。よって、これを受信した表示用 ECU はエラーログを出力する。
4) 攻撃 ECU が自 ECU の送信する ID のメッセージの DF を変更して送信した場合において、攻撃 ECU にメッセージ（CAN-ID 0x120）の DF を全て 0x00 に変更する不正プログラムを埋め込んだところ、不正プログラムが動作していない時はエンジン状態表示パネルの r.p.m.値は周期的にカウントアップされたが、不正プログラムを発動すると、r.p.m.値は常に 0 を示し、不正な DF のメッセージになりすましされていることが確認できた。

以上の通り、連続する複数メッセージの不正 DF 送信の再現を確認できた。

③ DoS（高頻度送信）

(ⅰ) 通信路全体

本シミュレーションに要した時間と CAN メッセージ総数を表 3.2.d.2-22 に示す。CAN メッセージ総数のうち、正規の送信メッセージはイベント送信、周期的送信の両方を合わせて約 8%、攻撃メッセージは約 92%であった。

表 3.2.d.2-22 通信路全体に対する高頻度送信のシミュレーション時間と CAN メッセージ総数

No.	項目	通信速度	シミュレーション結果
1	シミュレーション時間	500kbps	344 秒
		125kbps	370 秒
2	CAN メッセージ数（総数）	500kbps	41008 フレーム
		125kbps	46097 フレーム

攻撃 ECU が CAN バスを占有するメッセージを送信した結果､以下の現象が確認できた。

1）最も早い送信周期の正規メッセージよりも攻撃メッセージの送信周期が遅い場合（シミュレーションでは攻撃メッセージの送信周期が 100msec 以上の場合）は表示 ECU の表示は正規値と攻撃メッセージによる不正値を交互に表示し、攻撃メッセージの送信周期が早まるにつれ徐々に不正値を表示する時間が長くなり、最短送信周期である 1 msec になったとき最終的に常に不正値を表示するようになった。よって、1 種類の不正メッセージにより、バス上に接続される全ての ECU が発信するメッセージを抑制することが可能であり､バス上の任意の ECU に影響を及ぼす DoS 攻撃を再現することができた。

(ⅱ) 特定 ECU

特定 ECU に対する DoS 攻撃を再現した。本シミュレーションに要した時間と CAN メッセージ総数を表 3.2.d.2-23 に示す。CAN メッセージ総数のうち、正規の送信メッセージはイベント送信、周期的送信の両方を合わせて約 8%、攻撃メッセージは約 92%であった。

表 3.2.d.2-23 特定 ECU に対する高頻度送信のシミュレーション時間と CAN メッセージ総数

No.	項目	通信速度	シミュレーション結果
1	シミュレーション時間	500kbps	396 秒
		125kbps	325 秒
2	CAN メッセージ数（総数）	500kbps	67662 フレーム
		125kbps	181823 フレーム

攻撃 ECU が CAN バスを占有するメッセージを送信した結果、以下の現象が確認できた。

1) DoS メッセージの送信周期を 1msec とした時、CANoe の統計ウィンドウの占有率が 100%となり、エンジン状態表示パネルの r.p.m.値は正規の周期で更新されているにも関わらず、警告表示パネルの更新はほぼ停止した。これは、占有率が 100%となったことにより、優先度の低いメッセージ（CAN-ID 0x310）の送信が滞っているためである。
2)しかし、占有率が 100%となっても CAN-ID 0x310 のメッセージも送信されていることが確認できている。これは、CAN-ID 0x120 のメッセージのみで占有率が 100%となっているわけではなく、わずかに CAN バスが空いたタイミングで、CAN-ID 0x310 のメッセージが送信できているためである。
3)本攻撃により占有率が 100%になった場合、エンジン状態表示パネルは正規の周期で更新されることを目視により確認できている点については前述した通りである。しかし通信ログを確認すると、Display_ECU211 は新しい r.p.m.値（古い r.p.m.値から 250 カウントアップされた値）の設定されたメッセージ（CAN-ID 0x120）を受信後、古い r.p.m.値の設定されたメッセージ（CAN-ID 0x120）を数回受信している。これは、Engine_ECU321 が新しい r.p.m.値の設定されたメッセージを送信中、つまり Attack_ECU121 が新しい r.p.m.値を傍受する前に、Attack_ECU121 が古い r.p.m.値の設定されたメッセージの送信処理を行っているためである。この Attack_ECU121 からのメッセージは Engine_ECU321 からの新しい r.p.m.値の設定されたメッセージの送信完了を待ってから CAN バスに流れるため、Display_ECU211 は、新しい r.p.m.値を受信後に、古い r.p.m.値を受信することになってしまう。しかし、Attack_ECU121 はすぐに新しい r.p.m.値の設定されたメッセージの送信を開始するため、目視上、エンジン状態表示パネルの表示更新は正常に行われる。

以上の通り、バス上のある特定の ECU が発するメッセージを抑制できたことが確認されたため、特定 ECU に影響を及ぼす DoS 攻撃を再現できた。

④ DoS（正規メッセージとの衝突メッセージ送信（バスオフ誘発））

正規メッセージとの衝突メッセージ送信による攻撃方法を再現した。本シミュレーションに要した時間と CAN メッセージ総数を表 3.2.d.2-24 と表 3.2.d.2-25 に示す。表 3.2.d.2-24 は正規メッセージとの同期送信による衝突を行った際の結果である。CAN メッセージ総数のうち、正規の送信メッセージは約 67%、攻撃メッセージは約 33%であった。一方、表 3.2.d.2-25 は正規メッセージとの非同期送信による衝突を行った際の結果である。CAN メッセージ総数のうち、正規の送信メッセージは約 12%、攻撃メッセージは約 88%であった。

表 3.2.d.2-24　正規メッセージとの衝突メッセージ送信のシミュレーション時間と CAN メッセージ総数（正規メッセージとの同期送信）

No.	項目	通信速度	シミュレーション結果
1	シミュレーション時間	500kbps	11 秒
		125kbps	8 秒
2	CAN メッセージ総数	500kbps	3679 フレーム
		125kbps	2819 フレーム

表 3.2.d.2-25　正規メッセージとの衝突メッセージ送信のシミュレーション時間と CAN メッセージ総数（正規メッセージとの非同期送信）

No.	項目	通信速度	シミュレーション結果
1	シミュレーション時間	500kbps	22 秒
		125kbps	29 秒
2	CAN メッセージ総数	500kbps	30174 フレーム
		125kbps	12384 フレーム

攻撃 ECU が正規メッセージとの衝突メッセージ送信を行った結果､以下の現象が確認できた。

1) 被攻撃 ECU と攻撃 ECU は､送信エラーカウンタが 128 になるまでは短時間で送信エラーカウンタが増加した。これは、被攻撃メッセージと攻撃メッセージの衝突がひとたび発生すると、衝突後のエラーフレーム送信直後に被攻撃メッセージと攻撃メッセージが再送されて再度衝突が発生する、という事象が繰り返されるためである。
2) 被攻撃 ECU がエラーパッシブとなった後、被攻撃 ECU の送信カウンタは 8 増加→1 減少を繰り返した。これは、エラーパッシブ後も衝突が発生するため送信エラーカウンタが 8 増加するが、直後の再送処理により被攻撃メッセージの送信が成功するため送信エラーカウンタが 1 減少する。
3) 攻撃 ECU の送信エラーカウンタは 128 まで上がった後、1 ずつ減少する。これは、被攻撃 ECU がエラーパッシブとなることで衝突が発生してもエラーフレームが発生せず、攻撃 ECU のメッセージは送信成功するためである。
4) 被攻撃 ECU は、送信エラーカウンタが 255 に達しバスオフ状態となった。しかし、ログ上は約 2.7msec で送信エラーカウンタは 0 に戻り､バスオフ状態からエラーアクティブに復帰している。これは、11bit のリセッシブを 128 回受信（500kbps の場合、理論値で 11bit×128 回/500kbps=約 2.8msec）するとバスオフ状態から復帰するという CAN コントローラの仕様どおりの動作である。
5) 攻撃メッセージを正規メッセージに同期して送信した場合､（ア）被攻撃 ECU の送信エラーカウンタが 128 に到達するまでと、（イ）128 に到達した後から 255 に到達するまでとでは、単位時間当たりのエラーカウンタの増加量は（イ）の方が明らかに緩やかだった。（イ）では被攻撃 ECU がエラーパッシブ状態に移行しているため、衝

突後の再送までに待ち時間が追加され、再送時の攻撃メッセージとの衝突が回避されることに由来する。

5) 他方、攻撃メッセージを正規メッセージに同期せず高頻度に送信した場合、上記の（ア）と（イ）とで単位時間当たりのエラーカウンタの増加量はほぼ変化しなかった。これは攻撃 ECU が高頻度に送信することで、被攻撃 ECU の再送に待ち時間が追加されたとしても衝突が回避されなかったことに由来する。

以上のとおり、被攻撃 ECU が送信するメッセージに攻撃メッセージを連続して衝突させることで、被攻撃 ECU の送信エラーカウンタを上昇させ、バスオフ状態に至らせることができた。よって、正規メッセージ送信との衝突メッセージ送信（バスオフ誘発）による攻撃方法は再現できた。

⑤ DoS（診断用メッセージフレーム送信）

診断用メッセージフレーム送信を再現した。本シミュレーションに要した時間と CAN メッセージ総数を表 3.2.d.2-26 に示す。CAN メッセージ総数のうち、正規の送信メッセージは約 97%、攻撃メッセージは約 3%であった。

表 3.2.d.2-26　診断用メッセージフレーム送信のシミュレーション時間と CAN メッセージ総数

No.	項目	通信速度	シミュレーション結果
1	シミュレーション時間	500kbps	105 秒
		125kbps	95 秒
2	CAN メッセージ数（総数）	500kbps	1166 フレーム
		125kbps	1065 フレーム

攻撃 ECU がエンジン制御 ECU に対して診断用メッセージフレームを送信した結果、以下の現象が確認できた。

1) エンジン制御 ECU が通常モードのとき、攻撃 ECU から CAN-ID 0x120 のメッセージを送信した。このとき、エンジン状態表示パネルの r.p.m.値は通知した r.p.m.値を一瞬示すが、すぐに通常通りの（エンジン制御 ECU が通知する）r.p.m.値の表示に戻った。
2) 攻撃者が操作パネルから CAN-ID 0x701 のメッセージ（DF の変数部は xx=01 と指定）をイベント送信すると、エンジン制御 ECU は診断モードになりメッセージ送信を停止した。そして、エンジン状態表示パネルの r.p.m.値の更新（255 までのカウントアップ操作）も停止した。その状態で、攻撃 ECU から CAN-ID 0x120 のメッセージの DF を任意に設定した不正メッセージを送信することで、エンジン状態表示パネルの r.p.m.値を任意の値に変更できた。

以上のとおり、故障診断用メッセージにより攻撃対象の ECU を診断モードに移行させ、

正規メッセージの発信を抑制する攻撃方法を再現できた。

⑥ DoS（不正メッセージ送信）

不正 DLC 送信および不正 DL 送信による DoS は、④正規メッセージとの衝突メッセージ送信（バスオフ誘発）のシミュレーション結果を参照されたい。

(3) 考察

シミュレータ上での攻撃の再現、および、その結果を基に、車内通信プロトコルのセキュリティ対策に対してシミュレーションを用いた安全性評価方法の有効性について評価する。

(ｉ) なりすまし

リプレイ送信および不正メッセージ送信について 3.2d.2.②の観点により有効性評価した結果を表 3.2.d.2-27 に示す。

表 3.2.d.2-27　シミュレータによる評価方法の有効性評価（なりすまし）

有効性の評価観点	リプレイ送信	不正 DF 送信	不正 DLC 送信
1．仕様通りの挙動の再現性	○	○	△*
2．攻撃の再現	○	○	△*
3．攻撃の影響の確認性	○	○	×*
4．実施の効率性	○	○	×*

○：評価観点を満たす、△：評価観点を一部満たす、×：評価観点を満たさない
*：マイコン（Arduino UNO 等擬似 ECU 端末）を併用すれば「○」だと推測される

1) リプレイ送信の攻撃方法をシミュレータで再現できることを確認した。また、攻撃によって被攻撃 ECU の挙動が異常であることをシミュレータのログを用いて外部から観測し、攻撃による影響を詳細に確認できた。リプレイ送信は送信タイミングやメッセージの傍受の仕方によって各 ECU に与える影響も異なってくるため複数の攻撃パターンを作成する必要があり、評価する際に準備しなければならないシナリオ数が増加する。シミュレーションを用いた評価では、傍受によって得られたメッセージと送信タイミングの組み合わせによって考えられる複数の攻撃パターンを、簡単な設定変更等により短時間で効率的に実施することができる。
2) 不正 DF 送信攻撃の攻撃方法をシミュレータで再現できることを確認した。また、攻撃によって被攻撃 ECU の挙動が異常であることをシミュレータのログを用いて外部から観測し、攻撃による影響を詳細に確認できた。不正 DF 送信攻撃はメッセージの傍受の仕方や送信する不正データの内容によって各 ECU に与える影響も異なるため複数の攻撃パターンを作成する必要があり、評価する際に準備しなければならないシナリオ数が増加する。シミュレーションを用いた評価では、上記 1)と同様の方法

で効率的な評価ができる。

3) 不正 DLC 送信攻撃の攻撃方法をシミュレータで一部は再現できたが、完全には再現することは出来なかった。シミュレーションでは、攻撃 ECU で DLC を 0 に変更したメッセージを送信したにもかかわらず、受信側 ECU で値を含んだ DF（具体的には 8 バイト連続した 0x00）を取得できるなど、CANoe は DF の長さと DLC の値が一致しない場合でもメッセージの送受信が完了するという結果になった。

CANoe では CAN の DLC の挙動に関する仕様が一部再現されていなかったことにより、設定した DLC の値によって不正 DLC メッセージの送信は完全には再現できなかった。しかし、実バスを組み合わせることで DLC の挙動に関する仕様は再現され、攻撃の再現および攻撃の影響を確認できると推測する。なりすまし攻撃の主要な攻撃であるリプレイ送信や不正 DF 送信攻撃については攻撃を再現し短時間でシナリオ作成やシナリオ検証を実施できた。以上より、なりすまし攻撃においてシミュレーションを用いた評価方法は有効と考える。

(ⅱ) DoS

まず、高頻度送信ならびに診断用メッセージ送信について 3.2d.2.②の観点により有効性評価した結果を表 3.2.d.2-28 に示す。

表 3.2.d.2-28　シミュレータによる評価方法の有効性評価
（DoS（高頻度送信、診断用メッセージ送信））

有効性の評価観点	高頻度送信（通信路全体）	高頻度送信（特定 ECU）	診断用メッセージ送信
1. 仕様通りの挙動の再現性	○	○	○
2. 攻撃の再現性	○	△*	○
3. 攻撃の影響の確認性	○	△*	○
4. 実施の効率性	○	△*	○

○：評価観点を満たす、△：評価観点を一部満たす、×：評価観点を満たさない
*：マイコンを併用すれば「○」だと推測される

1) 高頻度送信の攻撃方法をシミュレータで再現できることを確認した。また、攻撃によって CAN バスに対して頻繁にメッセージを送信した際に、シミュレータのログを用いて外部から CAN バスの占有率を観測し、攻撃による影響を詳細に確認できた。正規のメッセージと優先度の高い CAN-ID の攻撃メッセージとを合わせた CAN バスの占有率が 80～90%程度では、正規の CAN メッセージの送信はほとんど遅延しなかった。通信ログを確認すれば微細な遅延を確認できたが、仮想 ECU の挙動の違いを目視で確認できるほどではなかった。攻撃メッセージによる CAN バスの占有率が 100%の状態、あるいは 99.**%といったほぼ 100%の状態を再現することで、正規の CAN メッセージの送信を停止、または遅延させることができた。シミュレータでは CAN バスの占有率を確認しながら攻撃メッセージの送信頻度を細かく容易に変更できる。

シミュレーションを用いた評価では、傍受によって得られたメッセージと送信タイミングの組み合わせによって考えられる複数の攻撃パターンを、簡単な設定変更等により短時間で効率的に実施することができた。

2) CANoe では実 ECU のハードウェアや OS の処理性能を各 ECU に対してパラメータとして設定できない。そのため、高頻度にメッセージを送信した結果、受信 ECU が描画する表示パネルの動作に処理の遅滞が確認できたものの、それが受信 ECU 単体の遅滞なのか、CANoe 全体の遅滞なのか特定できなかった。特定 ECU の処理性能を上回る負荷をかける高頻度送信による DoS をシミュレーションで評価するためには、実バスを組み合わせ、一部の仮想 ECU をマイコンに変更することで正確に再現できると推測する。

3) 診断用メッセージの攻撃方法をシミュレータで再現できることを確認した。また、正規の CAN メッセージと同じ CAN-ID かつ DF が不正な値の攻撃メッセージを攻撃 ECU から送信した際に、シミュレータのログを用いて受信 ECU の処理（表示処理など）の変化を確認することで、正規のメッセージ送信が抑制されている等の攻撃による影響を詳細に確認できた。

CANoe 単体では個々の ECU の性能をパラメータとして設定できないため特定 ECU に対する高頻度送信による DoS を完全に再現することはできないが、DoS 攻撃の通信路全体に対する高頻度送信攻撃や診断用メッセージ送信攻撃を再現できた。また、CANoe にマイコンを組み合わせることで個々の特定の ECU に対する高頻度送信を再現可能だと推測する。また、CAN バスや ECU への影響を詳細に確認、および送信頻度等の設定変更や攻撃再現が実環境と比較して容易であり、検証を効率的に実行できた。以上より DoS（高頻度送信、診断用メッセージ送信）においてシミュレーションを用いた評価方法は有効と考える。

次に、正規メッセージとの衝突メッセージ送信（バスオフ誘発）について 3.2d.2.②の観点により有効性評価した結果を表 3.2.d.2-29 に示す。

表 3.2.d.2-29　シミュレータによる評価方法の有効性評価（DoS（バスオフ誘発））

有効性の評価観点	正規メッセージとの同期送信によるバスオフ	正規メッセージとの非同期送信によるバスオフ
1. 仕様通りの挙動の再現性	△ （マイコン併用により○）	△ （マイコン併用により○）
2. 攻撃の再現性	○	○
3. 攻撃の影響の確認性	○	○
4. 実施の効率性	○	○

○：評価観点を満たす、△：評価観点を一部満たす、×：評価観点を満たさない

1) 上述のとおり、CAN コントローラの処理に依存する DoS 攻撃をソフトウェアシミュレータ（CANoe）単独で再現出来ないが、マイコンとシミュレータを組み合わせることで仕様通りの挙動が実施されていることを確認し、また正規メッセージとの同

期送信によるバスオフ攻撃、および正規メッセージとの非同期送信によるバスオフ攻撃の攻撃方法を再現できることを確認できた。

評価対象の影響は､マイコンのログ情報から被攻撃 ECU の送信エラーカウンタの上昇タイミングと上昇値を逐次確認できた。また、正規メッセージに衝突させる攻撃メッセージを送信した場合、シミュレータで取得した全通信ログから衝突の発生タイミングとバスオフ状態に至るまでの影響を詳細に確認できた。

実環境での実現難易度が高い攻撃をシミュレーションで評価し、攻撃成功の詳細条件を確認した上で実環境での攻撃の難易度を正しく把握できた。このことから、DoS（バスオフ誘発）においてシミュレーションを用いた評価方法は有効と考える。

最後に、不正メッセージ送信について 3.2d.2.②の観点により有効性評価した結果を表 3.2.d.2-30 に示す。

表 3.2.d.2-30　シミュレータによる評価方法の有効性評価（DoS（不正メッセージ送信））

有効性の評価観点	不正 DLC 送信	不正 DF 送信
1. 仕様通りの挙動の再現性	△ （マイコン併用により○）	△ （マイコン併用により○）
2. 攻撃の再現性	○	○
3. 攻撃の影響の確認性	○	○
4. 実施の効率性	○	○

○：評価観点を満たす、△：評価観点を一部満たす、×：評価観点を満たさない

1) 前述の正規メッセージとの衝突メッセージ送信（バスオフ誘発）が再現可能であり、同じ手順より不正 DLC メッセージ送信、不正 DF メッセージ送信による攻撃方法を再現できることを確認できた。
2) マイコンのログ情報から被攻撃 ECU の送信エラーカウンタの上昇タイミングと上昇値を逐次確認できた。また、マイコンのログと CANoe の通信ログを突合することで正規メッセージの送信遅延の発生状況を詳細に確認できた。

上述のとおり、CAN コントローラの処理に依存する DoS 攻撃をソフトウェアシミュレータ（CANoe）単独で再現出来ないが、マイコンとシミュレータを組み合わせることで仕様通りの挙動が実施されていることを確認した。また、本攻撃は攻撃成功の詳細条件を確認しながら実環境での攻撃の難易度を正しく把握できた。以上により、DoS（不正メッセージ送信）においてシミュレーションを用いた評価方法は有効と考える。

3.2d.3　まとめ

本検討では、車内ネットワークの通信プロトコルの仕様およびマイコンのアプリケーションにおける処理方法のセキュリティ対策を評価する際に必要となる攻撃方法の網羅的で体系的な整理・分類の検討を行った。また、攻撃方法の各分類に対応した具体的な攻撃シナリオとシミュレータで再現するための手順を明らかにし、それらに基づきシミュレータを用いた安全性評価方法の有効性を評価した。ここでは、それぞれについてのまとめ、および今後の展望について述べる。

① 車内通信プロトコルの仕様に基づく評価方法の検討のまとめ

CAN を対象とした攻撃について網羅的で体系的な攻撃方法の整理・分類を行うために、情報セキュリティの 3 性質に基づく大分類を行った上で、攻撃メッセージの作成・送信に関する攻撃者の調整可能箇所の観点からの細分化を行った。そうして得られた分類に対して、平成 27 年度の調査で抽出された CAN に対する遠隔からの既存の攻撃方法をすべて対応付け、同分類の十分性を確認した。

分類したそれらの攻撃方法について、攻撃事例の報告は少ないが CAN 以外の車内通信プロトコル（CAN-FD、FlexRay、LIN）への適用可能性を机上検討した。また、CAN 以外の各通信方式の仕様から考えられる新たな攻撃方法の可能性についても検討した。その結果、CAN を対象とした攻撃方法の抽出では挙げられておらず、また、既存文献等でも明確には指摘されていなかった新たな攻撃方法の可能性があることが明らかになった。それらを加えることで、攻撃方法の分類の拡張、網羅性の向上を図った。本検討で対象とした車内通信プロトコルと、攻撃方法の分類の対応について整理した表を表 3.2d.3-1、表 3.2d.3-2 に示す。

表 3.2d.3-1　攻撃方法（DoS）に対する各車内プロトコルの適用可能性

攻撃方法の分類		CAN	CAN-FD	FlexRay	LIN
(1)高頻度送信による送信抑制	①任意ノードに影響	可	可	可（新）※2	可（新）
	②特定ノードに影響	可	可	可（新）※2	可（新）※3
	③他ノードに影響	可	可	可（新）	可（新）※3
(2)正規フレームとの衝突による送信抑制（バスオフ）		可	可	難	可（新）※4
(3)正規フレーム以外のフレーム送信による送信抑制		可	可	可	可
(4)ヘッダの変更による送信抑制	ID	N/A	可	可（新）※2	可（新）※5
	ID 以外の部分	可	可	可（新）※2	可（新）※6
(5)データの変更による送信抑制		可	可	難	可
(6)CRC の変更による送信抑制		難※1	可	可（新）※2	可（新）

可：攻撃が可能、難：攻撃が難しい、(新)：既存の文献で明確には指摘されていない攻撃方法、N/A：CAN の攻撃の検討においては、ID は攻撃挿入箇所ではなく攻撃対象の ECU を指定するために用いるものと整理しているため、CAN-FD と異なる判定結果となっている。

（※1）CAN の攻撃の検討においては、標準的な CAN コントローラの実装を前提に攻撃の実現可能性を検討したため、CAN-FD と異なる判定結果となっている。

（※2）FlexRay では、受信ノード内のエラー発生後の実装方法に依存する。

（※3）正規ヘッダと不正ヘッダの衝突等を起こすことが可能な場合。

（※4）バスオフの発生は難しいが、limp home mode に移行させることは可能。

（※5）各ノードに割り当てられていない ID に変更できる場合。

（※6） Break Field の変更による攻撃方法は新規。ただし、Break-delimiter を衝突により 0 に変更できる場合。

表 3.2d.3-2　攻撃方法（なりすまし）に対する各車内通信プロトコルの適用可能性

攻撃方法の分類		CAN	CAN-FD	FlexRay	LIN
(1)リプレイ攻撃による不正挙動の誘発		可	可	可（新）	可（新）※8
(2)ヘッダの変更による不正挙動の誘発	ID	N/A	可	難	可
	ID 以外の部分	可	可	難	難
(3)データの変更による不正挙動の誘発		可	可	可（新）※7	可
(4)CRC の変更による不正挙動の誘発		難	難	難	難

可：攻撃が可能、難：攻撃が難しい、(新)：既存の文献で明確には指摘されていない攻撃方法、N/A：CAN の攻撃の検討においては、ID は攻撃挿入箇所ではなく攻撃対象攻撃対象の ECU を指定するために用いるものと整理しているため、CAN-FD と異なる判定となっている。

（※7）空スロットを利用した攻撃方法は新規

（※8）攻撃の成功可否は正規フレームの周期に依存。

② シミュレータによる評価方法の有効性検証のまとめ

①で整理・分類した遠隔から実施可能な CAN に対する攻撃方法を対象に、シミュレータで再現するための攻撃手順の詳細な検討を行った。既存文献から抽出した CAN の攻撃方法を基にして、既存文献では試されていない攻撃条件、例えば送信周期、攻撃 ECU の個数、通信帯域等のパラメータを変更し、攻撃が成功する詳細な条件を明らかにする手順を明らかにした。これらの攻撃手順について、①で整理・分類したほぼ全ての攻撃がシミュレータ上で再現することを確認した。しかし、バスオフ攻撃のような CAN コントローラの実装に依存する攻撃は、シミュレータ上の仮想 ECU だけでは攻撃の再現が出来なかった。このような攻撃を再現するためには、シミュレータと CAN コントローラが実装されたマイコンにより構成される環境を準備する必要があることが判った。

シミュレータ上で攻撃方法の再現を行った際に、攻撃対象 ECU の挙動をシミュレータのログにより観測し、その変化を検出することにより、攻撃が成功したと判断可能であることを確認した。このように、ログ等により攻撃が成功したか判断が可能なことに加えて、攻撃条件を変更した検証についてもシミュレータの設定変更等により容易に実施可能であることから、シミュレータを用いた評価方法は有効と考える。

③ 今後の展望

今後、車内通信プロトコルを実装した製品のためのセキュリティ対策の評価環境を構築し、運用していく上では、評価に用いる攻撃方法の多様性、網羅性が重要となる。机上検討による可能性の指摘にとどまっているものの、今回の検討において、幅広く洗い出した新たな攻撃は、前記の多様性や網羅性の向上に大いに活用可能と考える。また、今回検討したシミュレータ上での攻撃や被害の再現の手順や攻撃パラメータの選定も、シミュレータで新たな攻撃の再現性を検証する上で有効に活用可能と考える。

また、今回の検討ではシミュレータ上で評価対象とした仮想 ECU は必要最低限の処理機能しか保有しておらず、車内通信プロトコルの仕様や仮想 ECU 上のアプリケーションに対してセキュリティ対策が施されていなかったが、今後、そうしたセキュリティ対策の耐性を評価することも必要になる。その場合においても、今回洗い出した新たな攻撃やシミュレータで行った有効性検証の手順が活用可能と考える。

将来的には、車内通信プロトコルが実装された製品に対して十分なセキュリティ対策が施されているかを測るための評価基準が必要になると考える。この評価基準は、評価に用いられる攻撃それぞれに関する成功条件や成功時の影響等の普遍的な特徴・特性に基づく重み付け、それらに基づくスコアリングの方法、および、スコアと製品のセキュリティレベルとの対応付けから成ると考える。IT 分野では ISO15408 等において、このような評価基準が定められているが、車載システムの特性を考慮したものではない。今後、世の中で用いられている評価基準を参考にしつつ、車載システムや車内通信プロトコルへの攻撃の特性を考慮した評価基準の設計について検討が必要である。その際に、車内通信プロトコルへの攻撃の特性を考慮する上で、本検討で行った網羅的で体系的な攻撃方法の整理・分類やシミュレーションでの攻撃再現手順、シミュレーション結果が活用可能と考える。

参考文献

平成 27 年度実施した調査結果および調査対象

[I] 一般社団法人　日本自動車研究所, “平成 27 年度　戦略的イノベーション創造プログラム（自動走行システム）:V2X 等車外情報の活用にかかるセキュリティ技術の研究・開発プロジェクト”, pp.III-190-III260, 2016.

[II] B. Glas and M. Lewis, “Approaches to Economics Secure Automotive Sensor Communication in Constrained Environments”, escar Europe 2013.

[III] M. Wolf, A. Weimerskirch and C. Paar, “Security in Automotive Bus Systems”, escar Europe 2004.

[IV] R. Kurachi, Y. Matsubara, H. Takada, N. Adachi. Y. Miyashita and S. Horihata, “CaCAN - Centralized Authentication System in CAN （Controller Area Network）”, escar Europe 2014.

[V] R. Kurachi, H. Takada, T. Mizutani, H. Ueda and S. Horihata, “SecGW: Secure Gateway for in-vehicle networks”, escar Europe 2015.

[VI] Y. Ujiie, T. Kishikawa, T. Haga, H. Matsushima, T. Wakabayashi, M. Tanabe, Y. Kitamura and J. Anzai, “A Method for Disabling Malicious CAN Messages by Using a Centralized Monitoring and Interceptor ECU”, escar Europe 2015.

[VII] 松島秀樹, “車載制御システムを保護するセキュリティ技術”, escar ASIA 2015.

[VIII] 関口太樹, 向達泰希, 吉岡克成, 松本勉, “不正 CAN データ送信を抑制するホワイトリスト・ハブ”, SCIS 2014.

[IX] 倉知亮、高田広章、上田浩史、堀端啓史, “車載制御ネットワークにおける送信周期監視システムの提案”, SCIS 2015.

[X] 氏家良浩、岸川剛、芳賀智之、松島秀樹、田邉正人、北村嘉彦、安齋潤, “車載ネットワークにおける CAN フィルタの提案”, SCIS 2015.

[XI] 氏家良浩、岸川剛、芳賀智之、松島秀樹、田邉正人、北村嘉彦、安齋潤, “車載ネットワークにおける監視・検証モード切換えの提案”, SCIS 2015.

[XII] 氏家良浩、岸川剛、芳賀智之、松島秀樹、田邉正人、北村嘉彦、安齋潤, “車載ネットワークを保護するセキュリティ EC の提案：HW/SW 協調による更新可能な CAN の保護方法とその評価”, SCIS 2015.

[XIII] 松本勉、中山淑文、向達泰希、土屋遊、吉岡克成, “CAN における再同期を利用した電気的改ざん”, SCIS 2015.

[XIV] 松本勉、中山淑文、向達泰希、土屋遊、吉岡克成, “車載 ECU に対する CAN 経由のファジング方法”, SCIS 2015.

[XV] 矢嶋純、武仲正彦、長谷部高行, “攻撃メッセージの無効化機能を備えたホワイトリスト CAN ハブ”, SCIS 2015.

[XVI] 久保田貴也、中野将志、倉知亮、本田晋也、汐崎充、藤野毅, “車載 CAN 通信暗号化デモシステムの構築とサイドチャネル攻撃評価”, SCIS 2015.

[XVII] 森田信義、伯田恵輔、大和田徹, “車載ネットワーク向けメッセージ認証方式の提案”, SCIS 2015.

[XVIII] 氏家良浩、岸川剛、芳賀智之、松島秀樹、田邉正人、北村嘉彦、安齋潤, “車載ネットワークを保護するセキュリティ ECU の提案: 導入インパクトを抑えた CAN 保護方法のコンセプトとその評価”, SCIS 2015.

[XIX] 松本勉、向達泰希、土屋遊、中山淑文、吉岡克成, “電気的データ改ざんに対する CAN のインテグリティ強化策”, CSS 2014.

[XX] 倉知亮、高田広章、上田浩史、堀端啓史, “CAN におけるエラーフレーム監視機構の提案”, CSS 2015.

[XXI] 小林優希、中山淑文、松本勉, “CAN における不正送信阻止が可能となる条件”, CSS 2015.

[XXII] C. Valasek, C. Miller, “Remote Exploitation of an Unaltered Passenger Vehicle”, Black hat USA 2015.

[XXIII] K. T. Cho and K. G. Shin, “Error Handling of In-vehicle Networks Makes Them Vulnerable”, ACM CCS 2016.

3.2d.1.(1) 攻撃方法に関する前提

[1] Microchip Technology Inc., “SPI インタフェーススタンドアロン CAN コントローラ MCP2515,” 2005.

[2] デンソーエレクトロニクス研究会, “図解カーエレクトロニクス 下 要素技術増補版”, 日経 BP 社, pp.189-190, 2014.

3.2d.1.(2) 攻撃方法の分類

[1] Organisation for Economic Co-operation and Development, “OECD Guidelines for the Security of Information Systems and Networks: Towards a Culture of Security,” 2002.

[2] Microchip Technology Inc., “SPI インタフェーススタンドアロン CAN コントローラ MCP2515,” 2005.

3.2d.1.(4) 攻撃方法の他通信プロトコルへの適用可能性検討

[1] LIN Specification Package Revision 2.2A, LIN Consortium December 31, 2010.

[2] M. Wolf, A. Weimerskirch and C. Paar, Security in Automotive Bus Systems, escar 2004

[3] K. Kitamura, D. Fujimoto and T. Matsumoto, LIN Security: Making-ECU-Sleep Attack and Its Countermeasure, SCIS 2017 2E3-1, 8 pages （in Japanese）

[4] J. Takahashi, Y. Aragane, T. Miyazawa, H. Fuji, H. Yamashita, K. Hayakawa, S. Ukai and H. Hayakawa, Automotive Attacks and Countermeasures on LIN-Bus, Journal of Information Processing Vol.25 pp.220-228 （Feb. 2017）.

[5] FlexRay Communications System Protocol Specification Version 3.0.1, FlexRay™

[6] 佐藤道夫, 車載ネットワーク・システム 徹底解説, CQ 出版社, 2005.

[7] K.-T. Cho and K. G. Shin, Error Handling of In-Vehicle Networks Makes Them Vulnerable, CCS 2016, ACM pp.1044-1055.

[8] D. K. Nilsson, U. E. Larson, F. Picasso, and E. Jonsson, A First Simulation of Attacks in the Automotive Network Communications Protocol FlexRay, CISIS 2008, ACM 53, pp.84-91 2009.

[9] CAN with Flexible Data-Rate Specification Version 1.0 （released April 17th, 2012）

[10] ISO 11898-1:2015（en） Road vehicles — Controller area network （CAN） — Part 1: Data link layer and physical signalling

3.2d.2 シミュレータによる評価方法の有効性検証

[1] Vector, CANoe, https://vector.com/vi_canoe_en.html

付録Ａ　用語集

- BRS：Bit Rate Switch
- CAN：Controller Area Network
- CAN-FD：Controller Area Network with Flexible Data rate
- CSMA/CA：Carrier Sense Multiple Access with Collision Avoidance
- CRC：Cyclic Redundancy Check
- DoS：Denial of Service
- DF：Data Field
- DLC：Data Length Code
- ECU：Electronic Controll Unit
- ESI：Error State Indicator
- FDF：FD Format Indicator
- FTDMA：Flexible Time Division Multiple Access
- IDE：Identifier Extension
- LIN：Local Interconnect Network
- MOST：Media Oriented Systems Transport
- OBD-II：On-Board Diagnostics-II
- PID：Protected Identifier
- RRS：Reserved Bits
- res：Reserved Bits
- SENT：Single Edge Nibble Transmission
- TDMA：Time Division Multiple Access

3.2e　実機を用いた評価の実施

自動車を構成する実機に対して、サイバー攻撃を行うための技術を開発する。攻撃対象として現在、標準的なコンポーネント(ECU)、複数のコンポーネントから構成されるシステム、システムの組み合わせにより構成される車両本体、および車両本体とそれを取り巻くモビリティ社会を想定している。これらはインタフェースを含めた機能および計算機資源が異なるため、画一的な攻撃手法を確立することはできない。

本テーマでは一般的な組み込み機器に対する攻撃などを参照し、コンポーネントからシステムまでと範囲を広げながら､様々な攻撃を試みていくこととしており､平成 28 年度は、コンポーネントレベルおよびシステムレベルへの攻撃を試みた。その成果は評価技術開発へ展開し、評価基準作成および評価環境作成へ活用することを想定している。

3.2e.1　コンポーネントに対する攻撃側のプロファイル調査

攻撃側のプロファイル調査は、主に本テーマ作業従事者の属性をリストアップすることにより行う。リストアップ項目例を列挙する。

・計算機科学教育を受けた年数や学位、およびその時期と年齢
・IT 産業において作業に従事した年数
・ソフトウェア開発経験がある場合、1 日にコーディングできるライン数
・3.2e.2 で述べた攻撃調査に要した時間
・3.2e.3 で述べる攻撃実施に要する時間

平成 28 年度に実施した攻撃作業における被験者のプロファイルを以下に記載する｡また、攻撃の具体的な実施は、スキル・経験の異なる 4 チーム（チーム A、チーム B、チーム M、チーム S）を編成して、チームごとに独立に実施するという方法をとった。

チーム A：被験者 A、被験者 B：2 名ともに 20 代、学部卒相当のコンピュータサイエンスに関するリテラシを持つ。
チーム B：情報工学を専攻する学部 4 年生 4 名
チーム M：情報工学を専攻する大学院博士課程前期（修士課程）1 年生 4 名
チーム S：情報工学を専攻する大学院博士課程前期（修士課程）2 年生 3 名と博士課程後期 3 年生 1 名と博士研究員（ポスドク）1 名の計 5 名

上記の攻撃者には、事前にある程度の情報を与えた。与えた情報は以下のシナリオの手順により知見を得たという前提を実現するに十分な量である。

・攻撃者は自身が保有する自動車を対象として、ディーラで行われるリプログラミングをモニタし、正規ツールによるシーケンスを得た。
・UDS（Unified Diagnostic Services）と呼ばれる国際規格 ISO14229 にて定義されるセキュリティアクセス（ツールとコンポーネント間の認証など）の外部仕様は、インターネットなどへの流出情報を活用した。

3.2e.2　コンポーネントの仕様と想定される攻撃の調査

テーマ②b において開発した標準コンポーネントは、リプログラミングとデバッグの機能、その機能を制御するマイコンに搭載されたセキュリティ IP、マイコン上のソフトウェアとして AUTOSAR にて検討している BSW のセキュリティモジュール、さらに、自身以外のコンポーネントと連携するなどを目的とした車載 LAN のインタフェースをそれぞれ搭載している。まず、それら搭載機能の仕様や特徴から、定性的にどのような攻撃が想定可能かを把握する。

実際の現場においてリプログラミングおよびデバッグは、ハードウエアとしての標準コンポーネントが具備する JTAG などのデバッグ用インタフェース、および、車載 LAN を利用して実施される。デバッグ用インタフェースは開発時のみならず、製品として出荷した後に問題が発覚した際に、その解析目的にそのまま残されるケースがある。車載 LAN は車両として組み立てた後に、主にリプログラミングを目的に利用されるケースがある。車両のライフサイクルにおいて、双方のインタフェースは有効であり続けるため、攻撃のインタフェースとして検討することは現実的である。

JTAG については以降で述べる標準コンポーネントとして利用する評価ボード､および、評価ボードと接続するデバッガの機能をそのまま利用し､車載 LAN については ISO14229、および、ISO15765 にて定義されたプロトコルを利用する。攻撃の調査ではこれらプロトコルの事前調査を含め、攻撃の具体的な方法を検討した。これまでに組込みシステムに対して、以下の様な攻撃事例が報告されている。

- JTAG からメモリの内容を直接読み書きできるデバイスであり、セキュリティ対策が施されていない、あるいは迂回できることがあるため、ファームウェアの改ざんが可能となるといった攻撃
- JTAG を経由してメモリ上のデータを読み出すことで、暗号ハードウェアの秘密鍵を抽出し、その鍵を利用して行う攻撃

これらの攻撃は攻撃対象車両に対する物理的接触を伴う直接侵入型の攻撃であり、攻撃者にとってコストがかかる一方、多数のハードウェアチップで同じ鍵が用いられているような場合には、そのコストに見合うだけの効果がある可能性があり、評価の必要がある。

事前調査における､リプログラミング時の認証は ISO14229 を利用しており､また Secure Hardware Extension(以下、SHE）を利用しているという情報から、リプログラミングの認証の手順を以下の様に推定した（図 3.2e.2-1)。

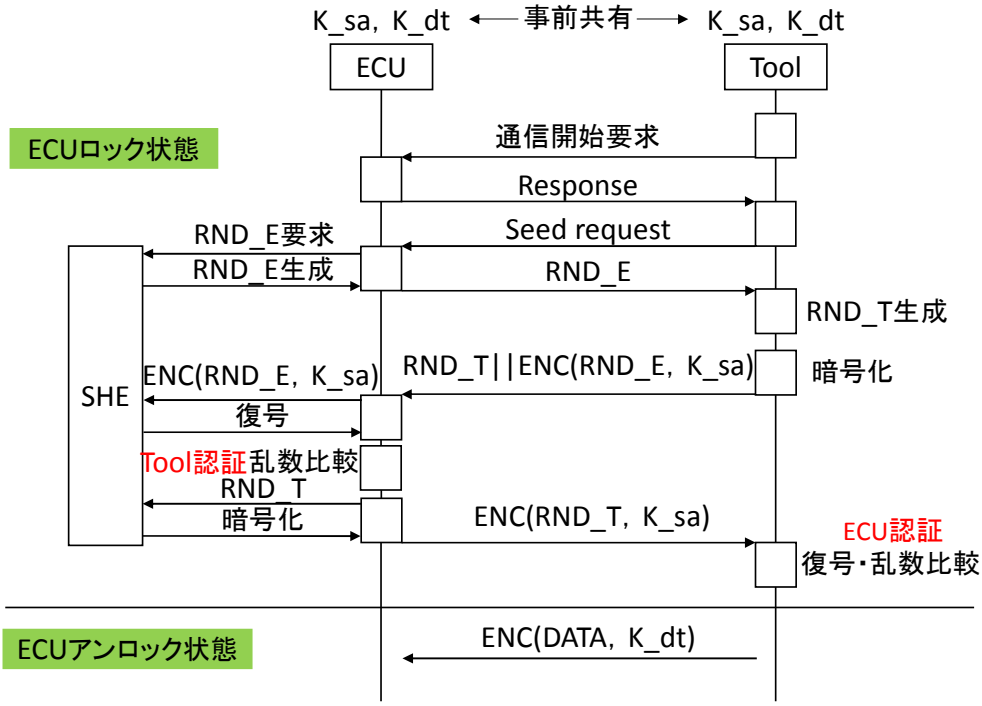


図 3.2e.2-1　リプログラミング時の認証手順（推定）

まず、2 種類の秘密鍵 K_sa、K_dt を ECU と Tool 間で事前共有しておく。Tool から通信開始要求のメッセージが送信され、ECU からそれに対するレスポンスが送信されることで認証が開始される。Tool から Seed 要求が送信されると、ECU は SHE に実装されているハードウェア乱数生成器に RND_E を要求する。ハードウェア乱数生成器で生成された RND_E を ECU が受け取り、それを Tool に送信する。次に Tool でも乱数 RND_T を生成する。また、ECU から受け取った RND_E に対して鍵 K_sa を用いて AES で暗号化し ENC(RND_E, K_sa)とする。RND_T と ENC(RND_E, K_sa)を ECU に送信する。次に ECU では、ENC(RND_E, K_sa)の復号を行う。暗号の処理は SHE 側で行うため、ENC(RND_E, K_sa)を SHE に渡し、復号した結果を得るものと考えられる。復号して得た値が RND_E と一致するかを比較し Tool 認証を行う。

乱数比較が一致しなかった場合にはネガティブメッセージが送信されそこで通信が終わる。乱数が一致すると、Tool から受け取った RND_T に対して鍵 K_sa を用いて AES で暗号化し ENC(RND_T, K_sa)とし、それを Tool に送信する。Tool で ENC(RND_T, K_sa)を復号して得られた値と RND_T を比較し ECU の認証を行う。乱数比較が一致しなかった場合には、ネガティブメッセージが送信されそこで通信が終わる。乱数が一致すると ECU アンロック状態となる。

また、図 3.2e.2-2 に UDS で定義されているリプログラミングに関する CAN パケットの仕様を示す。図 3.2e.2-2 中の「xx」は認証に関わるデータを表す。

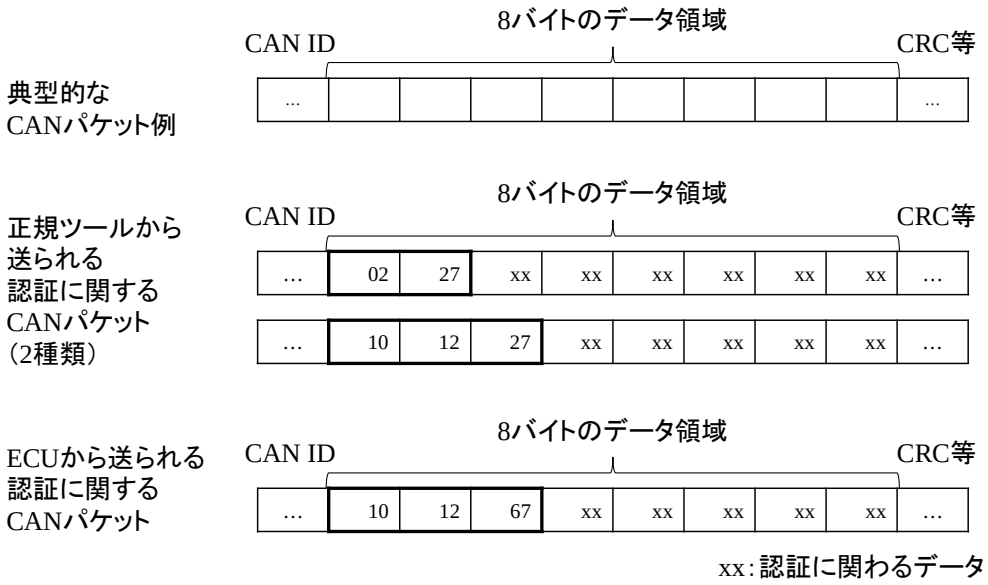


図 3.2e.2-2　UDS で定義されているリプログラミングに関する仕様

また、今回評価対象とした標準コンポーネントでは、SHE に準拠した実装が行われている。SHE の構成と、内部に格納されるデータを図 3.2e.2-3 に示す。SHE では真性乱数生成器（TRNG：True Randam Number Generator）と擬似乱数生成器（PRNG：PseudoRandom Number Generator）が実装されている。PRNG として暗号化にも使用する AES-128bit を CTR モードで使用する。PRNG に関わる値として、Non-volatile Memory の PRNG_SEED、RAM の PRNG_KEY と PRNG_STATE が存在する。PRNG_SEED は PRNG の開始値であり、AES で乱数を発生する際の初期値にあたる。PRNG_KEY は SECRET_KEY から算出された値であり、乱数発生時の鍵として使用されると考えられる。また、PRNG_STATE は PRNG の状態を保持することから、乱数発生時にフィードバックする値が格納されると考えられる。SECRET_KEY は製造時に決定された値であり、ECU の個体ごとに一意の値である。

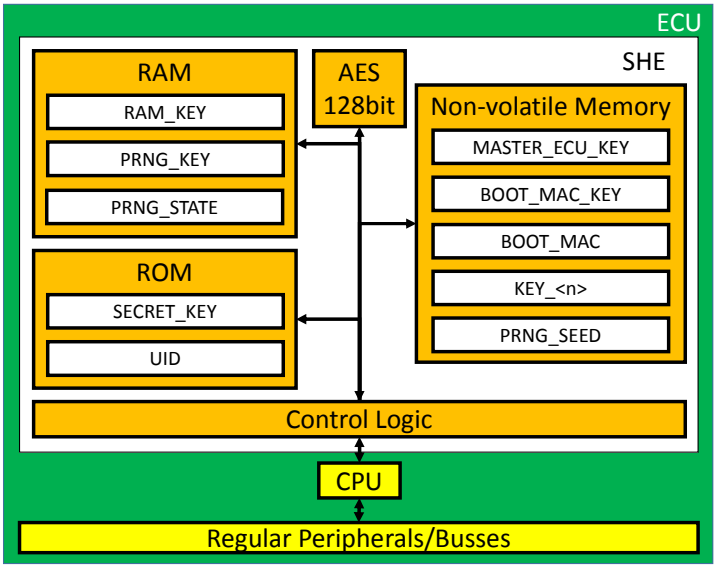


図 3.2e.2-3　SHE の構成と内部格納データ

これらの調査結果を踏まえ、以下で説明する攻撃を実施した。

3.2e.3　コンポーネントを対象とした攻撃の実施

攻撃作業は、作業コンソールとしての PC と標準コンポーネントを模擬した評価ボードを利用した UDS に対して行う。UDS を利用したリプログラミングでは、CAN を通信メディアとして利用している。今回の実験に使用した機器は以下の通りである。今回は、リプログラミング機能の導入先として SHE 準拠チップであるルネサスエレクトロニクス社製 RH850 シリーズを選定した（図 3.2e.3-1）。

- 正規ツール: コンソールとして Windows PC を利用し、CAN アダプタ「Elyzer」を具備する
- RH850 評価ボード（ECU）: UDS をサポートしたソフトウェアをインストール
- 盗聴およびリプレイ攻撃ツール: コンソールとして Windows PC を利用し、CAN をクリッピングして接続

この他、攻撃手法によっては、以下のツールも利用した。

- デバッガ「E1 エミュレータ」「CS+」「adviceLUNA II」
- CAN エミュレータ「CANoe」

これらを利用して、事前調査から得られた定性的な結果を元にした攻撃を実施した。

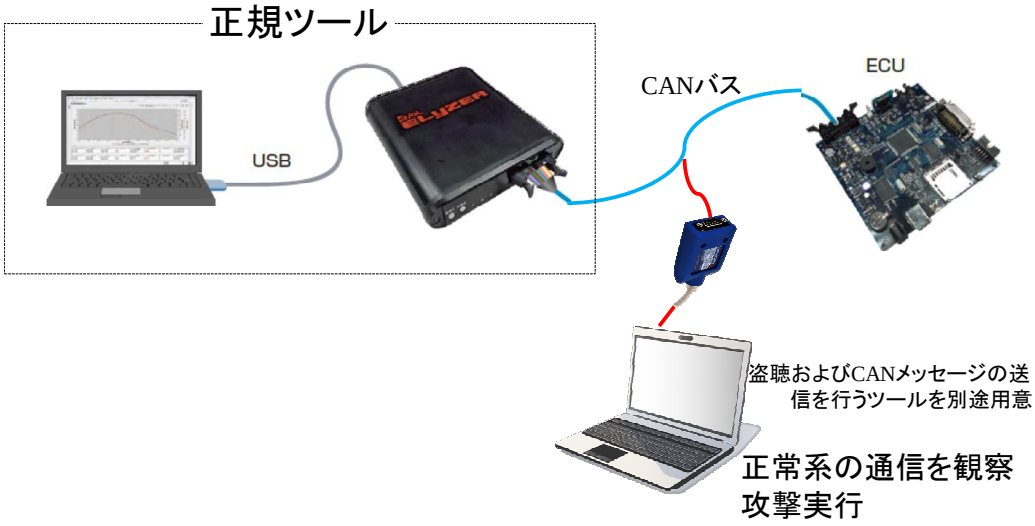


図 3.2e.3-1　攻撃環境

また、攻撃実施時において得られる成果物として、以下の項目を想定した。

- 各攻撃実施における具体的な手順
- 各攻撃実施における所要時間

(1) チーム A による攻撃評価

UDS として定義されているセキュリティ機能は、ECU・ツール間の認証フェーズと、実際にリプログラミングを行うフェーズに分割される。前者は互いに正しい機器同士であるかの確認、後者はイメージの完全性確認などを目的としている。今回は認証フェーズに関するセキュリティ機能へ以下の攻撃を行った。

・乱数生成時に重複する問題がある可能性を想定し、乱数生成と認証を 10,000 回実行し、ログを取得し解析
　－生成した乱数に重複は確認できなかった
　－生成された乱数の偏りも確認できなかった
・SHE における乱数生成を調査
　－SHE の乱数生成には PRNG を使用していることが判明していたため、シードの値を調査した結果、乱数の生成方式が判明
　－一方で TRNG の乱数生成の手法は実現方法に依存するため、公開情報ベースからの特定が困難であり、また物理的なリセットをかけることも同様に困難であることが明らかとなる
　－結果、乱数生成のリセットからのアプローチを断念

上記の手続きに費やした時間を図 3.2e.3-2 および図 3.2e.3-3 にまとめた。2 人の攻撃者が合計約 130 時間攻撃を試みた結果、リプログラミングの認証機能に対する攻撃は成功しなかった。

<table>
<tr><th>経過時間</th><th>作業者A</th><th>作業者B</th></tr>
<tr><td>START</td><td colspan="2">実験と装置の操作方法の説明</td></tr>
<tr><td></td><td colspan="2">通信ログ取得（リプロ）</td></tr>
<tr><td>1</td><td colspan="2" rowspan="2">SHEの仕様を調査</td></tr>
<tr><td>2</td></tr>
<tr><td>3</td><td rowspan="3">CAN通信用ツールのセットアップ</td><td>リプロにアプローチ開始</td></tr>
<tr><td>4</td><td>通信ログにおいて、
生成された乱数の重複などの調査</td></tr>
<tr><td>5</td><td>生成された乱数の偏り調査</td></tr>
<tr><td>6</td><td rowspan="2">引き続き、CAN通信用ツールのセットアップ
（ツール自体に問題があり失敗）</td><td>SHEの仕様調査</td></tr>
<tr><td>7</td><td>SHE準拠チップを生産している会社を調査</td></tr>
<tr><td>8</td><td rowspan="2">リプロについて、鍵や定数、パスワード等を
色々変更してレスポンスの生成を行う（失敗）</td><td>SHEにおける乱数生成の調査</td></tr>
<tr><td>9</td><td>SHEにおけるPRNGの仕様調査</td></tr>
<tr><td>10</td><td rowspan="2">リプロについて、KDF等を使用するのかどうか
等仕様を調査。結果は得られず。</td><td>SHEにおけるTRNGの仕様調査</td></tr>
<tr><td>11</td><td>他の攻撃手法の考察</td></tr>
<tr><td>12</td><td rowspan="2">SHEの仕様から改めて認証について調査</td><td>NXP社のドキュメント調査
（リプロに関する記述の有無を確認）</td></tr>
<tr><td>13</td><td>実験で使用するチップの調査</td></tr>
<tr><td>14</td><td rowspan="2">持っている鍵候補等でチャレンジに対するレス
ポンスの生成を行ったが失敗。</td><td>リプロにおけるKDFの仕様調査</td></tr>
<tr><td>15</td><td>リプロに関する論文検索</td></tr>
<tr><td>16</td><td rowspan="2"></td><td>配送鍵を用いたリプロの認証突破の試行錯誤</td></tr>
<tr><td>17</td><td>AESの条件付き全数探索（プログラム作成）</td></tr>
<tr><td>18</td><td rowspan="15">AESの条件付き全数探索（プログラム実行）</td><td rowspan="15">AESの条件付き全数探索（プログラム実行）</td></tr>
<tr><td>19</td></tr>
<tr><td>20</td></tr>
<tr><td>21</td></tr>
<tr><td>22</td></tr>
<tr><td>23</td></tr>
<tr><td>24</td></tr>
<tr><td>25</td></tr>
<tr><td>26</td></tr>
<tr><td>27</td></tr>
<tr><td>28</td></tr>
<tr><td>29</td></tr>
<tr><td>30</td></tr>
<tr><td>31</td></tr>
<tr><td>32</td></tr>
</table>

図 3.2e.3-2　リプロ攻撃の作業チャート (1)

経過時間	作業者A	作業者B
33	AESの条件付き全数探索（プログラム実行）	AESの条件付き全数探索（プログラム実行）
34		
35		
36		
37		
38		
39		
40		
41		
42		
43		
44		
45		
46		
47		
48		
49		
50		
51		KDFの定数部分を試行錯誤
52		リプロで利用する鍵の種類の特定
53		
54		リプロのアルゴリズム調査
55		CAN通信用ツールの利用を試みる
56		リプロの認証の仕様調査
57		
58		SHEにおけるマスター鍵の調査
59		SHEにおける鍵の更新について調査
60		タイムアウト時間の検証
61		リプロに関するUDSの調査
62		
63		
64		乱数の偏りの検証
65	実験終了	実験終了

図 3.2e.3-3　リプロ攻撃の作業チャート (2)

(2) チーム B による攻撃評価

3.2e.2 節で調査した情報より、ECU 上で RND_E を生成する手法は、図 3.2e.3-4 に示す手法であると考えた。PRNG_SEED を初期値として PRNG_STATE に格納し、SECRET_KEY より算出された PRNG_KEY を用いて暗号化し、暗号文を RND_E とすると考えられる。この手法では、PRNG_SEED が定数か、周期が小さい場合には、複数回初期化を行うことで同じ乱数列より RND_E が出力される場合があると考えられる。そこで、ECU で発生させる乱数に注目した攻撃を実施した。

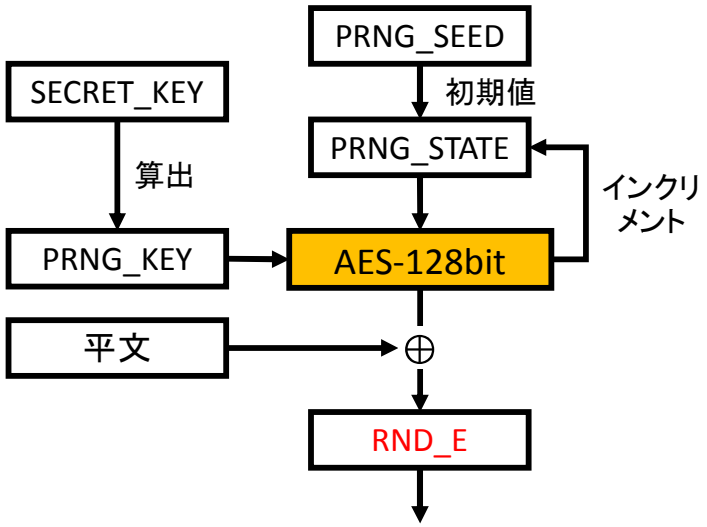


図 3.2e.3-4　ECU の RND_E 生成手法

① 攻撃 1：低電圧で評価ボードを動作させる攻撃

この攻撃では評価ボードを動作させる際に入力する電圧を評価ボードの稼動限界まで下げることで乱数の質を下げることを目的とした。まず、評価ボードに対し外部電源を接続し、稼動する下限電圧を調査した。次に、乱数の質に変化があるかを確認するため、稼動電圧を下限電圧から 0.01V ずつ上げていき、各電圧に対し乱数発生セッションを 1000 回ずつ行わせ、そのログを保存した。各ログから乱数にあたる部分に対し検定を行ったところ特に乱数の質に変化は見られなかった。この攻撃には約 2 時間を要した。

なお、この攻撃の主たる対象は評価ボードに搭載されたマイコン（RH850）であるが、評価ボード上への実装が想定される電圧レギュレータの影響によりマイコンに実際に印加された電圧は不明である。今回は、評価ボードのハード的な改造は行わない前提のため、外部からの供給電圧を制御する方法だけを試すこととした。

② 攻撃 2：セッション毎にリセットする攻撃

セッション毎に評価ボードのリセットボタンを押すことで、PRNG_STATE などが初期化されることで乱数に偏りが発生することを狙いとしてこの攻撃を行った。この攻撃の電源には安定化電源を用い攻撃 1 で求めた下限電圧で評価ボードを稼動させた。攻撃手順とし

ては乱数発生セッションを 1 回行い､評価ボードのリセットボタンを押すことを 50 回繰り返し、その際のログを保存した。このログの乱数部分に対し検定を行ったが特に偏りなどは見られなかった。この攻撃には約 1 時間 30 分を要した。

③ 攻撃 3：セッション毎に電源を落とす攻撃

攻撃 2 におけるリセットボタンを押す動作を評価ボードの電源を落とす動作に変更したのがこの攻撃である。この攻撃での電源は評価ボード付属の AC アダプタを使用した。攻撃手順としては乱数発生セッションを 1 回行い、評価ボードから AC アダプタを引き抜くことで電源を落とすことを 50 回繰り返しその際のログを保存した｡このログの乱数部分に対し検定を行ったが特に偏りなどは見られなかった｡この攻撃には約 1 時間 30 分を要した。

④ 攻撃 4：ROM からの鍵抽出攻撃

攻撃実験の目的は、ROM に格納されている SEACRET_KEY を解析によって得ることである。攻撃実験は、評価ボードを動作させたときの解析ログを確認するという方法を用いたが、得られた結果としてはログの値がすべて 0 になっており、解析することができなかった。この実験には 5 分要した。

⑤ 攻撃 5：RAM からの鍵抽出攻撃

攻撃実験の目的は、PRNG_KEY と PRNG_STATE を抽出することである。実験方法としては、評価ボードを動作させているときの解析ログをリアルタイムで観察した。実験の結果として、暗号化された平文の位置を特定することができたが、PRNG_KEY や PRNG_STATE の抽出は出来なかった。この実験に要した時間は約 4 時間であった。

⑥ 攻撃 6：Flash メモリからの鍵抽出攻撃

攻撃実験の目的は、SECRET_KEY の抽出である。SHE の規格で FLASH メモリの同じスロットに、SECRET_KEY と EMPTY ブロックと FLAGS が格納されている。評価ボードを動作させたときの解析ログにより､SECRET_KEY を見つけられるのではないかと考え実験を行った。この実験では、ボード上で動作しているときに解析ログをリアルタイムで確認しようとしたが、確認は出来なかった。この原因としては、SHE からのアクセスとの同時アクセスができないことによるものだと考えられる。そこで、評価ボードを動作させ、停止させた後に FLASH メモリを確認した。しかし、ログから SECRET_KEY を抽出することは出来なかった。この実験に要した時間は約 2 時間であった。

(3) チーム M による攻撃評価

攻撃評価の目標を図 3.2e.2-1 の手順において ECU アンロック状態にすることと設定し、乱数の推定、乱数の操作、Tool 認証時の乱数比較部の改竄、鍵の特定、の 4 つの攻撃可能性を考えた。

① 攻撃 1：相互相関関数を用いた乱数の周期性の検証（乱数の推定）

乱数生成にはハードウェアで生成しているため周期性がある可能性がある。そこで今回、相互相関関数を用いて乱数の周期性を確認した。まず、取得できた CAN 通信のログは乱数がマルチフレームで送信されているためにマルチフレーム内の乱数を抜き出して結合する必要があったため、その作業を行った。その後、相互相関関数の検証を行う際のデータ数及びビット幅は(a)128bit、入力乱数データ数 10000 件、(b)32bit、入力乱数データ数 5000 件の 2 種類とし、それぞれデータ間の相互相関関数を求めた。両ケースとも高い相関が得られることはなかった。データ処理に要した時間は、それぞれ約 5 時間であり、(b)の処理を行うのに必要なマルチフレームを結合するプログラムの作成に約 1 時間要した。さらに乱数の検証のプログラムを作成するのに約 2 時間を要した。

② 攻撃 2：CANoe とデバッガを使った攻撃

CANoe からメッセージを送信し、そのときの ECU の挙動をデバッガを用いて観察することで、乱数がメモリのどこに保持されているのかの特定と、乱数の検証部分の特定を試みた。CANoe とデバッガを使った攻撃では、生成された RND_E をデバッガにより書き換えることにより、マイコンの認証を通過することができた。攻撃対象のマイコンは認証なしでデバッガから制御可能であった。また、CANoe 端末ではトレース画面で CAN メッセージの送受信の様子が観測でき、プログラミングをすることで任意のメッセージを送信することが可能であった。

CANoe を使った攻撃の結果、ECU で生成された RND_E が格納されるメモリの場所を特定することが出来、任意の値に書き換えることが出来た。これにより再送攻撃が可能であると考え、事前に取得していたログを利用して CANoe から再送攻撃を行ったが、認証を通過することは出来なかった。これはメモリ上で RND_E を検索した際にデータが一致する箇所が他にもあったことから、両方のデータを比較し一致しないと認証が通らない仕組みになっているものと推察される。

また、CANoe からメッセージを送信し、そのメッセージを受信した時点で ECU の動作を停止させ、ブレークポイント・ステップ実行を行い、プログラムがどのような動作をしているかを確認したが、Tool 認証時の乱数比較をしている命令の箇所を特定することはできなかった。

この攻撃評価に要した時間は全体で約 13 時間であった。具体的には、メッセージを受信するまでループしていることを特定するまでに約 2 時間、メッセージの送信を行う命令の特定までに約 5 時間、乱数がメモリのどこに保持されているのかを特定するのに約 1 時間、再送攻撃に約 3 時間要した。乱数の検証を行っている部分の特定には約 2 時間要したが特定できなかった。

(4) チーム S による攻撃評価

① 攻撃 1：乱数生成器の評価

乱数生成器の評価にあたっては、Elyzer の CAN 通信ログ機能を用いて、認証時に使用された乱数ログを取得した。得られたログから乱数である 16Byte のデータを抽出し以下の動作条件下で重複がないかの確認を行った。

1) 通常動作時 10 万回
2) 低電圧動作時 1000 回
3) 物理リセット直後 1000 回（×リセット回数 5 回）

通常動作時、低電圧動作時、物理リセット直後のいずれにおいても乱数値が固定されることはなく重複する値は存在しなかった。また、物理リセット直後にリプログラミングを行った際にも重複する乱数は観測されなかった。

② 攻撃 2：サイドチャネル攻撃評価

非破壊攻撃によるサイドチャネル攻撃の検討のため、まず ECU からの電力解析攻撃について検討する。電力解析攻撃の環境を図 3.2e.3-5、図 3.2e.3-6 に示す。消費電力の取得には自作の EM プローブ（製作時間:1 時間）を用い、増幅器を介しオシロスコープに接続し電力波形を得るようにした。

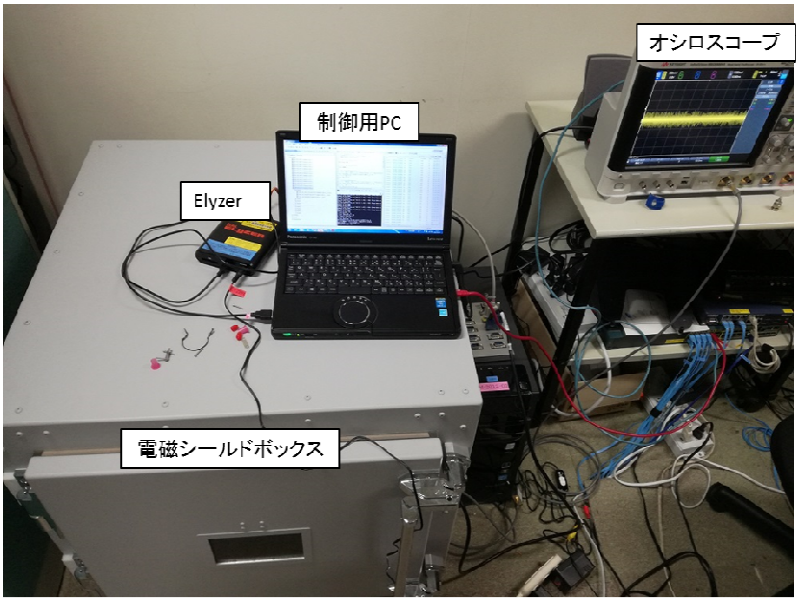


図 3.2e.3-5　磁界プローブによる測定環境外観

図 3.2e.3-6　電磁ボックス内のボードと EM プローブ

磁界プローブにより得られた波形例を図 3.2e.3-7 に示す。リプログラミングツールとオシロスコープのと間にトリガにできる信号はなかったため、電力波形自体のピークをトリガとして使用した。大きい波形ピークが何回か観測されたが、以下の図 3.2e.3-7 中に示すような処理（乱数生成・復号・暗号化）の波形と仮定して解析を進めた。乱数生成と復号処理の間にはリプログラミングを行うツールとの通信処理が入り、復号が行われる電力ピークとの間隔は不規則であった。また、得られた波形に対して位相限定相関を用いて位置あわせを行い、電力解析に用いることを考えた。

復号･暗号化の範囲は約 20ms であり､その全体を PC に転送する処理に 9sec かかり 10000 回の認証フェーズの電力波形を得るのに約 25 時間を要した｡さらに波形の位置あわせに約 6 時間を要したが、繰り返し試行することが困難であったため、電力解析攻撃の成功には至らなかった。

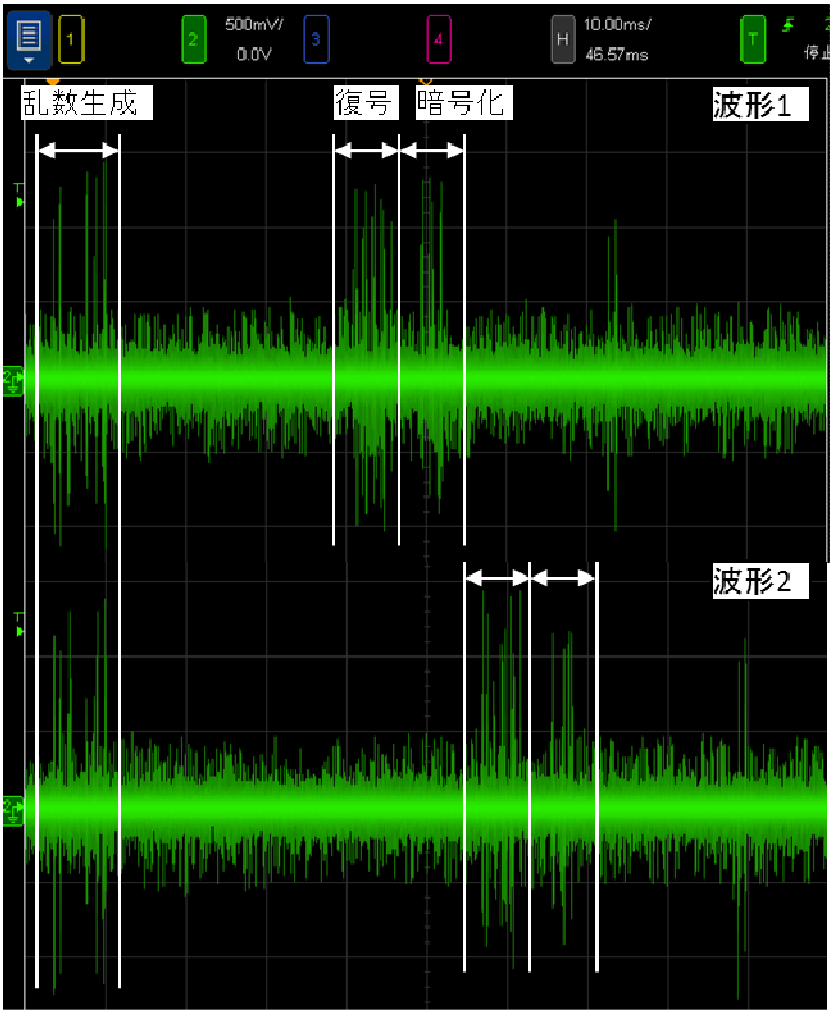


図 3.2e.3-7　磁界プローブにより取得された電力波形と認証フェーズとの対応想定

③ 攻撃 3：デバッガを使った攻撃

攻撃対象のマイコンは認証なしでデバッガから制御可能であった。デバッガを使って達成した攻撃を以下に示す。

・直接リプログラミング
・生成された RND_E の書き換え
・分岐命令スキップによる乱数検証の強制通過
・秘密鍵 K_sa の特定
・K_sa の生成元パスフレーズと鍵生成アルゴリズム SHA1 の特定

更に，デバッガを使った解析によって以下の予測を得た。

・RND_E 生成はハードウェア上で行われる。
・ECU による ENC(RND_E, K_sa)の復号はソフトウェアで行われる。

暗号機能はハードウェア上で行われると想定していたが、プログラム上に AES 復号関数と思われるものを確認した。また、RAM 上にラウンド鍵が展開されており、これが K_sa の特定につながった。K_sa はコード領域に格納してある文字列からハッシュ関数 SHA1 を

使って生成されていることがわかった。

この他、リプログラミング、生成された RND_E の書換えと再送攻撃、分岐命令スキップによる乱数検証の通過、秘密鍵 K_sa の特定、などの攻撃手法が考えられる。

3.2e.4　コンポーネントに対する攻撃結果の考察と展開

ここまでの調査結果、攻撃実施結果およびそれらに対する考察を行う。

乱数発生に着目した攻撃においては、評価ボードの外部電源による入力電圧を下げる攻撃や、乱数発生セッション毎にリセットを押す攻撃などでは乱数の質が下がることはなかった。このことからの単純な外部からの攻撃に対しては対策が施されていることが推定される。一方で、物理乱数生成器の性能に依存しているため、物理乱数生成器の仕様を超える条件下での動作を確認し、安全性の追検証が必要と考えられる。

また、今回の評価対象にはデバッガを接続することが可能であった。生成された乱数がメモリにそのまま保持されていたために、乱数がメモリのどこに保持されているかを特定できた。これを防ぐためには、メモリ上に乱数を置かないようにすることや、乱数をそのまま置くのではなく分散させておくなどが考えられる。また、デバッガを認証なしでは接続できないようにすることや、デバッガの接続を許可する場合には適切な時間でのタイムアウトを設けることも有効な対策となりえると考えられる。

ROM、RAM、FLASH メモリからの鍵抽出攻撃として行われる SECRET_KEY などの推定に関しては、デバッガを用いるなどの方法が有効であると考えられる。デバッガ等を使いこなせるなどの技能を十分に持ち合わせる攻撃者でなければ内部の情報を抜き取ることは困難ともいえるが、デバッガを使いこなすことができれば、CAN バスの通信ログと ECU のみでも攻撃可能となる可能性がある。

3.2e.5　鍵配布システムを対象とした攻撃の実施

鍵配布システムを対象とした攻撃評価は、チーム M とチーム S が実施した。

両チームには、配布用鍵 K_{AuthID} は“guanine”，“adenine”，“cytosine”，“thymine”のうちのいずれかであること、などが事前情報として与えられた。また、事前に与えられた情報及び CAN データを観測した結果から、図 3.2e.5-1 の仕様で行われていると想定した。鍵配布ではまず PC 側が、事前共有された鍵 K_AuthID を用いて配布したい鍵 KID を何らかの方式でエンコードし、ほかの何らかの情報も付け加えて鍵配布データを送信する。その後、ECU ではエンコードされたデータをデコードして KID を取り出す。これを何らかの方法で検証し、正常であれば KID を登録後、応答データを PC に送信する。

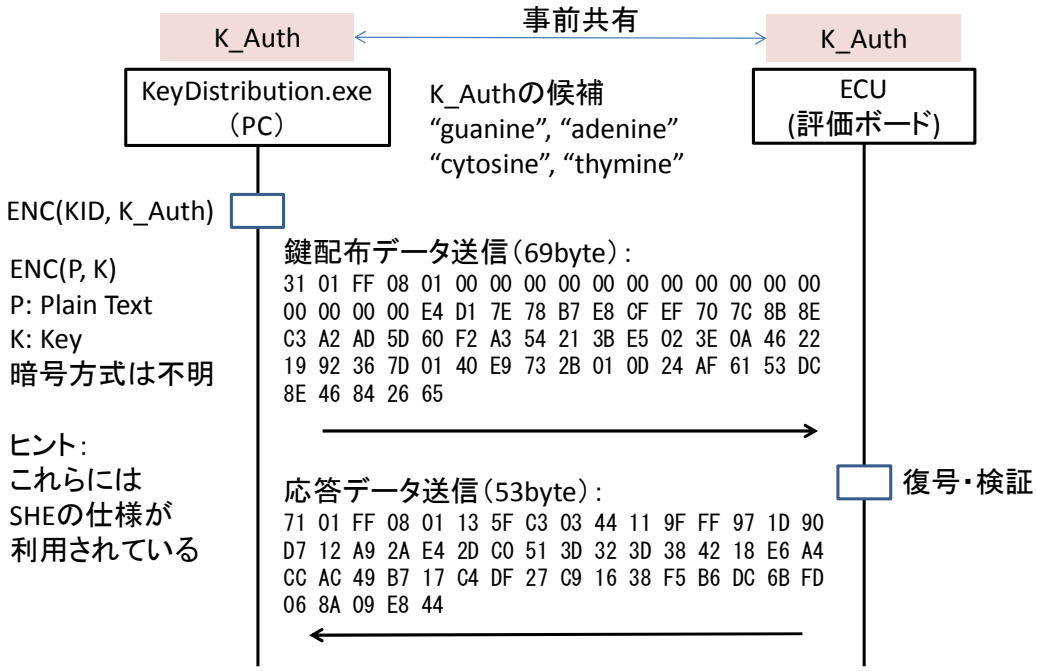


図 3.2e.5-1　鍵配布の仕様（推定）

(1) チーム M による攻撃評価

目標を K_{AuthID} の特定と登録する鍵の推定とし、通信の仕様を推定し送信データ TX を解析する手法での攻撃を試みた。以下にその手順を示す。

まず、通信の仕様の推定を行った。事前に与えられた情報から通信には SHE が用いられていることが分かっているため、SHE の仕様について約 2 時間調査を行い、以下の文献[1]、[2]を入手した.

[1] Geoff Emerson, Jurgen Frank, Stefan Luellmann, “Using the Cryptographic Service Engine (CSE) - An introduction to the CSE module,” http://cache.freescale.com/files/32bit/doc/app_note/AN4234.pdf, June 2011.

[2] 竹森 敬祐, 溝口 誠一郎, 窪田 歩, “車載暗号向け暗号鍵管理, ” SCIS2017, 2E3-3, Jan 2017.

暗号化のアルゴリズムは文献より AES-128 であることが分かった。よって暗号化に使われた鍵が分かれば復号が可能となる。その鍵は鍵導出関数（Key Derivation Function : KDF）のアルゴリズムを特定できれば求めることができる。KDF については文献[1]、[2]ともに記載されていなかったため、さらに約 2 時間調査を行い、次の文献[3]を入手した。

[3] FUJITSU SEMICONDUCTOR EUROPE GMBH, “SHE - Secure Hardware Extension,” https://www.escrypt.com/fileadmin/escrypt/pdf/WEB_Secure_Hardware_Extension_Wiewesiek.pdf, February 2012.

この文献では、KDF に Miyaguchi-Preneel を使用していたため、今回の鍵配布ツールも KDF に Miyaguchi-Preneel を使用していると仮定し、送信データ TX の復号を試みた。この

アルゴリズムを実装し、配布用鍵の 4 つの候補それぞれの場合で実行した結果、配布用鍵と登録する鍵を推定することができた。

(2) チーム S による攻撃評価

攻撃の最終目標を K_AuthID および KID の特定とすると、以下の 2 つの攻撃手法が考えられる。

① 攻撃 1：デバッガによる KeyDistribution.exe の解析

単に K_AuthID と KID を特定するだけならば、KeyDistribution.exe をデバッガで解析することで求めることが可能と考えられる。ECU のデバッガによる解析は、K_AuthID の値や保持している KID の値などが SHE によって保護されている可能性が高いため容易に導くことはできないが、KeyDistribution.exe ではソフトウェア実装されているので実行ファイル中に K_AuthID や KID が含まれているものと想定した。また、鍵配布ツールで配布鍵に関する命令を実行する際に、同じタイミングで ECU に書き込みたい鍵へのアクセスも行うと想定できるので、鍵配布ツールの解析を行うことで鍵の特定が可能だと予想した。

攻撃方法としては、配布鍵の候補の 4 種類に対しそれぞれ値を変更すると、正常時とは異なる挙動が起こることを想定しバイナリエディタで配布鍵の候補の値を変更した。結果として cytosine 以外の値を変更しても特に影響はなかったが、cytosine のみ CANUSB の OPEN に失敗した。従って、cytosine が配布鍵であると推測した。

次に、鍵配布ツールは KID を K_AuthID でエンコードするとあるので、配布鍵が格納されるメモリアドレスを参照する命令の近傍に文字列を格納するデータメモリのアドレスがあり、この文字列を変更すると鍵配布ツールが ECU 宛てに送信している値が変わることから、この文字列が登録する鍵であると特定することができた。

この解析による鍵の導出方法は、前提知識を得てから約 1 時間で成功した。

② 攻撃 2：鍵配布の仕様の特定

ここでは、チーム M と同様の攻撃手法により、同等の結果を得ることができた。この攻撃に要した時間は、文献の調査やプログラミングも含め約 5 時間であった。

(3) 鍵配布システムを対象とした攻撃結果の考察と展開

通信の仕様が分かり、K_{AuthID} の候補が 4 つに絞られていれば、送信データを盗聴するだけで容易に登録された鍵を算出することができることが確認できた。以下では、今回のような攻撃への対策について考察する。

まず、今回の攻撃は通信の仕様が分からなければ成り立たないため、KDF 等に一般的な関数をそのまま用いるのではなく、何らかのアレンジを加えるだけでもある程度攻撃は難

しくなると予想される。また、今回は K_{AuthID} の候補が 4 つに絞られていたために総当りでの攻撃が容易であったが、この候補が得られないようにする、例えば、総当りでの候補が 128bit になるようにできればこの攻撃は現実的な時間では行えないことになる。

また、復号した際に現れる平文には、その鍵が正しければ 94bit の連続した 0 が含まれる。これは仕様で定義されたものであり中盤に 0 パディングをするためである。すなわち、鍵候補があらかじめわかっていれば実際に復号し、94bit の 0 が含まれているかどうかを見ただけでどれが鍵か特定できる。これらの対策としてこの 0 パディングに乱数を使うことが考えられる。乱数を使ったとしても認証には問題がなく、復号しても 0 が連続することがないからである。

3.2e.6　システムを対象とした攻撃方法の調査

車内システムに対する調査として、攻撃調査を実施した。具体的には、無線通信を利用した車両外部からの攻撃を対象に、定性的にどのような攻撃が想定可能かを把握し、整理した。

図 3.2e.6-1 に、現在商用化されている自動車において想定できる車両外部からの攻撃一覧をまとめる。Bluetooth は対応する機器同士が共通のプロファイルを利用してサービスを実現している。また ITS ユニットは ITS Connect 推進協議会によりセキュリティを考慮した実機テストの仕様を策定している。今年度はインターネットプロトコル (IP) アドレス特定により個別車両への攻撃口となる可能性がある「WiFi」「テレマティックス」に着目し、攻撃方法の調査を行った。

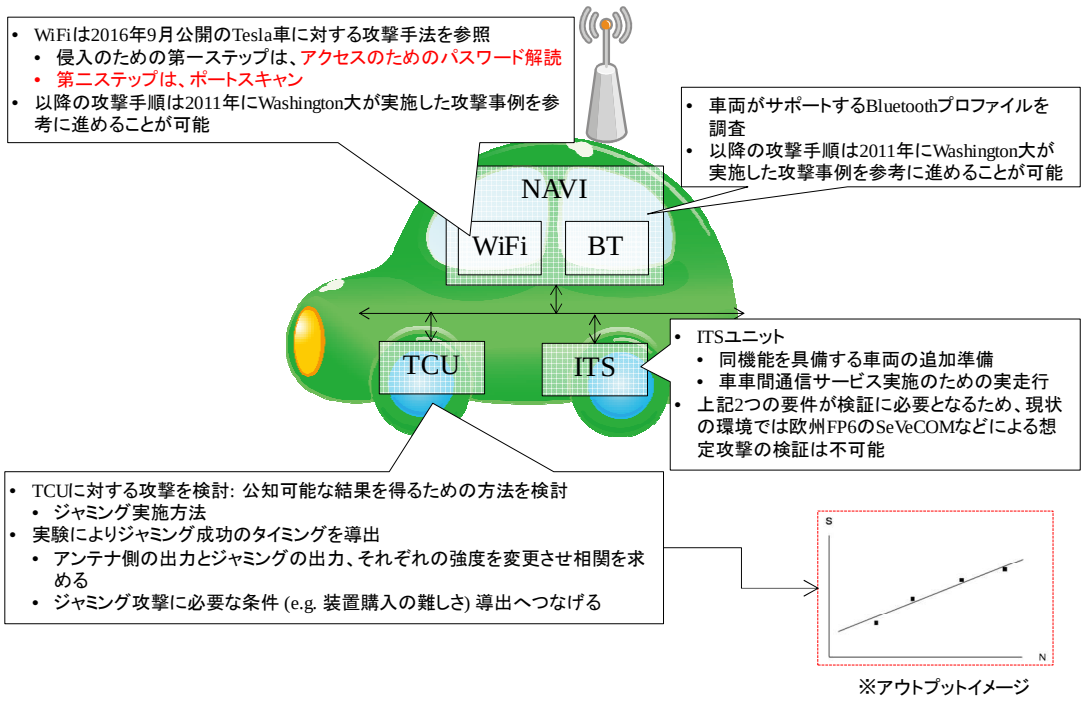


図 3.2e.6-1　車両外部からの攻撃一覧

WiFi 通信の多くは、汎用性に優れた IP をサポートしている。車両開発時において IP をサポートする WiFi 通信機能を実現した場合、開発者側が想定していない機能が有効となっている可能性が有る。一方で、攻撃者はこれらの機能をセキュリティホールとして利用するため、その耐性を評価することで製品化された車両に対して適切なセキュリティ対策が施されているかを確認する必要がある。

TCU を介した通信であるテレマティクスは、移動通信キャリアが提供する携帯電話網を利用して通信を行うケースが多いため、評価に関わるステークホルダ以外が上記携帯電話網を利用する際の影響を考慮しなければならない。具体的には評価に関わる関係者以外の一般利用者による通信の利用を妨げないことに配慮し評価を行う必要があり、例えば、電波暗室の利用などが考えられる。

図 3.2e.6-2 に WiFi への攻撃プラン例を示す。FCA および Tesla の攻撃事例は WiFi への攻撃をトリガとしている。そのためまずは、

- ✓ Tesla 同様「WiFi 子機」として動作するか確認
- ✓ FCA が具備しているテザリングルータ機能の有無確認

を実施することが考えられる。

加えて任意の WiFi ルータと接続した後に IP が利用可能となることで、以下の攻撃評価が実施可能となる。

- ✓ IP 網に接続した車外エンティティから実験車両への通信盗聴
- ✓ IP 網に接続した実験車両に対する IP ベースのポートスキャン
- ✓ IP 網に接続した実験車両に対する IP ベースのファジング
- ✓ IP 網に接続した実験車両に対する IP ベースのペネトレーション

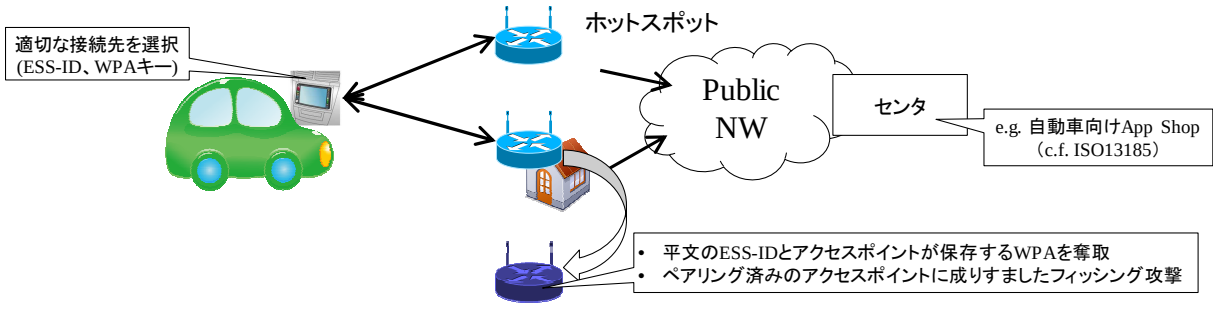


図 3.2e.6-2　WiFi への攻撃例

テレマティクス通信機能は移動通信キャリアが提供する携帯電話網を用いているため、一般利用者によるサービス享受を妨げてはならない。実験車両と携帯電話基地局との間の無線通信に対する攻撃実施に際し、実験による影響を最低限に抑えるため図 3.2e.6-3 に記載の装置を車両に対して導入する必要がある。さらにフィールドで行う場合には実験用無線免許が必要であり、実験用無線免許が準備できない場合には電波暗室での実施が必要になる。これらの準備を施した後に、実験車両と携帯電話基地局との間の無線通信に対するジャミング攻撃を行い、実験車両のテレマティクス受信機における受信電力性能を測定す

る。

測定結果を利用した受信電力性能に加えてテレマティクス受信機能が具備するサービスがジャミングによりサービス不可となったノイズ強度を測定することで、テレマティクス通信機能に対する耐性評価が可能となり、また今後の自動走行システムに求められる機能安全性能要件の検証へ寄与することが可能となる。

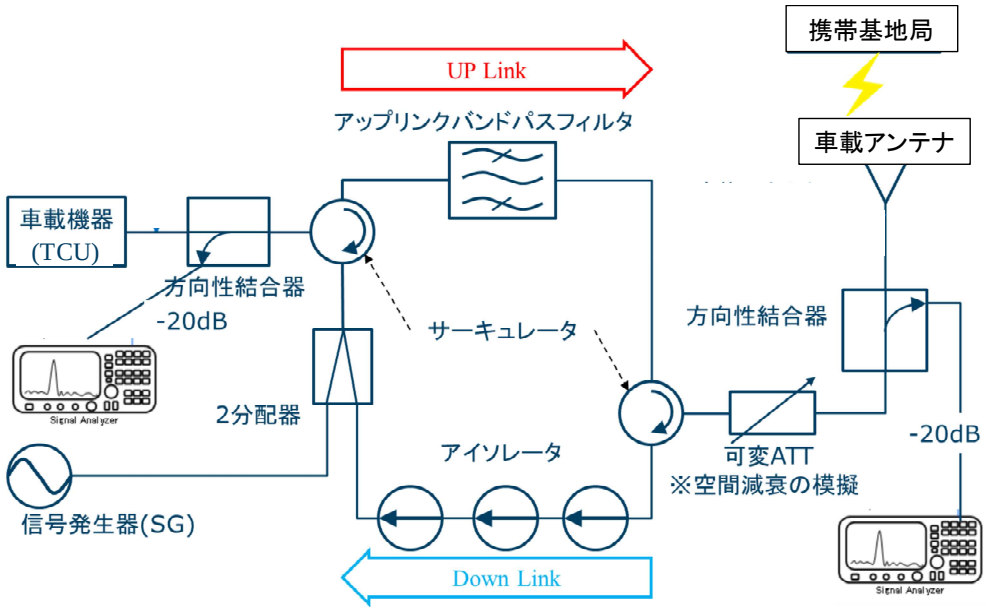


図 3.2e.6-3　TCU への攻撃時に必要な装置の実現例

3.2f　第三者認証に関する調査・検討

自動車のセキュリティに対する評価や認証については、どのように扱うかがまだ定まっておらず、国内・海外で課題となっている。そこで、評価・認証における課題を整理し、また、国内での取組にも影響を与えるであろう海外の動向について、どのように調査・検討を進めるかを議論するために、認証機関、研究機関、自動車業界等から有識者に参加いただき「認証研究会」を立ち上げた。

研究会では、まず、参加者の持つ他の業界の動きも含めた情報、特に海外の現状、及び動向に関する情報を共有した。また、海外動向調査を進めるにあたって調査すべき対象や今後の活動の方法について議論を行い、課題を明確化した。その際、それぞれが持つネットワークを互いに活用することで、海外調査活動を効率的に進められるよう、組織間の連携を図ることとした。

(1) 認証研究会における議論、及び調査

研究会は 2016 年 11 月 18 日及び 2017 年 1 月 20 日の 2 回開催した。議題は、①欧米のセキュリティ評価・認証動向、②認証の必要性と体制　③自動車セキュリティに関する人材の育成と運用体制　④ハッキングの研究と制度的支援　⑤サプライチェーンのセキュリティである。

以下、①～⑤について、認証研究会での議論に加え、主に Web 等で調査した内容を含めて整理した。

① 欧米のセキュリティ評価・認証動向

(ⅰ) UL CAP（UL、Synopsys 社、DHS）

米国 UL は、2016 年 4 月に、ネットワークに接続可能な製品/システムのセキュリティを試験・評価する技術基準として UL 2900 シリーズを発行するとともに、UL2900 への準拠を検証し認証するサイバーセキュリティ認証プログラム（Cybersecurity Assurance Program : UL CAP）を開始した。UL CAP では Synopsys 社のソフトウェア脆弱性評価ツール（Protecode、Coverity、Defensics）を評価ツールとして認定している。また、2016 年 2 月、米国ホワイトハウスの「サイバーセキュリティ国家行動計画（Cybersecurity National Action Plan : CNAP)において、米国国土安全保障省（United States Department of Homeland Security：DHS）が、UL 及び他の業界パートナーと協力してインターネット上のネットワークデバイスをテストして認証するサイバーセキュリティ保証プログラムを開発することを表明している。

UL CAP では業界別の認証規格を策定している状況であり、自動車対応は UL2900-2-4（Industry Product Testing - Automobile）であるが、現時点では発行されていない。

UL CAP はパイロット段階の制度であるが、UL 規格が米国、カナダ等で認知されていることもあり、UL、Synopsys 社、DHS の動向について、今後も注視していく必要がある。

(ⅱ) 米国 SAE

2016 年 1 月に、自動車向けセキュリティ開発プロセスのガイドライン（SAE J3061）を公開した。

米国 SAE は、J3061 の TF を組織し、ISO とも連携して “J3061-2：Testing Methods”、“J3061-3：Testing Tools”の議論を進めており、Synopsys 社の人が議長を務めている。

(ⅲ) 米国 Synopsys 社

Synopsys 社は、評価ツールの整備を進める一方で、ソフトウェアを調達する際のサプライヤに対する要求事項を記載した“Procurement Language for Supply Chain Cyber Assurance”を作成し、公開している。 この購買仕様に記載された要求事項は、IEC 62443 • ISO 27001 • NIST SP 800-53 • UL 2900 などの業界標準、ガイドラインに由来したものである。

OS からアプリケーションプログラムまでのソフトウェア全般が対象であり、

・サプライヤは、全てのシステムとデータが セキュリティプログラムを保っていることを書面で保証すること。

・評価には調達者の指定するツール、もしくは調達者の承認を受けたツールを使用すること。

などが記載されている

(ⅳ) 米国運輸省（DOT)、国家道路交通安全局（NHTSA）

2016 年 9 月、自動走行車の試験と導入を指導する国家自動走行車政策：次なる交通安全改革の促進（Federal Automated Vehicles Policy: Accelerating the Next Revolution in Roadway Safety)を発表した。法的拘束力のある規制ではなく、今後の技術進展に柔軟に対応していくことを考慮した指針となっている。

2016 年 10 月、自動車のサイバーセキュリティ・ベストプラクティス（Cybersecurity Best Practices for Modern Vehicles）を発表した。法的拘束力はなく、自動車メーカーに対して自動車サイバーセキュリティのための階層的アプローチを自発的に確立するよう求めている。

2016 年 12 月、コネクティッドカー導入に関して、一般自動車に対する V2V 通信機器搭載を義務付ける規制案を公表した。現在パブリックコメントを受付中であり、今後制定された場合は、制定の 4 年後には一般自動車の全ての新車に V2V 通信機器の搭載義務が適用されることになる。

規制への適合については、有効性を客観的に判断できるよう、同一の試験から同じ結果を得ることを条件に製造業者自身による認証を認めている。

(ⅴ) 自動車基準調和世界フォーラム（WP29）

従来の、国際的な装置単位の型式認証の相互承認制度に加え、車両全体の型式認証の相互承認制度（International Whole Vehicle Type Approval：IWVTA）が 2018 年 6 月に創設される見込みである。自動車の安全/環境基準を国際的に調和し、各国による自動車の認証を国際的に相互承認することにより、開発・認証・生産コストの低減、自動車の安全/環境性能の向上、及び我が国自動車メーカーの輸出競争力の強化等が期待される。認証の対象にセキュリティ事項が盛り込まれるかどうか注視する必要がある。

(vi) 欧州

認証機関としては TÜV が代表的な組織ではあるが、現在ヨーロッパでは自動車向けセキュリティ認証の需要が大きくないので、自動車セキュリティを扱う部門である TÜV Nord の中の TÜViT も規模は大きくないと見られる。全体的に米国が先行していると考えられるが、欧州は独自の戦略を展開することも考えられるため、今後注視していく必要がある。

② 認証の必要性と体制

自動車以外の業界でも認証の必要性が議論されているが、業界により認証の目的には差異がある。例えば情報通信、金融、鉄道、電力などの「重要インフラ」では市場は国内であり、ユーザに安心して使用してもらうための認証である。確認は自分で実施し、認証では追認するという考え方も出ている。

一方、「自動車」では国内市場に加え海外市場へも展開していることから、海外の顧客との信頼確保、及び海外で求められる仕様に速やかに対応するためには、海外の動向を的確に捉えて国際的に協調していく必要がある。

海外の状況を概観すると、現在は米国の動きが活発である。特に UL と Synopsys 社を中心として、SAE 及び米国政府をまきこんだ活動には注視する必要がある。国際会議等では、評価は必要だが必ずしも外部の評価機関に頼る必要はないといった論調とみているが、海外の情報が少ないため継続して評価・認証の動向について情報収集が重要である。自動車セキュリティの標準となる規格に準じた開発、評価、運用、及びエビデンスが必要になることが想定されるが、第三者認証が必要となるかどうかは現時点では見通せない。

認証の意義として、まず、インシデントが起きたときの説明責任が明確になることが挙げられる。認証の目的は統一された規格を用いて定まった手続きを実施することによる品質の安定確保であり、認証を取得していることで、世の中で認知された基準を満たしていることを公に証明することができる。また、自動車につながる種々の機器について、認証を実施することにより、セキュリティが考慮されていない機器を排除するなど、実質的な効果が期待できる。

一方、長期間使用する自動車では、ECU に搭載されたソフトウェアを認証した後も、自動車の全使用期間を通してセキュリティを確保しなければならず、ECU のソフトウエアを無線通信で更新（Over The Air ：OTA）する等の仕組みを導入することが必要と考えられる。その場合、更新するソフトウェアに対する認証の方法なども考慮する必要がある。

③ 自動車セキュリティに関する人材の育成と運用体制

自動運転では、自動車が通信ネットワークにつながり、自車の各種センサー情報の他、他車や周辺環境等の外部情報を利用して制御を行う。従って、通信ネットワークを含めたセキュリティが重要になり、企画・開発から製造・運用・廃棄に至る自動車のライフサイクル全体も見据えたセキュリティ対応が求められる。一方で自動車業界のセキュリティ対応に従事する人材は十分とは言えず、自動車業界としてセキュリティ対応の人材育成が急務である。

近年の自動車ではソフトウェアによる電子制御の増加が顕著であり、多くのセンサを搭載したラグジュアリ自動車では、電子制御ユニット（Electronic Control Unit：ECU）のマイクロプロセッサは 200 個以上搭載され、車載 ECU のソフトウェアの総コード数は 1 億行に達すると言われている。自動車に対する脅威はさまざま存在するが、まずは制御の核となる車載ソフトウェアが堅牢でなければならない。開発計画段階から車両安全設計同様にセキュリティを考慮することが重要であり、Security by Design を実行できる人材が重要になる。

検証段階では評価ツールの利用も考えられ、これらツールを適切に利用することが求められるが、現在は十分に活用されていないと考えられる。ツールの判定は Yes/No で示されるが、なぜ No なのかが重要であり、セキュリティの実力を調べるにはツールをモディファイして No の理由を調査するためのノウハウが必要である。加えて評価をするにあたって、どこをアタックするのかなどの見識が求められる。

また、長期間使用される自動車では、新たな攻撃手法があみだされ従来の守備方式が陳腐化する等、製造後に新たな脅威が発見されることが起こり得る。車両運用時における継続的なセキュリティを確保しなければならす、新たなセキュリティ脅威に長期間に亘り対応していくための人材育成が重要であると同時に、脆弱性情報を共有し、新たに発見されたソフトウェアの脆弱性をサプライチェーンの隅々にわたって修正・検証し運用中の車両へ対策済みソフトウェアを提供する仕組みが必須となる。同時に、ソフトウェアを提供するサプライチェーン自体のセキュリティ確保も重要な課題である。

現状は、車両へのペネトレーションテストを実施しようとした場合、海外の評価機関と比べて日本では十分な評価が行えないケースもあると見られており、先行する海外評価機関と同等の評価が可能となるための戦略が必要となる。

④ セキュリティ攻撃の研究と制度的支援

自動車のセキュリティ対応には、攻める側と守る側があるが、攻める側が時間、場所、攻撃対象を自由に選択できるのに対して、攻撃を受ける自動車は攻撃の対象・時期・手法が特定できず常に全てを守らなければならない不利な面を持っている。どう守っていくかを考えるためには、セキュリティ攻撃がどのように行われるかを研究し評価することが重要である。現在ではセキュリティ攻撃の検証という目的で、企業がスポンサーになり、製品の脆弱性を発見・報告してもらう見返りとして報奨金を支払う制度（Bug Bounty)が存在している。これには取り次ぐサイトを経由するケース、スポンサー企業が直接募集するケースがある。日本でもこうした取組みが広がっていくことが想定される。

脆弱性を見つけたときの対応について、ソフトウェアの不具合の場合には 2004 年に届出制度ができ、2014 年に改定されて現在に至っている。しかし適応範囲が不特定多数が使用するソフトウェア、及び Web に限定されており、例えば自動車への攻撃のように複合的な手段を利用する手口が見つかったときには受け取る組織がないことが課題と考えられる。

⑤ サプライチェーンのセキュリティ

自動車業界をとりまくセキュリティ状況の一つとして、サプライチェーンについて議論されていたセミナーの情報を研究会で共有した。自動車業界は巨大なサプライチェーンを抱えているが、第三者ソフト、オープンソースソフト、及び 3rd パーティ製品を使用する場合のセキュリティ・リスクについて、一番深い層まで保証しなければならないことが議論されており、自動車メーカーがどのように管理するかが課題と考えられる。

セキュリティ管理を末端まで浸透させることは困難であり、クラウドに依存するのがよいとの意見がある。仮に情報を一元管理した場合には万一漏洩した場合のリスク管理が重要であり、対策としてデータサイズを小さく分割し、複数個所に分散して格納することも一案と考えられる。

(2) 今後の課題

【海外・国内の調査】

自動車セキュリティに関して、検証が必要であることに異論は無いが、第三者評価・認証が必要か否かについては明確になっていない。評価や認証は利害関係者への説明責任の根拠となるものであり、規格の制定や制度化について国際的に協調していくためには、今後も海外動向等の情報収集を継続して実施することが重要である。

3.3　V2X 通信における署名検証の簡略化の研究（テーマ③）

平成 27 年度「V2X 等車外情報の活用にかかるセキュリティ技術の研究・開発」[1]にて、署名検証の簡略化方式の開発に係る机上検討として、簡略化方式の調査とともに評価と分析を行い、あるべき簡略化方式の在り方を挙げた。

本節では、この評価結果に基づき打ち出された「署名検証の簡略化方式」について、V2X 通信への適用を前提とした仮想環境を整え、署名検証の簡略化の効果について通信評価を行った結果を報告する。具体的には、仮想環境上での通信評価用機能の開発および構築を実施し、評価用データを生成の上、それを用いて簡略化方式を評価した。その際、署名検証のリアルタイム性確保については、平成 27 年度「V2X 等車外情報の活用にかかるセキュリティ技術の研究・開発」に従い、V2X 車載器が受信するメッセージ数として 1 秒あたり 1,000 程度を指標とした。また、V2X 通信に関連する標準化動向調査として、国際標準化ワーキンググループである ISO/TC204/WG16 会合の調査結果を報告する。

本評価にて、署名検証の簡略化方式による交差点右折のシナリオで、目標処理性能である 1,000messages/s を達成できることを示す。

3.3.1　V2X 通信のメッセージ検証処理と簡略化方式

本項では､本節の理解に必要となる､テーマ③の研究対象とその動作について説明する。まず､テーマ③の研究対象である V2X 通信のメッセージ検証を説明した後､メッセージ検証の動作を規定する V2X 通信のセキュリティ規格 IEEE1609.2[2]の概要と､メッセージ検証の内部処理について説明する。その後、既存の簡略化方式である Verify-on-Demand[3] [4]方式と、テーマ③で提案する簡略化方式である優先度付きメッセージ検証方式について説明する。

(1) テーマ③の研究対象

テーマ③の研究対象は、V2X システムを構成する V2X 車載器のメッセージ検証である（図 3.3.1-1 参照）。V2X システムは、V2X 通信を利用して V2X 車載器と路側機が互いにメッセージを送受信しながら、例えば運転支援等のサービスをドライバに提供する。V2X 車載器や路側機は自車位置や周辺情報等を V2X 通信のメッセージとして周期的にブロードキャストする。そのため、一つの V2X 車載器の通信範囲に N 台の V2X 車載器や路側機がある場合、伝送効率を無視すれば、毎周期 N 個のメッセージを受信する。欧州の研究事例[5]では一つの V2X 車載器が 1 秒あたりに受信するメッセージ数を最大で 1,000 程度と予測しており、メッセージ検証のリアルタイム性の確保が重要な課題となっている。そこでテーマ③の研究対象をメッセージ検証（図 3.3.1-1 中の破線内）とした。

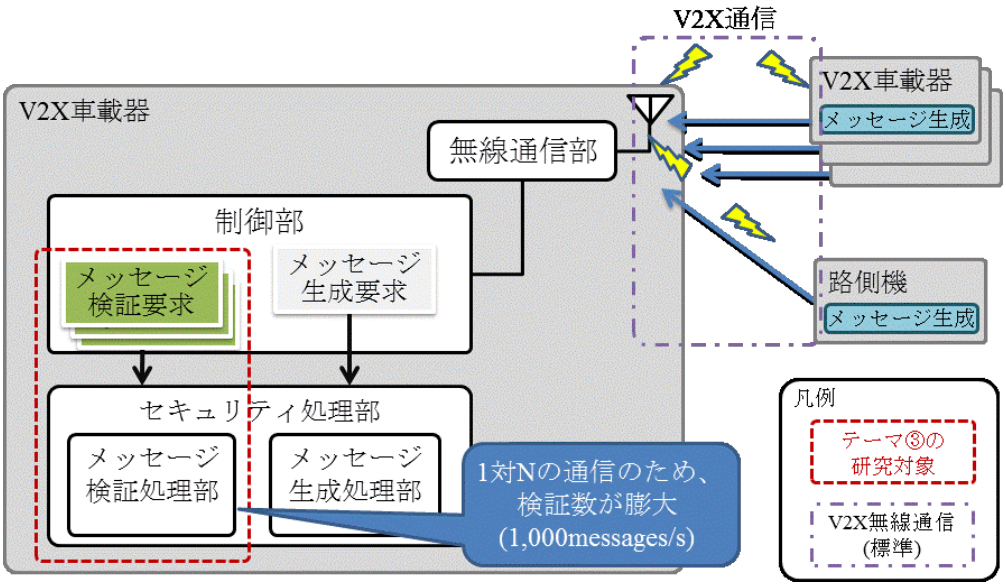


図 3.3.1-1　V2X システム構成図

テーマ③の研究対象であるメッセージ検証はキューを用いることを想定しており、次の様に動作する（図 3.3.1-2 参照）。制御部は(1)他の V2X 車載器から受信したメッセージを検証するための検証要求を生成しセキュリティ処理部に送信する。セキュリティ処理部は(2)受信した検証要求をキューに入れる一方で、(3)キューから検証要求を取り出し、メッセージ検証処理を行う。メッセージ検証が終了した際には、(4)その結果を制御部に送信する。(5)制御部は受信した検証結果を検証結果キューに入れる一方で、(6)検証結果キューから検証結果を取り出してアプリケーション処理を行う。

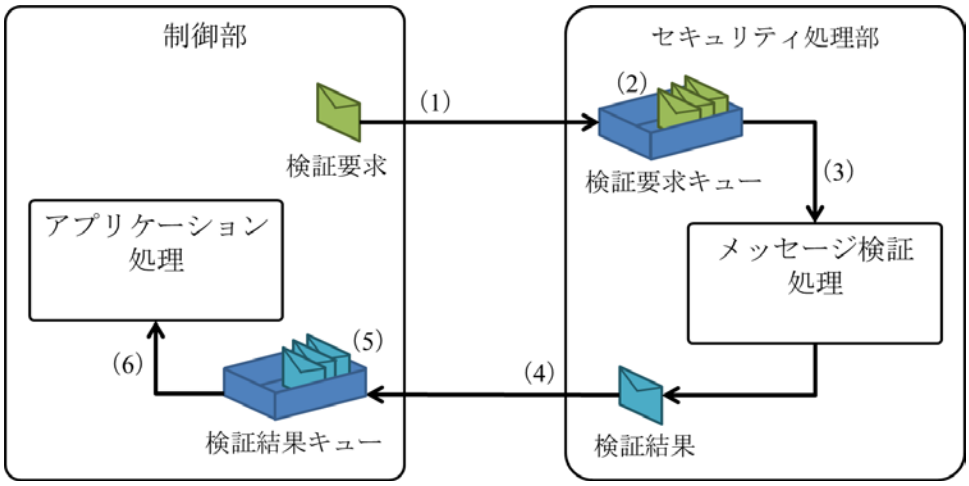


図 3.3.1-2　メッセージ受信時の V2X 車載器の動作

(2) V2X 通信のセキュリティ規格（IEEE1609.2）の概要

IEEE1609.2 は、V2X 通信のアーキテクチャやインタフェース等の標準を定める IEEE 1609 ファミリの一つであり、セキュリティに関するプロトコルやフォーマット、サービス

等を規定する規格である。IEEE1609.2 の特徴は、不特定多数の車両と通信することから PKI（Public Key Infrastructure）を使用することや、V2X 通信は比較的帯域の狭い通信であることから独自フォーマットの証明書等を用いることである。

IEEE1609.2 が規定するメッセージフォーマットは 1609Dot2Data という型の構造体として定義されている。1609Dot2Data で定義されるメッセージのタイプとして、署名付きメッセージ、暗号化メッセージ、証明書失効リスト（CRL：Certificate Revocation List）等がある。

(3) 署名付きメッセージの構造

IEEE1609.2 の署名付きメッセージのフォーマット（概略）を図 3.3.1-3 に示す。

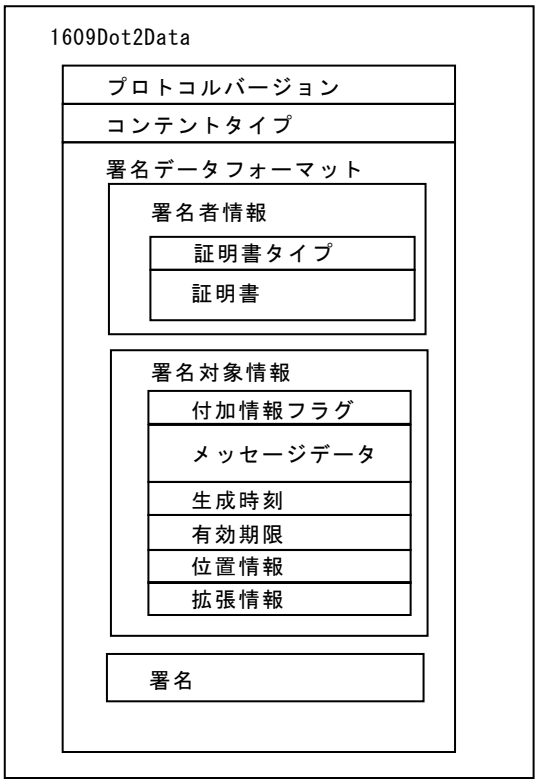


図 3.3.1-3　IEEE1609.2 の署名付きメッセージのフォーマット（概略）

プロトコルバージョンには、バージョン情報を示す値が格納される。

コンテントタイプにはメッセージのタイプ（署名付きメッセージ、暗号化メッセージ等）を示す値が格納され、そのタイプに応じて後続のデータ構造が定まる。コンテントタイプが署名付きメッセージを示す場合、後続のデータ構造は署名データフォーマットになる。

署名データフォーマットは、署名者情報、署名対象情報、署名の 3 つから構成される。署名者情報は証明書タイプと証明書から構成され、証明書タイプには後続のデータが証明書、証明書チェインもしくは証明書の ID のいずれであるのかを示す値が格納される。署名対象情報は付加情報フラグ、メッセージデータ、生成時刻、有効期限、位置情報、拡張

情報等から構成される。付加情報フラグは、署名対象情報の中に生成時刻、有効期限、位置情報、拡張情報等の各情報が存在するかどうかを示す。生成時刻は本メッセージが生成された時刻を、有効期限は本メッセージの有効期限を、位置情報は本メッセージが生成された位置を、それぞれ示している。署名は署名対象情報に対する署名値が格納されている。

(4) 署名付きメッセージの検証処理

前段で述べた署名付きメッセージを検証する際の処理について、説明する。本節では、便宜上、内部処理を計 9 つの Step に細分化した（図 3.3.1-4）。

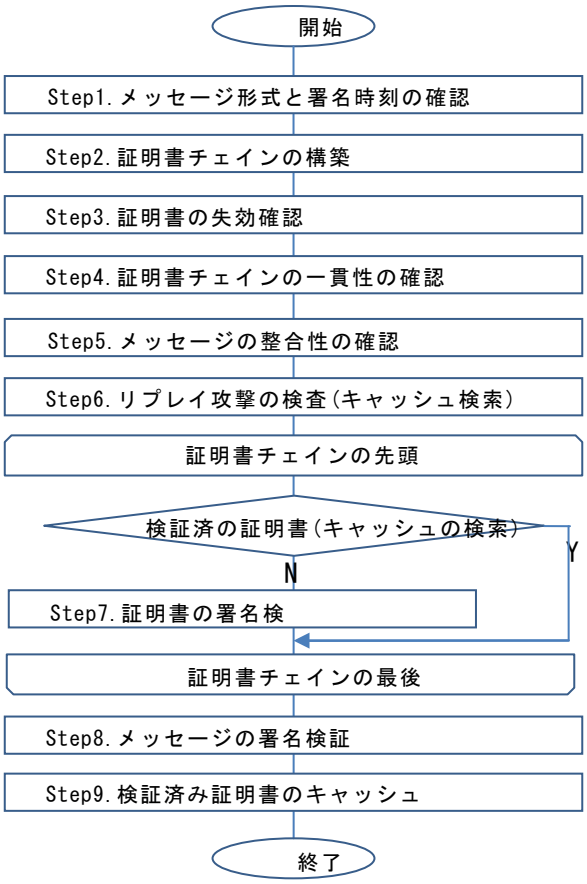


図 3.3.1-4　IEEE1609.2 の署名付きメッセージの検証処理フロー

各 Step の処理内容の詳細について表 3.3.1-1 に示す。

表 3.3.1-1　各 Step の処理内容の詳細

Step	処理の詳細
Step1 （形式と時刻の確認）	メッセージのコンテントタイプが署名付きメッセージを示す値であるかを確認する。また、メッセージの有効期限が過ぎていないか、生成時刻と有効期限の時系列が不正でないか等を確認する。
Step2 （チェイン構築）	メッセージ内および機器内に格納されている証明書を用いて、送信者の証明書から Root 証明書までのパスを構築する。
Step3 （失効確認）	Step2（チェイン構築）でパスを構築した証明書が失効していないかを検証する。
Step4 （チェインの一貫性確認）	パスを構築した証明書チェインの証明書に記載されている有効期限や有効範囲等について証明書間の一貫性を確認する。
Step5 （メッセージの整合性確認）	メッセージ中に格納された生成時刻、有効期限、位置情報等と証明書の有効期限や有効範囲等からその整合性を確認する。
Step6 （リプレイ攻撃検査）	リプレイ攻撃への対処のため、過去に受信したメッセージ情報のキャッシュを検索し、同じメッセージを受信済みでないかを確認する。未受信の場合は検索終了後に当該メッセージの情報をキャッシュへ登録する。
Step7 （証明書の署名検証）	検証済みの証明書情報のキャッシュを検索し、存在しない場合は証明書の署名検証を行う。
Step8 （メッセージの署名検証）	メッセージの署名検証を行う。
Step9 （証明書のキャッシュ）	検証の済んだ証明書情報を機器内のキャッシュに保持する。

(5) 簡略化方式～Verify-on-Demand 方式～

IEEE1609.2 の署名付きメッセージの検証処理では、図 3.3.1-5 に示すとおり、受信した全てのメッセージについて、メッセージ検証部でメッセージの署名検証を行う。そして、正しい署名の付いたメッセージに対してのみ、重要度の判定等、メッセージの内容を解析する。

しかしながら、このメッセージの署名検証処理には、メッセージ署名検証等、多くの処理時間を要するものがある。そのため、多くのメッセージを受信する環境では、全てのメッセージを処理できない問題が生ずる。そこで、米国において Verify-on-Demand 方式という方式が提案されている。

Verify-on-Demand 方式では、図 3.3.1-6 に示すとおり、まず先にメッセージの内容を見て重要度を判定し、重要度の高いメッセージについてのみメッセージ検証処理を行い、重要度が低いと判断されたメッセージは処理をせずに破棄する。重要度の判定では、例えばユーザーに警告通知を行ったり安全運転制御を動作させたりする必要のあるメッセージかどうかを判断基準とする。これにより、多くのメッセージを受信する環境であっても、重要

なメッセージを迅速に処理することが可能となる。

また、図 3.3.1-6 からも分かるとおり、Verify-on-Demand 方式は、メッセージフォーマットの変更を必要としないため、既存のプロトコルとの親和性も高い。

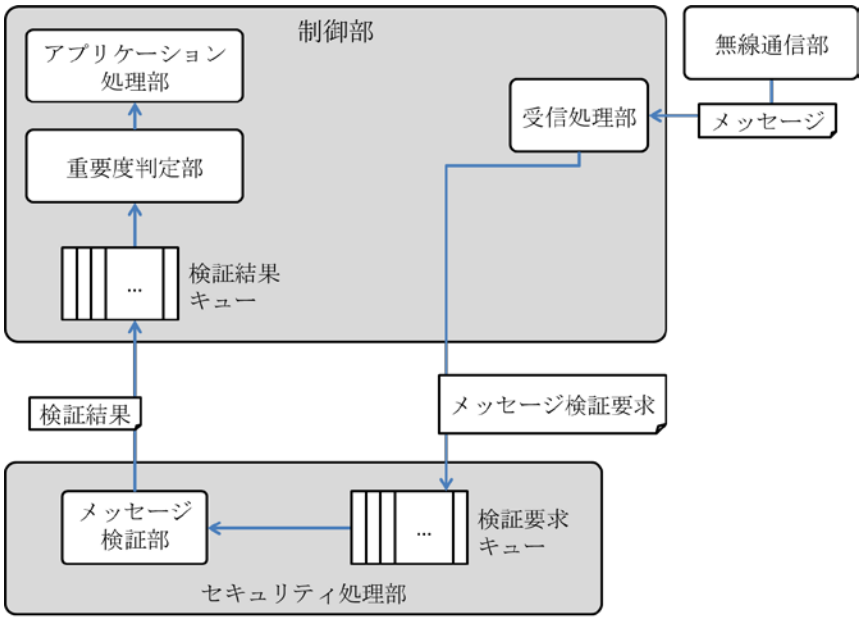


図 3.3.1-5　従来の処理フロー

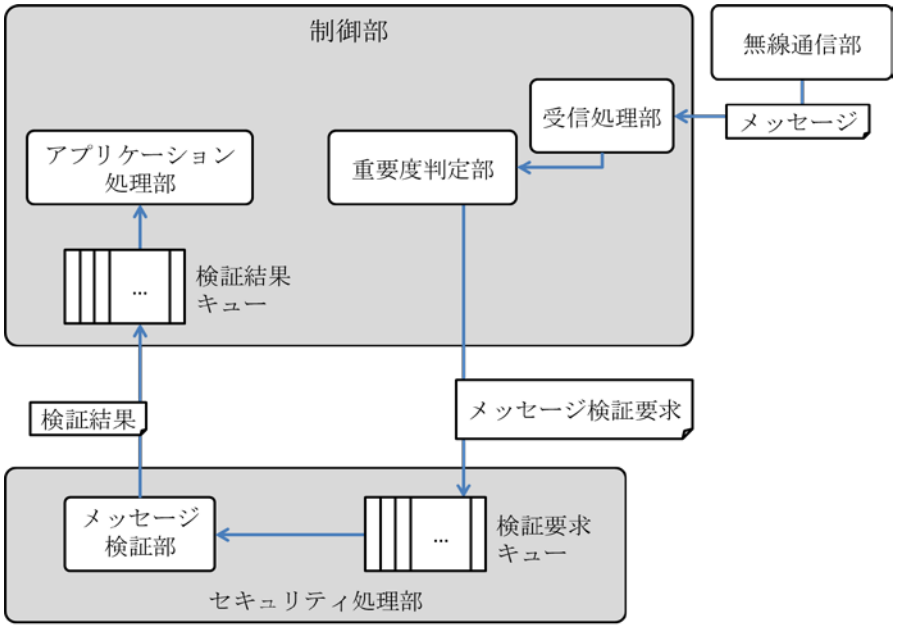


図 3.3.1-6　Verify-on-Demand 方式の処理フロー

(6) 簡略化方式～優先度付きメッセージ検証方式～

テーマ③では、平成 27 年度「V2X 等車外情報の活用にかかるセキュリティ技術の研究・開発」において、Verify-on-Demand 方式よりも高度な DoS 攻撃への対応が可能な優先度付きメッセージ検証方式を提案した（図 3.3.1-7 参照）。

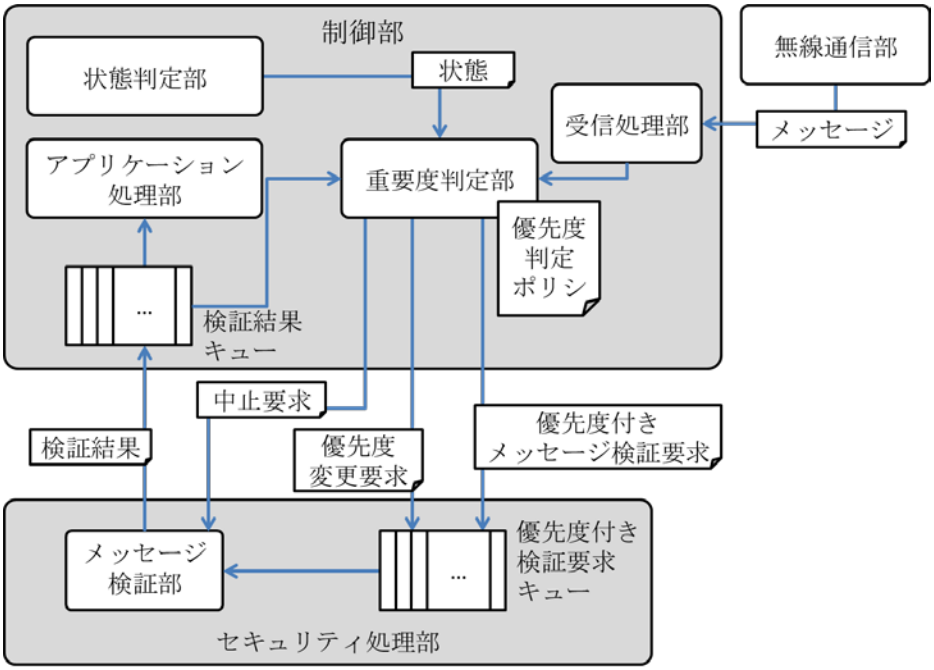


図 3.3.1-7　簡略化モデル（優先度付きメッセージ検証）

この優先度付きメッセージ検証方式は、Verify-on-Demand 方式と比較して、車両や車載器の状態を取得する「状態判定部」が追加され、検証要求発行順に従って検証を行っていた「検証要求キュー」が優先度に従って検証を行う「優先度付き検証要求キュー」に換わる。また、Verify-on-Demand 方式では検証要否のみを判定していた「重要度判定部」は、「優先度判定ポリシ」と「状態判定部」から得た「状態」や「検証結果キュー」の検証結果を用いて、受信したメッセージの優先度を判定し、優先度付きメッセージ検証要求を発行する。「重要度判定部」は、「優先度付き検証要求キュー」に対して優先度変更要求を用いて、発行済の優先度付き検証要求の優先度の変更や削除も行える。また、「重要度判定部」は中止要求により「メッセージ検証部」に対して実施中のメッセージ検証の中断を要求できる。

このようにすることで車両の状態や負荷に応じて検証する順番を動的に変更できる。また、図 3.3.1-7 からも分かるとおり、優先度付きメッセージ検証方式は、メッセージフォーマットの変更を必要としないため、既存のプロトコルとの親和性も高い。

3.3.2　V2X 通信におけるメッセージ検証簡略化方式の評価

本項では、V2X 通信への適用を前提とした仮想環境を整え、優先度付きメッセージ検証方式の効果について通信評価を行い、他の方式と比較した結果を報告する。本年度の評価では、V2X 通信の仮想環境として、交通量が多く、交差交通による事故のリスクが高いシナリオと想定される、交差点右折のケースについて検討した。また簡略化の効果を評価するために、リアルタイム性の評価として、処理スループットおよび処理遅延について評価した。また、セキュリティの評価として、単純な DoS 攻撃、リプレイ攻撃、高度な DoS 攻撃の 3 つの攻撃パターンに対する耐性を評価した。

(1) V2X 通信におけるメッセージ検証簡略化方式の評価項目と評価条件

ここでは、まず、仮想環境として模擬した交差点右折シナリオについて説明する。次に、評価に用いた簡略化方式の概要を説明の上、評価項目と目標値を示す。評価項目および目標値に関しては、リアルタイム性の評価とセキュリティの評価についてそれぞれ示す。

① V2X 通信におけるメッセージ検証簡略化方式の評価のシナリオと簡略化方式

本評価で模擬した交差点右折シナリオ（以下、評価シナリオという）は、片側 3 車線十字路（信号あり）において、交差点進入後の対向車の通過待ちから右折完了までの 7.0 秒間である。7.0 秒間は、各車両のメッセージ送信周期（100ms）に対して 70 周期に相当する。ここで、通信可能範囲に存在する車両は、目標処理性能である 1,000[message/s]を評価するために、100 台以上である必要がある。これに対し、本評価では、後述のデータ生成プログラムの上限である 200～218 台の車両が通信可能範囲に存在する想定とした。

評価シナリオのイメージを図 3.3.2-1 に示す。

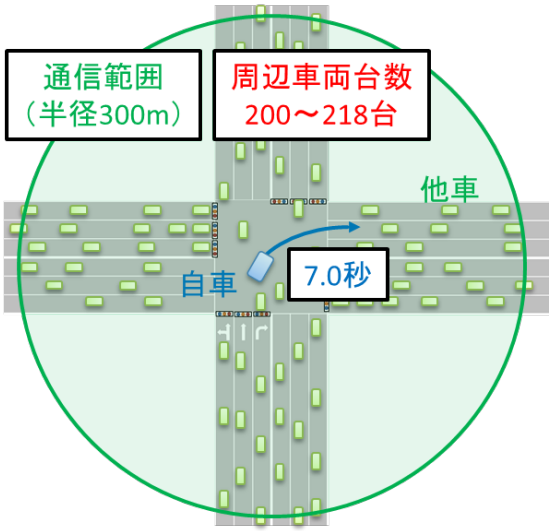


図 3.3.2-1　評価シナリオ（イメージ）

本評価では、以下 3 つの方式を用いる。

・簡略化方式を持たない場合 ～簡略化方式なし～

・簡略化方式～Verify-on-Demand 方式～

・簡略化方式～優先度付きメッセージ検証方式～

各方式のメッセージ処理フローは、3.3.1 項で示した内容から図 3.3.2-2 のとおりに示せる。図 3.3.2-2 において Proc.1～Proc.5 の処理内容を示す。

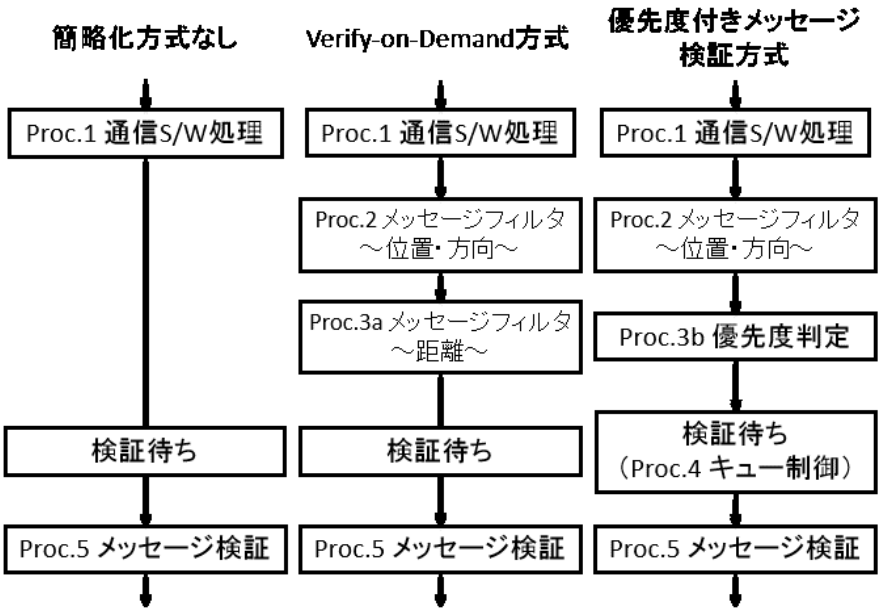


図 3.3.2-2 簡略化方式とメッセージ処理フロー

・Proc.1 通信 S/W 処理

V2X 通信規格である IEEE1609.3[6]に相当する処理を実行する。

・Proc.2 メッセージフィルタ～位置・方向～

Proc.2 では、交差点右折の判断に必要なメッセージか否かを判定する。具体的には、対向車両および右折後先行車両からのメッセージは必要と判定され、その他のメッセージは破棄される。また、対向車両の内、右折走行中に近方まで接近しないと予想される車両からのメッセージは破棄される。近方か否かの判定基準は、国交省の通信利用型実用化システム基本設計書[7]に記載の情報提供開始位置（80m）を基にしている。

本評価では、地図情報を取得できるものとし、メッセージ内に含まれる緯度、経度や進行方向、車速を基に判定される。

・Proc.3a メッセージフィルタ～距離～

Proc.3a では、他車との距離に応じて交差点右折の判断に必要なメッセージか否かを判定する。具体的には、前述の情報提供開始位置である 80m 以内に位置する車両からのメッセージは必要と判定され、その他のメッセージは破棄される。Proc.3a は、Verify-on-Demand 方式特有の処理である。

・Proc.3b 優先度判定～距離～

Proc.3b では、他車との距離に応じてメッセージ処理の優先度を判定する。本評価では、他車との距離が近いほど高優先度と判定される。Proc.3b は、優先度付きメッセージ検証方式特有の処理である。

・Proc.4 キュー制御

Proc.4 では、検証要求キューに積まれたメッセージに対して、1. 再優先度判定、2. 同一車両メッセージの削除、3. タイムアウト判定を実行する。本処理は自車位置が更新されるタイミングで実行される。なお、自車位置は走行状態にかかわらず定期的に更新される。

1. 再優先度判定処理では、Proc.3b の判定処理を現在の自車位置に従って再度実行する。2. 同一車両メッセージの削除処理では、キュー内に含まれる同一車両からのメッセージの内、最新のメッセージを残し、その他古い送信時刻を持ったメッセージを削除する。3. タイムアウト判定処理では、受信時から 100ms 以上経過したメッセージを削除する。

・Proc.5 メッセージ検証

3.3.1 項(4)で示した検証処理を実行する。

② リアルタイム性に関する評価項目と目標

リアルタイム性に関する評価では、V2X メッセージに対する処理性能を評価する。評価条件は、前述の評価シナリオとする。

リアルタイム性に関する評価の対象、実施内容および評価項目は次のとおりである。

(ⅰ)評価対象

次の 3 つの方式についてそれぞれ評価する。

- 簡略化方式を持たない場合 ～簡略化方式なし～
- 簡略化方式～Verify-on-Demand 方式～
- 簡略化方式～優先度付きメッセージ検証方式～

(ⅱ)実施内容

表 3.3.2-1 リアルタイム性に関する評価の実施内容

実施項目	説明
交差点右折シナリオ(評価シナリオ)	通信可能範囲に存在する 200～218 台の内、各周期 110 台がメッセージを送信可能と想定（欧州の研究事例よりも厳しい条件を設定）し、1 秒間の受信メッセージ数を 1,100 とした。 110 台は毎周期ランダムに決定した。

(iii)評価項目

次の 2 つの項目を評価する。

・評価シナリオ全体を通してのリアルタイム性

・メッセージ当たりの処理リアルタイム性

評価シナリオ全体を通してのリアルタイム性は、処理スループットを用いて評価する。実世界の V2X システムで想定される 1 秒間当たりのメッセージ受信数は、欧州の研究事例から 1,000 メッセージと想定される。そのため本評価における処理スループットの目標値を 1,000messages/s 以上と設定した。

メッセージ当たりの処理リアルタイム性は、メッセージ受信後からメッセージ検証完了までの処理時間を用いる。各車両のメッセージ送信周期は 100ms であることから、本評価における最大処理時間の目標値を 100ms 以下と設定した。Verify-on-Demand 方式と優先度付きメッセージ検証方式を公平に評価するため、メッセージ当たりの処理リアルタイム性を、他車との距離（以下、他車距離という）が 80m 以内と判定されるメッセージに限定して評価する。以降、このメッセージを要検証メッセージと呼ぶ。また、要検証メッセージと判定することを要検証と判定するという。

リアルタイム性に関する評価項目と目標値を表 3.3.2-2 にまとめた。

表 3.3.2-2　リアルタイム性に関する評価項目と目標値

評価項目	評価内容	目標値
評価シナリオ全体を通してのリアルタイム性	評価シナリオ全体を通しての処理スループット	1,000 以上 [messages/s]
メッセージ当たりの処理リアルタイム性	要検証メッセージにおける最大処理時間	100 以下 [ms]

ここで、本評価における処理スループットは、評価シナリオ中の受信メッセージのうち、処理したメッセージ数から、それらの処理に要した時間を割ることで算出した。本評価における処理メッセージとは、次の 5 つのメッセージである。

・Proc.2 メッセージフィルタ～位置・方向～により、「検証不要」と判定され、破棄されたメッセージ

・Proc3a メッセージフィルタ～距離～により、「検証不要」と判定され、破棄されたメッセージ

・Proc.4 キュー制御における同一車両メッセージ判定により、削除されたメッセージ

・Proc.4 キュー制御におけるタイムアウト判定により、削除された要検証ではないメッセージ

・Proc.5 メッセージ検証による処理を完了したメッセージ

これら 4 つ以外のメッセージ、例えば、予期せず破棄されたメッセージやキューからの溢れたメッセージ、タイムアウト判定された要検証メッセージなどは、処理メッセージとは扱わない。つまり、処理できなかったメッセージとして扱う。

③ セキュリティに関する評価項目と目標

セキュリティに関する評価では外部からの攻撃に対する耐性を評価する。評価条件は、前述の評価シナリオとする。具体的な攻撃は、サービス妨害となる DoS 攻撃とリプレイ攻撃である。目標を要検証メッセージが攻撃によって破棄または遅延しないこととする。

セキュリティに関する評価の対象、攻撃パターンおよび評価項目は次のとおりである。

(ⅰ) 評価対象

次の 3 つの方式について評価し、比較する。

・簡略化方式を持たない場合～簡略化方式なし～
（IEEE 1609.2 のリプレイ対策機能は有効化）

・簡略化方式～Verify-on-Demand 方式～

・簡略化方式～優先度付きメッセージ検証方式～

(ⅱ) 攻撃パターン

次の攻撃パターンで評価する。

表 3.3.2-3　セキュリティ評価の攻撃パターン

項目	説明
単純な DoS 攻撃	任意のメッセージを大量に送信。次の 2 つのパターンを計測。 • 他車 1 台からの大量のメッセージ送信 1 秒間の受信メッセージ数: 1,200、1,600、2,000 • 他車複数台からの同時メッセージ送信（DDoS） 1 秒間の受信メッセージ数: 1,200、1,600、2,000 (各攻撃メッセージについて、シナリオ中の他車をランダムに送信元として選択)
リプレイ攻撃	過去に送信された任意のメッセージを大量に再送。 再送するメッセージの選択およびタイミングは、本来のメッセージが送信された以降においてランダムに決定。1 秒間の受信メッセージ数が 1,200、1,600、2,000 になる場合で計測。
高度な DoS 攻撃	各簡略化方式にて、高い優先度と判定されるようなメッセージを大量に送信。以下の A）～C）の 3 つのパターンを計測。いずれも 1 秒間の受信メッセージ数が 1,200、1,600、2,000 になる場合で計測。 A) 要検証と判定される境界のメッセージを大量送信。つまり、図 3.3.2-2 の Proc.2 と Proc.3a のフィルタ基準を満たし、現在位置が丁度 80m となる車からのメッセージを大量に送信。 B) 要検証メッセージを大量に送信。つまり、図 3.3.2-2 の Proc.2 と Proc.3a のフィルタ基準を満たし、現在位置が 0～80m となる車からのメッセージをランダムに大量に送信。 C) 攻撃者が攻撃対象の車両の走行動作を想定して、攻撃対象車両の右折経路上に車両が停止していると見せかけたメッセージを大量に送信（A、B と異なり他車の現在位置には注目せずに攻撃）。

(ⅲ) 評価項目

各攻撃により、要検証メッセージがどの程度破棄または遅延するかを評価する。ただし、攻撃に用いたメッセージは、要検証メッセージには含めない。具体的には、表 3.3.2-4 の項目を評価する。

表 3.3.2-4　セキュリティに関する評価項目と目標値

評価観点	評価項目	目標値
要検証メッセージ破棄率	評価シナリオ中に受信した要検証メッセージが攻撃により破棄された割合	0%
要検証メッセージ処理時間の平均値	シナリオ中に受信した要検証メッセージの処理時間の平均値	100 以下［ms］

なお、評価で使う攻撃を DoS 攻撃およびリプレイ攻撃としたのは、平成 27 年度「V2X 等車外情報の活用にかかるセキュリティ技術の研究・開発」での机上検討の結果からである。この結果では、V2X 通信の簡略化方式に関係する主な脅威は、DoS 攻撃とリプレイ攻撃とであることが報告されている。評価シナリオでは、偽のメッセージの送信や改ざんも可能であるが、メッセージの署名検証でアプリケーションに渡る前に攻撃を検知可能であるため今回は評価しない。他にも正当な車載器や路側機が不正なメッセージを送信する脅威も考えられる。しかし、この脅威はメッセージ検証だけで防ぐことは困難であるため評価の対象外とする。

(2) 簡略化方式の評価環境の構築と評価用データ

ここで、簡略化方式の評価に使用した評価環境について説明する。評価環境の全体構成を図 3.3.2-3 に示す。評価環境は、PC を用いて構築したデータ生成装置と評価用基板を用いて構築した測定装置の 2 つの装置で構成される。データ生成装置は、評価用データ生成プログラムにより、評価条件に応じた評価用データを生成する。測定装置は、評価測定プログラムにより、評価用データを読み込み、簡略化方式対応 V2X プログラムを動作させ、その性能を測定した。

簡略化方式対応 V2X プログラムと評価測定プログラムの動作イメージを図 3.3.2-4 に示す。簡略化方式対応 V2X プログラムは先に示した Proc.1～Proc.5 の処理を実行する。評価測定プログラムは、図 3.3.2-4 に示す時刻計測①～⑥のとおり、Proc.1～Proc.5 の処理開始時間と処理完了時間の時間を計測する。例えば、時刻計測②では Proc.1 の処理完了時間を計測する。

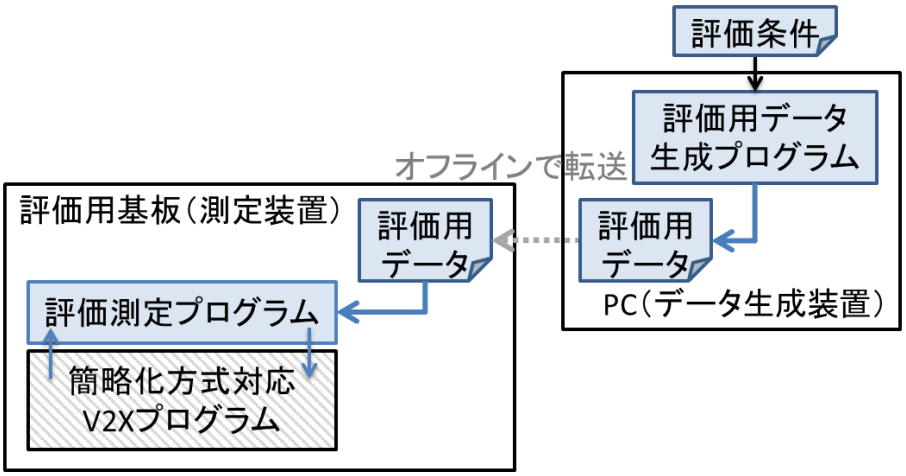


図 3.3.2-3　評価環境の全体構成

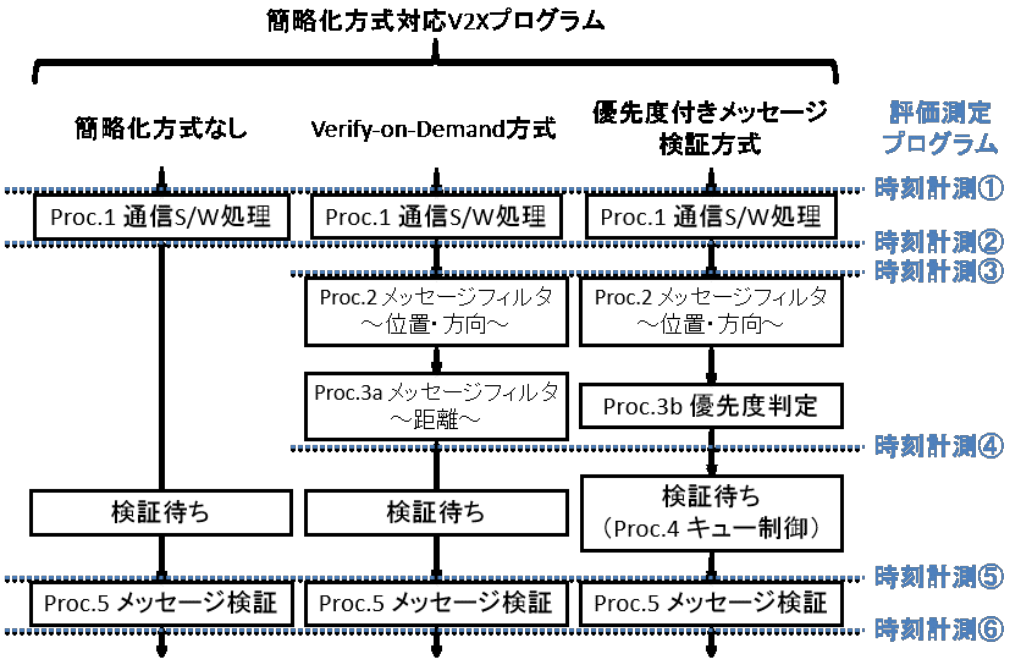


図 3.3.2-4 簡略化方式対応 V2X プログラムと評価測定プログラムの動作イメージ

本評価で使用した測定装置の仕様を表 3.3.2-5 に示す。測定装置（評価用基板）には、市販の V2X 車載器向けマイコンが搭載されている。署名検証処理の時間が一律 5ms となるようにマイコンを設定し、署名検証処理を擬似的に実現した。なお、検証要求キューのキューサイズは、32 メッセージ分である。

表 3.3.2-5　測定装置の仕様

項目	仕様	備考
マイコン仕様	動作周波数：1 GHz メモリ（RAM）：1 GB	暗号 H/W 有り （署名検証処理時間 5ms）
OS	Linux OS	

評価用データの概要を図 3.3.2-5 に示す。評価用データは、V2X メッセージに評価用タグが付与された形で構成される。評価測定プログラムは評価用タグに記述されている処理開始タイミングに従って、メッセージ処理フローを開始する。この様にすることで、測定装置において、他車からの V2X メッセージ受信を仮想的に実現した。

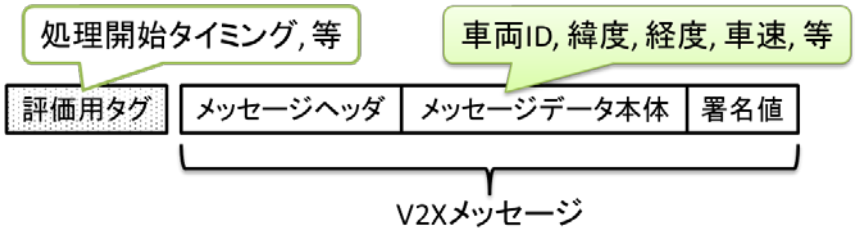


図 3.3.2-5　評価用データ概要

(3) 簡略化方式の通信評価

① 簡略化方式なしのリアルタイム性の評価

ここでは、簡略化方式の効果を明確にするための予備評価として、簡略化方式を持たない場合（以下、簡略化方式なしという）のリアルタイム性の評価結果を示す。表 3.3.2-6 に簡略化方式なしの評価結果を示す。本結果から、簡略化方式なしでは、評価シナリオにおいて、目標とするリアルタイム性を確保できないことが分かった。表 3.3.2-7 には簡略化方式なしのリアルタイム性評価結果の詳細を示す。

表 3.3.2-6　簡略化方式なしのリアルタイム性評価結果

評価項目	目標値	評価結果
シナリオ全体を通しての処理スループット	1,000 以上[messages/s]	98.5 [messages/s]
要検証メッセージの最大処理時間	100 以下[ms]	434.7 [ms]

表 3.3.2-7 簡略化方式なしのリアルタイム性評価結果の詳細

補足評価項目	評価結果
受信メッセージ数	7,700 [messages]
処理メッセージ数	717 [messages]
メッセージの処理割合	9.31 [%]
検証メッセージ数	717 [messages]
受信した要検証メッセージ数	429 [messages]
処理した要検証メッセージ数	29 [messages]
要検証メッセージの処理割合	6.76 [%]
要検証メッセージの平均処理時間	313.7 [ms]
Proc.1 通信 S/W 処理の平均処理時間 【時刻計測②－時刻計測①】	1.58 [ms]
証明書キャッシュヒット時の Proc.5 メッセージ検証の平均処理時間 【時刻計測⑥－時刻計測⑤】	7.61 [ms]
証明書キャッシュミス時の Proc.5 メッセージ検証の平均処理時間 【時刻計測⑥－時刻計測⑤】	14.9 [ms]
その他メッセージ検証待ちや OS 処理に係る平均処理時間【時刻計測⑤－時刻計測②】	302.7 [ms]
証明書キャッシュヒット率	74.8 [%]

簡略化方式なしでは、受信メッセージ数 7,700messages に対し、処理メッセージは 717messages であった。これは全体の 9.31%しか処理できていない。また、Proc.5 メッセージ検証の処理に到達したメッセージ数である検証メッセージ数は、処理メッセージ数と同様 717messages であった。これは、本評価の測定環境のハードウェア性能では、およそ 700messages を処理することが限界であることを示している。

要検証メッセージに着目すると、簡略化方式なしでは、要検証メッセージ全体の 6.76%しか処理できないことがわかった。要検証メッセージの平均処理時間は 313.7ms であり、評価シナリオ全体を通してリアルタイム性が低いことが分かった。

本評価結果から、実際の V2X システムにおいては、簡略化方式が必要であると結論付けられる。

また、国交省の通信利用型実用化システム基本設計書[7]によると、高い支援レベルを提供すべきメッセージに対して、より低遅延な処理時間が求められている。そこで、簡略化方式なしにおける他車距離とメッセージ平均処理時間の関係を図 3.3.2-6 に示す。評価シナリオ中に 20m-0m 内に接近する車両は存在しなかったため、図 3.3.2-6 ではプロットされていない。図 3.3.2-6 から簡略化方式なしでは、他車距離に関わらず 310ms 程度要することが分かった。

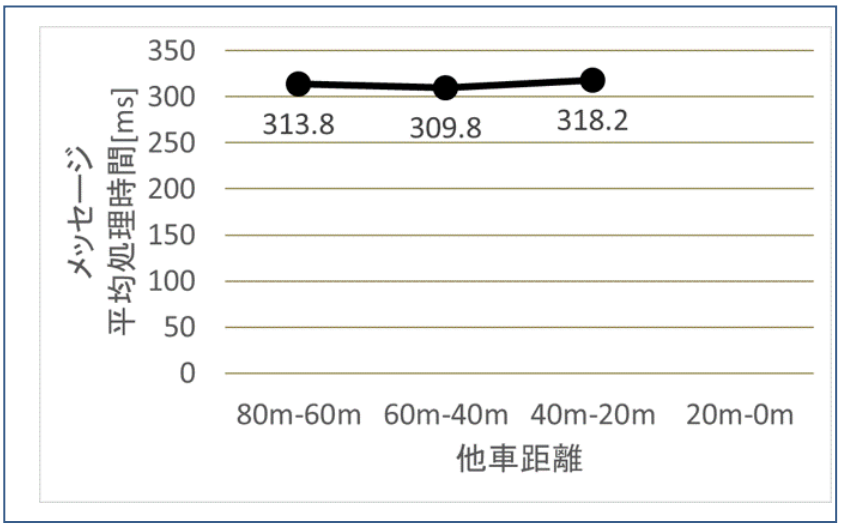


図 3.3.2-6　簡略化方式なしの他車距離とメッセージ平均処理時間の関係

② Verify-on-Demand 方式のリアルタイム性の評価

ここでは、Verify-on-Demand 方式のリアルタイム性に関する評価結果を示す。表 3.3.2-8 にリアルタイム性の評価項目および目標値と、Verify-on-Demand 方式における評価結果を示す。本結果から、Verify-on-Demand 方式では、評価シナリオにおいて、目標のリアルタイム性能を達成できることが分かった。表 3.3.2-9 には Verify-on-Demand 方式のリアルタイム性に関する評価結果の詳細を示す。

表 3.3.2-8 Verify-on-Demand 方式のリアルタイム性評価結果

評価項目	目標値	評価結果
評価シナリオ全体を通しての処理スループット	1,000 以上 [messages/s]	1,097.4 [messages/s]
要検証メッセージの最大処理時間	100 以下 [ms]	44.7 [ms]

表 3.3.2-9　Verify-on-Demand 方式のリアルタイム性評価結果の詳細

補足評価項目	評価結果
受信メッセージ数	7,700 [messages]
処理メッセージ数	7,700 [messages]
メッセージの処理割合	100 [%]
検証メッセージ数	429 [messages]
受信した要検証メッセージ数	429 [messages]
処理した要検証メッセージ数	429 [messages]
要検証メッセージの処理割合	100 [%]
要検証メッセージの平均処理時間	14.6 [ms]
Proc.1 通信 S/W 処理の平均処理時間 【時刻計測②－時刻計測①】	1.51 [ms]
Proc.2 メッセージフィルタ～位置・方向～ および Proc.3a メッセージフィルタ～距離～ 【時刻計測④－時刻計測③】	0.03 [ms]
証明書キャッシュヒット時の Proc.5 メッセージ検証の平均処理時間 【時刻計測⑥－時刻計測⑤】	7.47 [ms]
証明書キャッシュミス時の Proc.5 メッセージ検証の平均処理時間 【時刻計測⑥－時刻計測⑤】	14.7 [ms]
その他メッセージ検証待ちや OS 処理に係る平均処理時間【(時刻計測⑤－時刻計測④）＋（時刻計測③－時刻計測②)】	5.27 [ms]
証明書キャッシュヒット率	95.6 [%]

Verify-on-Demand 方式では、評価シナリオ中のすべての受信メッセージを処理できた。また、検証メッセージ数と処理した要検証メッセージ数が同数であることから、要検証メッセージのみをメッセージ検証処理したことが分かる。要検証メッセージの平均処理時間は 14.6ms であり、評価シナリオ全体を通したメッセージの処理リアルタイム性が高い。メッセージフィルタ処理（Proc.2 と Proc.3a）の平均処理時間は 0.03ms であり、メッセージ検証処理に対して極めて短い時間であることが分かる。

Verify-on-Demand 方式における他車距離とメッセージ平均処理時間の関係を図 3.3.2-7 に示す。Verify-on-Demand 方式では、他車距離に関わらず 14ms 程度で処理されている。

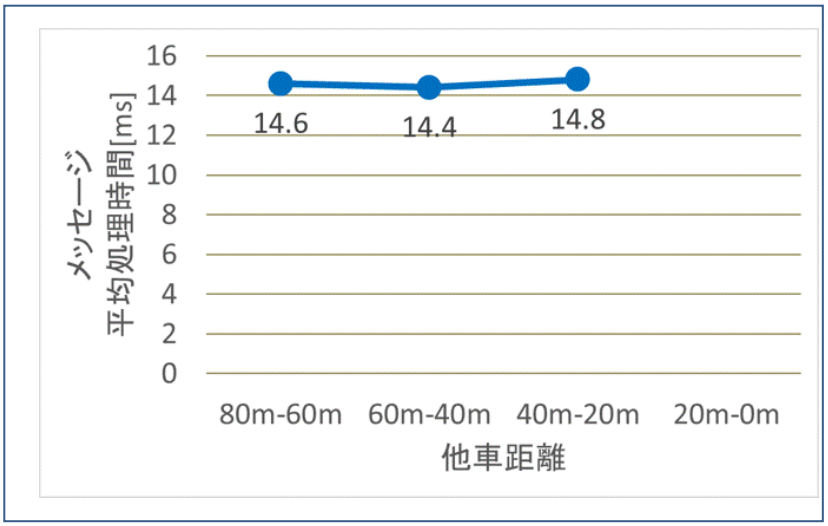


図 3.3.2-7　Verify-on-Demand 方式の他車距離とメッセージ平均処理時間の関係

③ 優先度付きメッセージ検証方式のリアルタイム性の評価

ここでは、優先度付きメッセージ検証方式のリアルタイム性評価結果を示す。表 3.3.2-10 にリアルタイム性の評価項目および目標値と、優先度付きメッセージ検証方式における評価結果を示す。本結果から、優先度付きメッセージ検証方式では、本評価シナリオにおいて、目標のリアルタイム性能を達成可能であることが示せた。表 3.3.2-11 に優先度付きメッセージ検証方式のリアルタイム性評価結果の詳細を示す。

表 3.3.2-10　優先度付きメッセージ検証方式のリアルタイム性評価結果

評価項目	目標値	評価結果
評価シナリオ全体を通しての処理スループット	1,000 以上[messages/s]	1,098.9 [messages/s]
要検証メッセージの最大処理時間	100 以下[ms]	73.5 [ms]

表 3.3.2-11　優先度付きメッセージ検証方式のリアルタイム性評価結果の詳細

補足評価項目	評価結果
受信メッセージ数	7,700 [messages]
処理メッセージ数	7,700 [messages]
メッセージの処理割合	100 [%]
検証メッセージ数	751 [messages]
受信した要検証メッセージ数	429 [messages]
処理した要検証メッセージ数	429 [messages]
要検証メッセージの処理割合	100 [%]
要検証メッセージの平均処理時間	24.5 [ms]
Proc.1 通信 S/W 処理の平均処理時間 【時刻計測②－時刻計測①】	1.72 [ms]
Proc.2 メッセージフィルタ～位置・方向～ および Proc.3b 優先度判定の平均処理時間 【時刻計測④－時刻計測③】	0.03 [ms]
証明書キャッシュヒット時の Proc.5 メッセージ検証の平均処理時間 【時刻計測⑥－時刻計測⑤】	7.46 [ms]
証明書キャッシュミス時の Proc.5 メッセージ検証の平均処理時間 【時刻計測⑥－時刻計測⑤】	14.7 [ms]
その他メッセージ検証待ちや OS 処理に係る平均処理時間【(時刻計測⑤－時刻計測④）＋（時刻計測③－時刻計測②)】	15.0 [ms]
証明書キャッシュヒット率	96.3 [%]

優先度付きメッセージ検証方式においても、評価シナリオ中のすべての受信メッセージを処理できた。一方、検証メッセージ数は 751messages であり、処理した要検証メッセージ数より多い。このことから、優先度付きメッセージ検証方式は、高い処理スループットを持ち、かつ、多くのメッセージ検証処理が可能である方式であることが分かった。

要検証メッセージにおける平均処理時間は 24.5ms であり、シナリオ全体を通したメッセージの処理リアルタイム性も高いことが分かる。位置・方向によるメッセージフィルタ処理および優先度判定処理（Proc.2 と Proc.3b）の平均処理時間は 0.03ms であり、メッセージ処理に対して少ない時間であることが分かった。

優先度付きメッセージ検証方式における他車距離とメッセージ平均処理時間の関係を図 3.3.2-8 に示す。優先度付きメッセージ検証方式では、他車距離が近いほどメッセージが高速に処理できる傾向にあることが分かる。

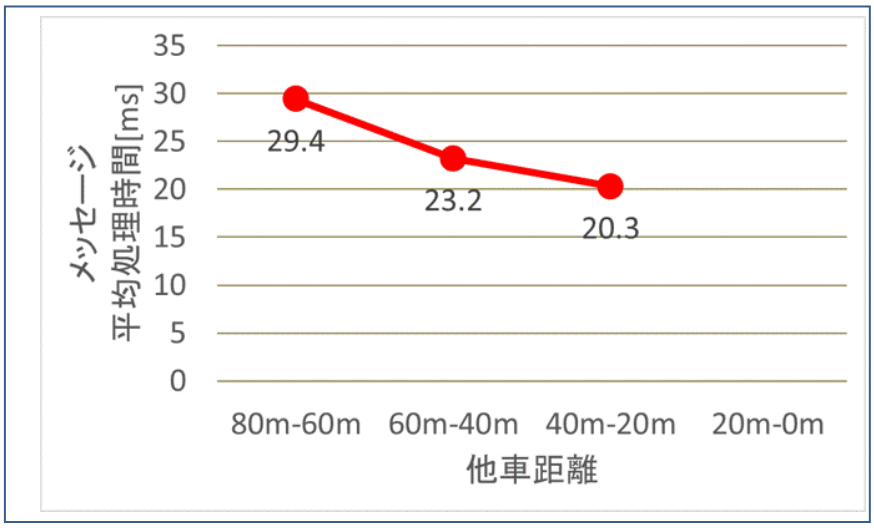


図 3.3.2-8　優先度付きメッセージ検証方式の他車距離とメッセージ平均処理時間の関係

④ 簡略化方式なしのセキュリティの評価

ここでは簡略化方式なしのセキュリティの評価結果およびその分析結果を説明する。表 3.3.2-12～3.3.2-17 に、評価の目標値および簡略化方式なしの各攻撃パターンに対する各評価項目の測定結果を示す。なお、表中の測定結果欄下の数字は、それぞれ 1 秒間の受信メッセージ数である。1,100 については攻撃メッセージを含まない状態であり、1,200～2,000 が攻撃メッセージを含む状態である。測定結果である数値が*斜字体*の場合は、目標値に対し未達であったことを示す（セキュリティ評価の攻撃パターンは表 3.3.2-3）。

本結果から、簡略化方式なしでは、評価シナリオにおいては、攻撃パターンによらず目標に対し未達となる。

表 3.3.2-12　簡略化方式なしのセキュリティ測定結果（単純な DoS:攻撃元同一）

評価観点	目標値	測定結果			
		1,100	1,200	1,600	2,000
要検証メッセージ破棄率［%］	0%	*93.24*	*92.09*	*97.20*	*98.60*
要検証メッセージ処理時間の平均値［ms］	100 以下	*313.67*	*327.22*	*352.48*	*400.78*

表 3.3.2-13　簡略化方式なしのセキュリティ測定結果（単純な DoS:攻撃元複数）

評価観点	目標値	測定結果			
		1,100	1,200	1,600	2,000
要検証メッセージ破棄率［%］	0%	*93.24*	*90.93*	*97.44*	*98.14*
要検証メッセージ処理時間の平均値［ms］	100 以下	*313.67*	*317.30*	*350.11*	*387.50*

表 3.3.2-14　簡略化方式なしのセキュリティ測定結果（リプレイ）

評価観点	目標値	測定結果			
		1,100	1,200	1,600	2,000
要検証メッセージ破棄率［%］	0%	*93.24*	*93.22*	*95.09*	*97.20*
要検証メッセージ処理時間の平均値［ms］	100 以下	*313.67*	*304.61*	*318.12*	*320.77*

表 3.3.2-15　簡略化方式なしのセキュリティ測定結果（高度な DoS:A）

評価観点	目標値	測定結果			
		1,100	1,200	1,600	2,000
要検証メッセージ破棄率［%］	0%	*93.24*	*88.79*	*95.79*	*97.20*
要検証メッセージ処理時間の平均値［ms］	100 以下	*313.67*	*314.19*	*347.55*	*408.72*

表 3.3.2-16　簡略化方式なしのセキュリティ測定結果（高度な DoS:B）

評価観点	目標値	測定結果			
		1,100	1,200	1,600	2,000
要検証メッセージ破棄率［%］	0%	*93.24*	*92.99*	*93.93*	*97.66*
要検証メッセージ処理時間の平均値［ms］	100 以下	*313.67*	*292.14*	*318.25*	*414.08*

表 3.3.2-17　簡略化方式なしのセキュリティ測定結果（高度な DoS:C）

評価観点	目標値	測定結果			
		1,100	1,200	1,600	2,000
要検証メッセージ破棄率［%］	0%	*93.24*	*92.33*	*94.87*	*98.59*
要検証メッセージ処理時間の平均値［ms］	100 以下	*313.67*	*311.94*	*352.20*	*371.30*

上記の結果から分かるとおりに、すべての攻撃パターンにおいて、簡略化方式なしの場合には目標を達成していない。簡略化方式なしでは、受信したメッセージを受信した順に処理する。このため、処理するメッセージが多い場合には、処理に時間がかかり後から受信したメッセージは検証要求キューが溢れてメッセージが破棄される。また、簡略化方式なしでは、メッセージの種類や負荷の状況等において、特定のメッセージを優先的に処理する仕組みがない。このため、受信するメッセージが多い場合には、安全運転に影響を与える要検証メッセージも破棄されることになる。

攻撃者が任意のメッセージを大量に送信し、通信負荷を上げることは容易である。つまり、実際の V2X システムにおいても、簡略化方式を持たない場合は容易に攻撃が可能であると言える。

⑤ Verify-on-Demand 方式のセキュリティの評価

ここでは Verify-on-Demand 方式のセキュリティの評価結果およびその分析結果を説明する。表 3.3.2-18～3.3.2-23 には、評価の目標値および Verify-on-Demand 方式における各攻撃パターンでの各評価項目の測定結果を示す。なお、表中の測定結果欄下の数字は、それぞれ 1 秒間の受信メッセージ数である。1,100 については攻撃メッセージを含まない状態であり、1,200～2,000 が攻撃メッセージを含む状態である。測定結果である数値が*斜字体*の場合は、目標値に対し未達であったことを示す（セキュリティ評価の攻撃パターンは表 3.3.2-3）。

本結果から、Verify-on-Demand 方式の場合には、単純な DoS 攻撃およびリプレイ攻撃では目標を達成できることが分かる。しかし、高度な DoS 攻撃ではその攻撃パターンや負荷状況によらず目標に対し未達となる。

表 3.3.2-18　Verify-on-Demand 方式のセキュリティ測定結果（単純な DoS:攻撃元同一）

評価観点	目標値	測定結果			
		1,100	1,200	1,600	2,000
要検証メッセージ破棄率［%］	0%	0	0	0	0
要検証メッセージ処理時間の平均値［ms］	100 以下	14.59	16.30	22.86	80.99

表 3.3.2-19　Verify-on-Demand 方式のセキュリティ測定結果（単純な DoS:攻撃元複数）

評価観点	目標値	測定結果			
		1,100	1,200	1,600	2,000
要検証メッセージ破棄率［%］	0%	0	0	0	0
要検証メッセージ処理時間の平均値［ms］	100 以下	14.59	13.68	25.26	60.08

表 3.3.2-20　Verify-on-Demand 方式のセキュリティ測定結果（リプレイ）

評価観点	目標値	測定結果			
		1,100	1,200	1,600	2,000
要検証メッセージ破棄率［%］	0%	0	0	0	0
要検証メッセージ処理時間の平均値［ms］	100 以下	14.59	14.77	14.57	20.80

表 3.3.2-21　Verify-on-Demand 方式のセキュリティ測定結果（高度な DoS:A）

評価観点	目標値	測定結果			
		1,100	1,200	1,600	2,000
要検証メッセージ破棄率［%］	0%	0	*51.17*	*85.75*	*92.76*
要検証メッセージ処理時間の平均値［ms］	100 以下	14.59	*205.52*	*104.47*	*78.95*

表 3.3.2-22　Verify-on-Demand 方式のセキュリティ測定結果（高度な DoS:B）

評価観点	目標値	測定結果			
		1,100	1,200	1,600	2,000
要検証メッセージ破棄率［%］	0%	0	*49.07*	*86.92*	*91.59*
要検証メッセージ処理時間の平均値［ms］	100 以下	14.59	*205.92*	*86.56*	*51.86*

表 3.3.2-23　Verify-on-Demand 方式のセキュリティ測定結果（高度な DoS:C）

評価観点	目標値	測定結果			
		1,100	1,200	1,600	2,000
要検証メッセージ破棄率［%］	0%	0	*56.98*	*84.62*	*91.55*
要検証メッセージ処理時間の平均値［ms］	100 以下	14.59	*205.39*	*86.56*	*54.26*

上記の結果から分かるとおりに、単純な DoS 攻撃およびリプレイ攻撃では目標を達成している。これは、Verify-on-Demand 方式の場合には、受信したメッセージの内容に応じて検証要否を判定し、処理するメッセージ数を削減するためである。このため、受信するメッセージが増加した場合でも検証要求キューにキューイングするメッセージ数を抑えることができる。

しかし、Verify-on-Demand 方式の場合には、高度な DoS 攻撃ではその攻撃パターンや負荷状況によらず目標が未達成である。これは、高度な DoS で送信される攻撃のメッセージが要検証メッセージと判定されるメッセージであり、攻撃のメッセージも含む要検証と判定されるメッセージの大量送信により検証要求キューが溢れるためである。今回のシナリオでは、Verify-on-Demand 方式では、現在の距離が 80m 未満であり、かつ自車に接近すると予想されるメッセージを検証要と判定するが、この条件に合致するメッセージが多い場合には、安全運転に影響を与える要検証メッセージも破棄されることになる。

Verify-on-Demand 方式の場合には、単純な DoS 攻撃およびリプレイ攻撃に耐性がある。攻撃を行うには、攻撃者が検証要否の判定基準を知り、攻撃対象車の現在状態をある程度把握した上で、要検証と判定されるメッセージを大量に送信する必要がある。攻撃の難易度が高いため、実際の V2X システムにおいても、セキュリティ維持に関して、ある程度の効果が期待できる。

⑥ 優先度付きメッセージ検証方式のセキュリティの評価

ここでは優先度付きメッセージ検証方式のセキュリティの評価結果およびその分析結果を説明する。表 3.3.2-24～3.3.2-29 には、評価の目標値および優先度付きメッセージ検証方式における各攻撃パターンでの各評価項目の測定結果を示す。なお、表中の測定結果欄下の数字は、それぞれ 1 秒間の受信メッセージ数である。1,100 については攻撃メッセージを含まない状態であり、1,200～2,000 が攻撃メッセージを含む状態である。測定結果である数値が*斜字体*の場合は、目標値に対し未達であったことを示す（セキュリティ評価の攻

撃パターンは表 3.3.2-3)。

本結果から、優先度付メッセージ検証方式の場合には、単純な DoS 攻撃およびリプレイ攻撃では概ね目標を達成していることが分かる。高度な DoS 攻撃では、要検証メッセージ処理時間の平均値は概ね目標を達成しているが、要検証メッセージ破棄率については目標に対し未達となることが分かった。ただし、高度な DoS の A については、破棄率は 15%以下と比較的低く抑えることができている。

表 3.3.2-24　優先度付き方式のセキュリティ測定結果（単純な DoS：攻撃元同一）

評価観点	目標値	測定結果			
		1,100	1,200	1,600	2,000
要検証メッセージ破棄率［%］	0%	0	0	0	0
要検証メッセージ処理時間の平均値［ms］	100 以下	24.46	17.29	28.29	43.28

表 3.3.2-25　優先度付き方式のセキュリティ測定結果（単純な DoS：攻撃元複数）

評価観点	目標値	測定結果			
		1,100	1,200	1,600	2,000
要検証メッセージ破棄率［%］	0%	0	0	0	*0.93*
要検証メッセージ処理時間の平均値［ms］	100 以下	24.46	18.66	21.63	65.85

表 3.3.2-26　優先度付き方式のセキュリティ測定結果（リプレイ）

評価観点	目標値	測定結果			
		1,100	1,200	1,600	2,000
要検証メッセージ破棄率［%］	0%	0	0	0	0
要検証メッセージ処理時間の平均値［ms］	100 以下	24.46	18.34	20.58	27.69

表 3.3.2-27　優先度付き方式のセキュリティ測定結果（高度な DoS:A）

評価観点	目標値	測定結果			
		1,100	1,200	1,600	2,000
要検証メッセージ破棄率［%］	0%	0	*3.97*	*10.77*	*12.15*
要検証メッセージ処理時間の平均値［ms］	100 以下	24.46	30.74	36.09	19.84

表 3.3.2-28　優先度付き方式のセキュリティ測定結果（高度な DoS:B）

評価観点	目標値	測定結果			
		1,100	1,200	1,600	2,000
要検証メッセージ破棄率［%］	0%	0	*31.07*	*84.35*	*98.60*
要検証メッセージ処理時間の平均値［ms］	100 以下	24.46	97.14	95.72	55.65

表 3.3.2-29　優先度付き方式のセキュリティ測定結果（高度な DoS:C）

評価観点	目標値	測定結果			
		1,100	1,200	1,600	2,000
要検証メッセージ破棄率［%］	0%	0	*33.49*	*98.60*	*99.70*
要検証メッセージ処理時間の平均値［ms］	100 以下	24.46	*103.96*	23.18	9.38

上記の結果から分かるとおりに、単純な DoS 攻撃およびリプレイ攻撃では目標を達成している。これは、優先度付きメッセージ検証方式の場合には、受信したメッセージの内容に応じて検証要否を判定し、処理するメッセージ数を削減するためである。このため、受信するメッセージが増加した場合でも検証要求キューにキューイングするメッセージ数を抑えることができる。これは前述したリアルタイム性の評価において、検証待ちや OS 処理に係る時間が 15 ms 程度と短いことからも明らかである。なお、攻撃元が複数である単純な DoS 攻撃において、1 秒間に 2,000 メッセージ送信する場合には、要検証メッセージの破棄率が 0.93%になっている。これは、優先度付きメッセージ検証方式で前述したとおり、Verify-on-Demand 方式に比べて処理するメッセージが多いため、処理が間に合わない場合があったためと考えられる。

また、要検証メッセージ処理時間の平均値は、高度な DoS 攻撃 C の 1 秒間に 1,200 メッセージを受信する場合を除き目標を達成している。これは優先度付きメッセージ検証方式が他車の現在位置の近さに基づいてメッセージを優先的に処理する仕組みがあるためである。これにより、検証要求キューにメッセージが一杯でも、優先度が高いと判定されたメッセージ（以後、高優先度メッセージと呼ぶ）が優先して処理される。要検証メッセージも高優先度メッセージも他車の現在位置の近さに関係するため、高優先度なメッセージを優先して短時間に処理することが、結果として他車距離が 80m 以内である要検証メッセージは短時間で処理されることとなる。しかし、破棄率については目標を達成できていない。これは、高度な DoS 攻撃で送信される攻撃のメッセージは要検証メッセージと判定されるメッセージであり、攻撃のメッセージも含む要検証と判定されるメッセージが大量に送信されることによって検証要求キューが溢れるためである。今回のシナリオでは、他車の現在位置が近いと判定する攻撃メッセージが多い場合には、安全運転に影響を与える高優先度メッセージも破棄されることになる。

なお、表 3.3.2-29 の高度な DoS 攻撃 C の 1 秒間に 1,200 メッセージを受信する場合には、破棄率や処理時間の平均のいずれも目標を達成していない。ただし、表 3.3.2-28 の 1 秒間に 1,200 メッセージを受信する場合もほぼ同様の結果であり、特殊な結果ではないことが

分かる。むしろ、高度な DoS 攻撃 C の 1 秒間に 1,600 以上のメッセージを受信する場合については、攻撃メッセージに高優先度メッセージが多く、他のメッセージよりも優先度が高いと判定されるメッセージのみが処理されることで、結果として平均処理時間が短くなっていることが分かる。

優先度付きメッセージ検証方式の場合には、単純な DoS 攻撃およびリプレイ攻撃に耐性がある。攻撃を行うには攻撃者が要検証と判定されるメッセージを大量に送信する必要があり、攻撃者が検証要否や優先度を設定するための判定基準を知った上で、さらに攻撃対象の車の現在状態なども把握する必要がある。このため、実際の V2X システムにおいても、セキュリティを維持するための効果がある程度期待できる。

⑦ 各方式でのセキュリティ評価の比較

ここでは簡略化方式なしの場合、Verify-on-Demand 方式および優先度付きメッセージ検証方式について、各攻撃パターンに対する耐性や特徴を考察した。

(ⅰ)単純な DoS

送信元が単一の場合と複数の場合のいずれも、簡略化方式なしの場合は目標値を達成できず、単純な DoS 攻撃による耐性がないことが分かった。また、Verify-on-Demand 方式および優先度付きメッセージ検証方式のいずれも単純な DoS 攻撃には耐性があることが分かった。

なお、送信元が単一で 1 秒間に 2,000 メッセージを受信する場合は、Verify-on-Demand 方式が 80.99ms なのに対して、優先度付きメッセージ検証方式の処理時間は 43.28ms と半分程度で処理できている。これは、優先度付きメッセージ検証方式のメッセージ処理フローのキュー制御（Proc.4）で同一車両メッセージを、破棄しているためと思われる。

(ⅱ)リプレイ攻撃

リプレイ攻撃についても簡略化方式なしは目標を達成できず耐性がないことが分かった。Verify-on-Demand 方式および優先度付きメッセージ検証方式のいずれも目標を達成しているためリプレイ攻撃に耐性があることが分かった。

簡略化方式なしの場合も IEEE 1609.2 のリプレイ攻撃対策（表 3.3.1-1 の Step6）を有効にして計測している。このことから、リプレイ攻撃対策は、検証要求キューにキューイングする前に行う必要があることが分かった。

(ⅲ)高度な DoS 攻撃

簡略化方式なしの場合と、Verify-on-Demand 方式は目標値を達成できず、高度な DoS 攻撃に対しては耐性がないことが分かった。優先度付きメッセージ検証方式についても破棄率は目標未達となるものの、処理時間は、高度な DoS 攻撃の C の 1 秒間に 1,200 メッセージを送信される場合を除き目標を達成できることが分かった。また、高度な DoS 攻撃の A に限れば、優先度付きメッセージ検証方式の破棄率は負荷が高くとも 15%以下に抑えることができる。

なお、より優先度が高くなるメッセージを送信した他車の現在位置に着目した場合について、各高度な DoS 攻撃での高優先度メッセージの破棄率および処理時間の平均に対して、Verify-on-Demand 方式と優先度付きメッセージ検証方式の比較した結果を、表 3.3.2-30～3.3.2-35 に示す。なお、表中の測定結果欄下の数字は、それぞれ 1 秒間の受信メッセージ数である。1,100 については攻撃メッセージを含まない状態であり、1,200～2,000 が攻撃メッセージを含む状態である。測定結果である数値が*斜字体*の場合は、目標値に対し未達であったことを示す。

表 3.3.2-30　高度な DoS 攻撃：A における高優先度メッセージの破棄率の比較

方式	他車の現在位置	測定結果			
		1,100	1,200	1,600	2,000
Verify-on-Demand 方式［%］	80m 未満	0	*51.17*	*85.75*	*92.76*
	60m 未満	0	*48.92*	*86.69*	*92.68*
	40m 未満	0	*50.74*	*87.50*	*91.18*
優先度付きメッセージ検証方式［%］	80m 未満	0	*3.97*	*10.75*	*12.15*
	60m 未満	0	0	0	0
	40m 未満	0	0	0	0

表 3.3.2-31　高度な DoS 攻撃：A における高優先度メッセージの処理時間の平均の比較

方式	他車の現在位置	測定結果			
		1,100	1,200	1,600	2,000
Verify-on-Demand 方式［ms］	80m 未満	14.59	*204.52*	*104.47*	78.95
	60m 未満	14.59	*205.11*	*109.47*	84.62
	40m 未満	14.59	*203.93*	*112.29*	93.18
優先度付きメッセージ検証方式［ms］	80m 未満	24.46	30.74	36.09	19.84
	60m 未満	24.46	24.39	33.06	17.17
	40m 未満	24.46	22.82	31.69	15.70

表 3.3.2-32　高度な DoS 攻撃：B における高優先度メッセージの破棄率の比較

方式	他車の現在位置	測定結果			
		1,100	1,200	1,600	2,000
Verify-on-Demand 方式［%］	80m 未満	0	*49.07*	*86.92*	*91.59*
	60m 未満	0	*49.46*	*86.69*	*92.83*
	40m 未満	0	*51.47*	*86.03*	*93.38*
優先度付きメッセージ検証方式［%］	80m 未満	0	*31.07*	*84.35*	*98.60*
	60m 未満	0	*23.30*	*76.98*	*98.21*
	40m 未満	0	*2.94*	*55.15*	*97.06*

表 3.3.2-33　高度な DoS 攻撃:B における高優先度メッセージの処理時間の平均の比較

方式	他車の現在位置	測定結果			
		1,100	1,200	1,600	2,000
Verify-on-Demand 方式［ms］	80m 未満	14.59	*205.92*	86.56	51.86
	60m 未満	14.59	*206.91*	92.94	61.15
	40m 未満	14.59	*206.54*	92.86	72.06
優先度付きメッセージ検証方式［ms］	80m 未満	24.46	24.46	97.14	95.72
	60m 未満	24.46	66.00	99.41	63.68
	40m 未満	24.46	55.36	*104.41*	72.72

表 3.3.2-34　高度な DoS 攻撃:C における高優先度メッセージの破棄率の比較

方式	他車の現在位置	測定結果			
		1,100	1,200	1,600	2,000
Verify-on-Demand 方式［%］	80m 未満	0	*56.98*	*84.62*	*91.55*
	60m 未満	0	*53.76*	*85.30*	*92.09*
	40m 未満	0	*56.62*	*83.82*	*94.81*
優先度付きメッセージ検証方式［%］	80m 未満	0	*33.49*	*98.60*	*99.77*
	60m 未満	0	*35.84*	*99.64*	*99.64*
	40m 未満	0	*30.15*	*100*	*100*

表 3.3.2-35　高度な DoS 攻撃:C における高優先度メッセージの処理時間の平均の比較

方式	他車の現在位置	測定結果			
		1,100	1,200	1,600	2,000
Verify-on-Demand 方式［ms］	80m 未満	14.59	*205.39*	88.56	54.26
	60m 未満	14.59	*208.19*	93.91	61.88
	40m 未満	14.59	*206.28*	95.24	91.39
優先度付きメッセージ検証方式［ms］	80m 未満	24.46	*120.39*	23.18	9.38
	60m 未満	24.46	*122.07*	17.97	9.38
	40m 未満	24.46	*121.12*	-	-

以上の結果より、高度な DoS 攻撃の A と B については、優先度付きメッセージ検証方式では、他車の現在位置が近くなればメッセージの破棄率が低くなることが分かる。これは優先度付きメッセージ検証方式では他車の現在位置が近いメッセージをより優先度が高いとして検証要求キューの前の方にキューイングするからである。一方で、Verify-on-Demand 方式においては、他車の現在位置による優先度処理を行わないため、破棄率は変わらない。

逆に高度な DoS 攻撃の C では、いずれの方式も破棄率は高いが、1 秒間のメッセージの受信数が多くなると、優先度付きメッセージ検証方式は破棄率が 100%となる。これは、より優先すべきと判断される攻撃メッセージが多数含まれることによる。

（ⅳ）結論

攻撃者が任意のメッセージを大量に送信し、通信負荷を上げることは、攻撃手段としては容易であると言える。つまり、実際の V2X システムでは、セキュリティの観点からも何らかの簡略化方式を持つ必要があると結論付けられる。

Verify-on-Demand 方式および優先度付きメッセージ検証方式は、単純な DoS、リプレイ攻撃には耐性があるといえる。これらは、目標性能の 2 倍となる 1 秒間に 2,000 メッセージを受信させても問題が発生しないことから、このシナリオにおいては、実際の運用上も問題は発生しないと考えられる。

また、高度な DoS 攻撃については、優先度付きメッセージ検証方式の方が、Verify-on-Demand 方式に比較して優位になるケースがあった。これは、優先度付きメッセージ検証方式が検証すべきと判定したメッセージにおいても優先度をつけて検証要求キューにキューイングするためである。逆に言えば、その判定ロジックを悪用さえできれば、すべてのメッセージを破棄される恐れもある。

総論として、優先度付きメッセージ検証方式の方が攻撃に対して耐性が強いとの結果となっている。ただし、判定条件や優先度の判定条件を知っている攻撃者が意図して攻撃した場合には、優先度付きメッセージ検証方式の方が、メッセージ処理に悪影響がでる。しかしながら、攻撃の可否は、優先度が高くなるための条件に依存し、条件が厳しい場合には攻撃するのは困難となる。例えば、今回の評価における高度な DoS 攻撃の A や B では、走行中の車両の位置等の状況／状態を正確・即時に把握する必要がある。一方で、高度な DoS 攻撃の C では、V2X 通信のメッセージを利用する走行シーンから、要検証や優先度が高いと判定されるメッセージは何かを考慮した攻撃であり、任意の車両を攻撃するには、高度な DoS 攻撃の A や B よりも攻撃が成立しやすい。

このため、攻撃であることを早期に検知する方法、攻撃であった場合に安全運転に問題が発生しないように最低限必要なメッセージのみを処理するような方法（縮退モード）、IEEE 1609.2 の処理で実施しているリプレイ対策やメッセージの整合性検証などを検証要求キューにキューイングする前に行う方法等による対策の検討が必要である。

3.3.3 調査結果の整理および分析

前項までの調査結果を基に、各簡略化方式を比較すると共に、その効果を考察した。

(1) 簡略化方式のリアルタイム性とセキュリティへの影響に関する比較調査

簡略化方式なし、Verify-on-Demand 方式、優先度付きメッセージ検証方式について、リアルタイム性とセキュリティへの影響を評価した結果を、表 3.3.3-1 に示す。

表 3.3.3-1　リアルタイム性とセキュリティへの影響の評価結果

	評価項目		簡略化方式 なし	Verify-on- Demand 方式	優先度付き ﾒｯｾｰｼﾞ検証方式
リアルタイム性	シナリオ全体を通しての処理スループット[messages/s] 目標：:1,000 messages/s 以上		× 98.5	○ 1,097.4	○ 1,098.9
	要検証メッセージの処理遅延時間[ms] 目標：100ms 以内	最大	× 434.7	○ 44.7	○ 73.5
		20-40m	× 318.2	○ 14.8	○ 29.4
		40-60m	× 309.8	○ 14.4	○ 23.2
		60-80m	× 313.8	○ 14.6	○ 20.3
	検証メッセージ数		717	429	751
セキュリティ	単純な DoS：同一攻撃		× 耐性なし	○ 耐性あり	○ 耐性あり
	単純な DoS：複数攻撃				
	リプレイ攻撃				
	高度な DoS 攻撃 A(閾値近傍)		× 耐性なし	× 耐性なし	△ 耐性あり
	高度な DoS 攻撃 B(車両近傍)		× 耐性なし	× 耐性なし	× 耐性なし
	高度な DoS 攻撃 C(交差点経路上)		× 耐性なし	× 耐性なし	× 耐性なし

① 簡略化方式のリアルタイム性に関する比較

本評価によって、リアルタイム性について次のことが分かった。

R1：本研究で使用した交差点右折のシナリオにおいて簡略化方式（Verify-on-Demand）方式、優先度付きメッセージ検証方式）は目標である、スループット：1,000[messages/s]以上と、要検証メッセージ最大処理遅延：100[ms]以内を達成した。

R2：優先度付きメッセージ検証方式は、より優先度の高いメッセージを、より速く、優先的に処理できる特長を持つ。

R3：優先度付きメッセージ検証方式は、目標を達成しつつ、実際のメッセージの検証数が Verify-on-Demand 方式より多い。

R1 は、要検証と判断しなかったメッセージの検証を省略することの効果である。

R2 は、優先度付きメッセージ検証方式がメッセージの優先度に従って処理順序を変更することにより得られる特長である。

R3 は、R2 によって得られる効果である。R3 を言い換えると、優先度付きメッセージ検証方式は、処理に余裕がある場合に、“要検証ではない”メッセージも検証できる。なぜなら、“要検証ではない”メッセージを検証キューに積んだ後に要検証メッセージを受信したとしても、優先度付きメッセージ検証方式は要検証メッセージを優先的に処理できるからである。一方で Verify-on-Demand 方式は、要検証メッセージ以外をキューに積まないよう破棄しているので、処理に余裕があっても”要検証でない”メッセージを検証できない。

また、優先度付きメッセージ検証方式がより多くのメッセージを検証できるということは、より快適に自動車を制御できるという効果が得られると考えられる。なぜならば、要検証と判定する閾値にわずかに達しないメッセージを検証できれば、より早く制御を開始できるからである。つまり優先度付きメッセージ検証方式は、快適な運転に寄与すると推察できる。

また、優先度付きメッセージ検証方式は、優先度の高いメッセージを受信する前に優先度の低いメッセージを大量に受信した場合、優先度の高い（緊急性の高い）メッセージを優先的に検証できるが、Verify-on-Demand 方式は、優先的に検証できない。本評価では、Verify-on-Demand の方が要検証メッセージの処理時間が短い結果となったが、検証メッセージ数を両方式で同数にした場合は、優先度付きメッセージ検証方式の方が、高優先度メッセージの処理時間が短くなると推測される。

その他の効果として、優先度付きメッセージ検証方式は、設計が容易になると考えられる。例えば、V2X 通信を用いたシナリオは、本評価で使用した交差点右折のほかにも様々なものが検討されている。また道路状況も様々である。Verify-on-Demand 方式では、その時々の状況で厳密な閾値を設計することが要求されるが、その様な様々な状況に対応できる厳密な閾値を設計することは困難と思われる。一方で優先度付きメッセージ検証方式は、算出した優先度に従って、優先度の高いメッセージから検証するだけであり、検証要否を判定する厳密な閾値を必要としない。

② 簡略化方式のセキュリティに関する比較

簡略化方式（Verify-on-Demand 方式、優先度付きメッセージ検証方式）のセキュリティへの影響について次のことが分かった。

S1：簡略化方式は、単純な DoS 攻撃やリプレイ攻撃に対して耐性がある。

S2：簡略化方式は、そのフィルタ条件やアルゴリズム等に着目した高度な DoS 攻撃に対して脆弱である。

S3：優先度付きメッセージ検証方式は、低優先度のメッセージによる高度な DoS 攻撃に対して、高優先度メッセージを保護できる（高度な DoS 攻撃：A に対して耐性があった）

S4：簡略化方式を用いたとしても、検証要求キューを溢れさせる DoS 攻撃が想定されるため対策が必要である。

S1 は、“要検証ではない”と判断されるメッセージ、つまり署名検証を行わないで済むメッセージによる DoS 攻撃に対して、簡略化方式は有効であることを示している。

S2、S3 は、以下のことを示している。簡略化方式は、そのフィルタ条件やアルゴリズム

等に着目した攻撃に対して本質的に脆弱であるが、攻撃を成立させるための条件を限定することは、攻撃の難易度を上げる可能性がある。例えば、今回の Verify-on-Demand 方式のフィルタ条件の場合、自車から 80m 以内に存在することを示す攻撃メッセージであれば検証負荷を大幅に上げられる。一方で、今回の優先度付きメッセージ検証方式であれば、自車から 40m~80m 以内に存在することを示す攻撃メッセージは、同じ範囲に存在することを示す要検証メッセージ（厳密には、優先度が同じか、それ以下のメッセージ）の処理負荷を上げられるが、0~40m 以内に存在することを示す要検証メッセージの検証負荷を劇的には上げられない。このように優先度付きメッセージ検証方式は、DoS 攻撃による影響範囲を限定できる。

また、高度な DoS 攻撃：B や C を行う場合、ある空間に収容できない程の車両が多数存在する等、交通状況等からみて明らかに疑わしいメッセージがある。メッセージ検証を行う前に、メッセージの疑わしさを確認する手法、若しくは、メッセージの確からしさを確認できる手法があれば、DoS 攻撃による影響を低減できる可能性がある。

これまで DoS 攻撃による影響の軽減について触れてきたが、V2X 通信で異常な状況を検出した場合には、V2X 通信を用いずセンサ等による自立型の運転に移行することも考えられる。異常な状況とは、例えば、実質的な通信帯域の限界が、欧州の研究事例で挙げられている 1,000messages/s 程度とすると、それを大幅に超えるような通信が行われていることである。その他、空間に収容できない程の車両が多数存在することや、そもそも署名検証に失敗するメッセージがひとつでも存在すること自体が異常な状況とも考えられる。この場合に、V2X 通信を用いずセンサ等による自立型の運転に移行することは、自動車を安全に制御する、若しくは、不当な制御を回避するという観点からは、対策の一つとなり得る。一方で、攻撃者が攻撃時の車両挙動を予測できる場合、そのことを利用した攻撃も想定されるため、対策導入時の影響について十分考慮する必要がある。

(2) まとめ

3.3.1 項から 3.3.3 項において、「署名検証の簡略化方式」について、V2X 通信への適用を前提とした仮想環境を整え、署名検証の簡略化の効果に関する評価結果を報告した。評価結果の要点は次の 4 点である。

- 交差点右折のシナリオにおいて簡略化方式を用いることで目標性能を達成した。
- テーマ③で提案する簡略化方式である優先度付きメッセージ検証方式は、優先度に応じてメッセージを検証でき、Verify-on-Demand 方式と比較して、より多くのメッセージを検証できる。
- 簡略化方式は、リプレイ攻撃や単純な DoS 攻撃の影響を軽減できるが、簡略化方式のアルゴリズムに着目した高度な DoS 攻撃に対しては脆弱である。
- DoS 攻撃への耐性を高めるためには、メッセージ検証の前に、簡易的にメッセージの確からしさを検証する仕組みが必要である。

平成 28 年度の研究・開発では、V2X 通信を適用したシナリオの一例として交差点右折を用いて、セキュリティ機能と通信 SW を含めて評価した。リアルタイム性の観点での今

後の課題として次の 3 つが挙げられる。

・今年度は V2X の一例として交差点右折を取り上げたが、現在欧米で検討されている、より現実に即したユースケースにおける簡略化方式の効果について評価し、簡略化方式が実用的であることを示す必要がある。
・今年度は仮想環境上に実装した簡略化方式を評価したが、車載器向け HW に簡略化方式を実装した際の影響について評価する必要がある。
・簡略化方式が欧米等､既存の運用方式に対して親和性があるかを確認する必要がある。また、セキュリティの観点での今後の課題として、DoS 攻撃への耐性を高める、簡易的にメッセージの確からしさを検証する仕組みの評価が必要である。

3.3.4 標準化動向調査

今回、提案した簡略化方式である「優先度付きメッセージ検証方式」は、その導入を仮定した場合、V2X 通信における署名検証方式として国際レベルでの利用者の利便性の確保が要件となる。よって日米欧いずれにおいても適用可能な標準化の検討が必要と考えられることから、標準化に当たって適切な提案先の調査を開始した。

(1) 標準化動向調査の目的

提案先候補のひとつである V2X に関連する国際標準化ワーキンググループ ISO TC204 WG16 に参加し、標準化提案の可能性を探るための情報収集を行う。

(2) 標準化動向調査の結論

① WG16 会合の結果

本会合で 4 つの新規プロジェクト開始を合意。新規プロジェクトの 1 つは、WG16 が規格化した CALM (Communication Access for Land Mobile)アーキテクチャの構成要素であり、セキュリティ確保に関係する BSME(Bounded Secured Managed Entity)について、技術要件等を記載する技術レポート(TR)作成。他の新規プロジェクトは、既存標準規格の技術的詳細に関わる改訂。また、7 つのドキュメントが、ISO の手続き上、投票もしくは発行という次ステップへ進む予定である。

② 標準化提案先としての評価

WG16 では ITS 用機器のアーキテクチャの標準規格を策定済みであり、例えばアーキテクチャの拡張要件として受信処理における優先順位付けを追加する提案の可能性も考えられる。現時点では提案先の明確な選定はできないが、WG16 を標準化提案先候補として、調査検討を継続して行うべきと考える。また、簡略化方式の標準化先の検討だけでなく、提案の実現可能性についても検討する必要がある。

参考文献（テーマ③）

[1] “平成２７年度戦略的イノベーション創造プログラム（自動走行システム）：Ｖ２Ｘ等車外情報の活用にかかるセキュリティ技術の研究・開発プロジェクト”, 一般財団法人 日本自動車研究所, 2016 年 3 月

[2] IEEE 1609.2: 2013. IEEE Standard for Wireless Access in Vehicular Environments — Security Services for Applications and Management Messages

[3] Krishnan, H. et al., ""Verify-on-Demand” - A Practical and Scalable Approach for Broadcast Authentication in Vehicle-to-Vehicle Communication”, SAE Int. J. Passeng. Cars – Mech. Syst. 4(1):536-546, 2011.

[4] André Weimerskirch, “V2V Communication Security: A Privacy Preserving Design for 300 Million Vehicles”, CHES 2014, Sep. 2014.
http://www.chesworkshop.org/ches2014/presentations/CHES_2014_Invited.pdf

[5] ”PRESERVE Deliverable 1.1 Security Requirements of Vehicle Security Architecture”, PRESERVE, June 2011.
https://www.preserve-project.eu/sites/preserve-project.eu/files/PRESERVE-D1.1-Security%20Requirements%20of%20Vehicle%20Security%20Architecture.pdf

[6] IEEE 1609.3: 2013. IEEE Standard for Wireless Access in Vehicular Environments — Networking Services

[7] “第 4 機 ASV 推進計画成果報告会（成果報告書）”, 国土交通省, 2011 年 6 月.
http://www.mlit.go.jp/jidosha/anzen/01asv/resourse/data/asv4pamphlet_seika.pdf

3.4　V2X セキュリティに関する海外の仕様や技術動向に関する情報共有（テーマ④）

海外の V2X 通信は、国内と異なる仕様となっているため、国内で議論・情報共有する場がない。また、V2X 通信の端末を製造している Tier-1、デバイスメーカー等が単独では十分な情報を収集できていない。そこで、V2X 通信に関する機器の輸出に貢献するために、海外の動向調査とその情報共有を行った。

3.4.1　海外の動向調査

V2X セキュリティに関して調査した海外の会議等について、表 3.4-1 海外動向調査の対象に記載する。

表 3.4-1　海外動向調査の対象

会議名	日程	開催場所
ITS World Congress	2016/10/10 - 10/14	Melborne, Australia
escar Europe (+ Workshop)	2016/11/16 - 11/17	Munich, Germany
Consumer Telematics Show	2017/1/4	Las Vegas, USA
Consumer Electronics Show	2017/1/5 - 1/8	Las Vegas, USA
Transportation Research Board	2017/1/8 - 1/12	Washington D.C., USA
Mobile World Congress	2017/2/27 - 3/2	Barcelona, Spain

(1) ITS World Congress

ITS World Congress では、ITS に関する様々なセッション・デモが行われ、自動運転及び自動車セキュリティ関連のセッションがあることから、これに参加して自動運転の実用化動向及びセキュリティ技術関連の動向調査を行った。結果の要旨を以下に記載する。

① 今回の ITS 世界会議では､Big Data という言葉と Open data という言葉がよく聞かれた。また、データを扱うときの注意事項として、Security と Privacy がセットで登場することが多かった。
② セキュリティの話では､攻撃事例を挙げたりするスピーカは多かったが､具体的な対応の話は少なかった。また、セキュリティ対策の話としては、IT 系で語られるものとほぼ同じ内容であった。KPIT 社からの報告では、セキュリティに関する研究の結果、仕様の問題が 41%、設計が 14%であり、仕様と設計をきちんとやっていれば 55%のセキュリティ不良は防ぐことができ､他に Maintenance の問題が 20%を占めるとのことであった。
③ 今回のスピーカの間でも、今日の環境では 100%セキュリティは不可能であり、効果的な対応としては侵入検知と被害を受けたシステムの復旧にフォーカスすべきであるという意見を述べる人もいた。

④ DSRC と LTE の比較で 1 つのセッションが設けられており、どちらの技術も課題を抱えていること、それぞれの特性から将来は DSRC と LTE/5G が共存（役割分担）していくとの意見がほとんどであった。但し、DSRC では既に実証実験が行われており実用化も近いのに対して、LTE を使った V2X は大規模実証実験が行われていないと状況が異なっている。

⑤ Regulation（規制）のセッションでは、Regulator（規制者）がどう考えてルール作りを行うのかの説明があり、自動運転に対応したルールを考える前提などを整理して説明されていた。Regulator の役割は、目的を達成するために、標準を定め、入口・出口を管理し、情報を出し、実行させることであるが、自動車技術の進歩によって Regulation も変えていく必要がある。例えば、Safety & Roadworthiness（安全と路上運転適正）は今と将来とで、また新車や使用中車両という条件ごとに変わること、利用者の ID 管理では、将来は運転免許がなくなるなど、いくつかの違いが出てくることが考えられる。

⑥ また、展示会場での情報として、V2X の仕様については、HW は Fix、SW はほぼ Fix だが、アメリカのセキュリティ仕様がまだ固まっていないとの情報を得た。セキュリティのどの部分かまでは聞き出すことが出来なかったが、アメリカ向けの HW は出来ているが、SW がまだという状況との説明であった。おそらく、インフラ側の仕様と思われる。

(2) escar Europe

escar Europe は自動車のセキュリティに関する会議であり、第一回は 2003 年に開催され、2016 年で 14 回目を数える。 BlackHat（DEF CON）とは異なり、車のハッキングに関する研究ではなく、CAN のセキュリティ、セキュリティテスト、侵入検知等の技術に関する発表が主なものである。

① CAN のセキュリティに関しては、既に様々なアプローチが発表されているが、基本的にはプロトコルやハードウェアへの変更が無い方式が採用されると考えられ、まだどのような技術が主流となるかは未知数であると考えられる。

② ファジングによる車のセキュリティテスティングについては、市販ツール（例：Defensics（旧名：Codenomicon））もあるが、今回の発表を見る限り、技術的に確立して実用的に利用されるまでは、まだ時間がかかるように予想される。

③ OTA（Over The Air）はセキュリティ・安全対策が非常に重要であるが、UPTANE（自動車向けソフトウェアアップデートのセキュリティシステム）や Automotive Linux との関連がある OTA plus など、オープンスタンダードに基づくアプローチが、どこまで業界に浸透するかは注目する必要があると考えられる。

④ 今回の発表では農業機械に関連する発表が２件あったが、自動車のセキュリティと同様な課題が農業機械にもあり、今後セキュリティ上の脅威が顕在化する可能性が高いと予想される。

(3) Consumer Telematics Show（CTS）

CTS は毎年、Consumer Electronics Show（CES）の前日に同じラスベガスで開催されており、2017 年は 1 月 4 日に開催された。CTS においても自動車セキュリティ関連の話題が取り上げられていたため、これを調査した。主な話題は以下の通りであった。

① SAE J3061 "Surface Vehicle Recommended Practice（Cybersecurity Guidebook for Cyber-Physical Vehicle Systems）"の概要報告

Green Hills から J3061 の概要報告があった。ポイントは「プロセスフレームワーク」と「Cybersecurity は設計段階にビルトインされなければならず、完成品にアッドオンするものではない」という点にある。考え方としては、ISO26262 Functional Safety に類似したもので、「全ての safety-critical System は、Cybersecurity-critical である」ということと「考えられる脅威を示すことは、引き起こされる障害を考えることよりも検討が難しい」ことから、設計者は「ハッカーになったつもり」で考える必要があると説明された。

最小権限の原則（principle of least privilege）、縦深防御（defense in depth）、インシデント対応（incident response）、現場監視（field monitoring）、OTA、セキュアブートなどの言葉が紹介されていた。

② 2016 年 10 月 NHTSA 発行のガイダンス"Cybersecurity Best Practices for Modern Vehicles"

ガイダンスのポイントは、「cybersecurity 対策を最重点項目として扱う」「車両 cybersecurity に関連した組織ランク全てに対してシームレスに直接のコンタクトをとれるようにする」「cybersecurity の訓練について従業員をサポートする」「階層化アプローチを用いる」「リスクベースで優先順位を付け、セーフティクリティカルな車両制御システム/個人情報の識別と保護に基づいて設計する」ことであると説明された。

また、インシデントへの対応のあり方として、「車両での潜在的な車両サイバーセキュリティ被害のタイムリーな検出と迅速な対応」「インシデントが発生したときの迅速なリカバリを実現する設計と対策」「Auto ISAC に参加するなどの効果的な情報共有を通じて、業界全体で学んだ教訓を促進するための方法を制度化する」ことが求められる。

③ V2X「DSRC vs 5G」

V2X に関しては様々な意見があり、DSRC 向けとして 1999 年に 5.9GHz スペクトラムに帯域 75MHz を割り当てたが、その後 16 年以上も使われずに放置されている。それがこれから本当に展開できるのかという疑問が残る。一方 5G については現在仕様策定中で実用化には、あと 5 年かかるという認識を持つ人もいる状態である。

また、V2X 通信の大きな課題として、ハッキングされると「インフラを通して、ハッカーが加減速制御やステアリング制御を行い、車をクラッシュさせる可能性」があることが挙げられる。これについては、政府としても「各 OEM にサイバーセキュリティ計画を立てさせ、予防的措置と自動車のライフサイクル全般にわたってのセキュリティアップデー

トをどのように行うか」を検討させておく必要があるのではないか、との意見が出されていた。

さらに、ユーザへの説明として、送信される情報については、DSRC 搭載車の購入者に対して、どのような個人情報が収集されその情報をどのように使用するのかを知らせる義務を課し、さらにその内容に対して、データ違反が発生した場合、情報を収集している製造業者は顧客に通知するなどの制度化も必要になるとの議論があった。

(4) Consumer Electronics Show 22017（CES 2017）

CTS に引き続き、1 月 5 日から CES が開催された。元々は家電製品を中心とした展示会であったが、近年は自動車そのものが展示されていたり、自動車関連の企業（Tier-1 など）の展示ブースがあったりと、近年では自動運転などの自動車技術の発表の場にもなってきている。また、CES は展示だけではなく、コンファレンス等も行われている。

① CAR HACKING VILLAGE

CAR HACKING VILLAGE は、Def Con 23 からスタートし、Def Con 24 でも実施され、小規模ながら DERBYCON、GrrCON、CES、Cypher Con でも実施したとのこと。The Badge と呼ばれるハッキングキット($50)を使って様々な車両ハッキングに挑戦するコンテストであり、DRIVER Information Zone/ANTILOCK Brake It Zone（車両分解） / 12-VOLT buck Hacking Zone / FCA Zone / TURBO Learning Zone（トレーニングコース）に分かれて開かれている。

特筆すべきことは、このコンテストをサポートしているのが、FCA、Delphi、spirent や bugcrowd などの車両メーカ、セキュリティソリューション会社という点であり、これらの会社はセキュリティ技術者、あるいはホワイトハットハッカーを探す一つの手段として場を提供していると推測される。

② GAO (U.S. Government Accountability Office: 米国会計検査院)

2013 年 11 月には V2V の有効性検証や、2014 年 1 月には Connected Car に関する Privacy 保護の調査などを行っている。GAO は米国立法府に属する調査機関であるが、業務の範囲が広く、「政府の政策・プログラムに対する評価」や「違法性や不適切に関して疑惑のある活動への捜査」なども業務範囲に入っているため、V2V や Connected Car に対しても調査を行っている。

(5) Transportation Research Board（TRB）Annual Meeting

2017 年の Annual Meeting は、1 月 8 日～12 日にワシントン DC で開催された。TRB には Cybersecurity Subcommittee があり、Subcommittee Meeting を含め、Security がテーマとなっているセッションは 10 以上あった。また、今回から Hot Topics と名付けられたセッションがいくつか設けられ、その一つに、セキュリティにも関連する Resilience が選ばれていた。

① Resiliency (復旧力)

TRB のセッションでいたるところで聞かれた言葉が Resiliency である。セキュリティ対応力のレベルが上がり、今は Resiliency のレベルの議論が進んできている。

Resiliency は、セキュリティ対策の次のようなステップの 4 つ目として扱うことが出来る。①Tools Based「何かのツールを使って発見するツールベース」→②Integrated Framework「設計する段階からプロセスとしてセキュリティを織り込むフレームベース」→③Dynamic Defense「リアルタイムに攻撃から守るダイナミックディフェンス」→④Resilient「何か起きてもすぐに復旧する復旧力」。

② サイバーフィジカルシステムへの対応

街レベルや地域レベルの「セキュリティ」は、現在フィジカルだけでなく、サイバーフィジカルシステムへの転換期である。まだフィジカル部分が先行していて、サイバー部分がシステムの中に入り込んでいない状態にある。

しかし、フィジカルセキュリティを担当してきたメンバーも、サイバーフィジカルシステムの恩恵は十分感じており、必要性も重要性も分かっていることから、これから活動が加速するものと思われる。（例：交通管制へカメラを導入して、人が映像を見て判断に使うことをしているが、その映像がハッキングされ嘘の情報を流されたらどうなるかというところまでは対応ができていない。しかしそれは大変重要な問題であるという認識を持っている、など）

③ VMS（Variable-Message Sign）のハッキング

交通メッセージボード VMS がハッキングされたことによる事故の原因分析からその対策は以下の通りだと報告されていた。元々ハッキングされることを前提としては作られていない車両の CAN バスでも、同様にハッキングが起こり得ることが想定される。ここで述べられていた対策は以下の通りである。

・最新の IT システムにアップデートする。旧来のものに追加対策しても不十分。
・賢い人材を作る=オペレータ教育。
・サイバー環境の衛生と健康=パッチ当てを怠らずにやるなどの教育（サイバーセキュリティ強化のためのステップ）。
・古い技術と IT 技術のギャップを架け橋する。
・情報セキュリティとモニタリングをする。
・システムに冗長性を持たせる。
・侵入検知システムを導入する。
・ホワイトハットハッカーの知見をもらう。

(6) Mobile World Congress

Mobile World Congress2017(MWC)にて V2X 通信関連の動向調査を行った。MWC は移動体通信の展示会にも関わらず多くの欧州自動車 OEM が展示ブースを出展しており、自動運転に向けての車載分野における通信の重要性を認識することになった。

車両に対する通信の適用は①IoT の端末として車両情報が使用される、②無線をセンサーとして制御システムへのデータ取得する、の 2 種類が考えられる。5G では①、②を融合する提案を 5GAA（5G Automotive Association）が行っており、今後 5G の TCU や V2X への適用を継続的に調査することが重要である。

5G の V2X への適応は物理層の刷新、802.11p から 5G 無線層への切り替えを、5GAA を中心に進めており、ドイツやフランスでは今年度中にフィールドテストも開始される。今後過渡的には 802.11p と 5G の Dual EUC 検討する必要が有る。現在 802.11p のチップベンターは3社（Qualcomm（旧 NXP）、Autotalks、Renesas）、5Gのチップベンダーは2社（Qualcomm、Intel）と非常に少なく、サービス開始前から部品サプライヤが限定されている。そのことが V2X の技術開発や市場形成に大きな影響を与えることが予想される。

Qualcomm は今回の展示の中でもっとも V2X に関する技術情報を発信しており、自動車向けのネットワークだけでなく、既存の移動体通信網との親和性も提案していた。デバイス間の直接通信とネットワークとの通信を前提しているが、物理層の入れ替えを目的としており、現行 V2X のプロトコルやセキュリティは継続使用での提案である。

また、VW と Sierra Wireless の共同発表があり、VW としては通信技術としては LTE を適用し、Active Safety には 802.11p を使用するが、今後 5G に統合していくことが説明された。VW は車載通信技術の初期段階から開発にかかわると言及しており、そのパートナーとして Sierra Wireless を上げている。

3.4.2 情報共有の仕組み運営

海外向け V2X 通信に用いられるセキュリティ方式は国内と異なっているため、そのセキュリティ仕様などを調査しておくことは、海外向け V2X 通信機器の開発などを行う国内サプライヤにとって重要である。そこで本事業では、これらの情報収集と情報共有を行う仕組みとして、3.5.3 で述べるワーキンググループ B を設置した。ワーキンググループ B において、海外動向として調査すべき国際会議等の検討を行うとともに、調査の分担についても決定した。また、調査した結果については、ワーキンググループ B で共有した。

3.5　研究開発全体企画・管理

テーマ①（自動運転の共通モデルの構築と、それに基づく脅威分析、セキュリティ要件及び対策の検討)、テーマ②（車両への攻撃に対する対策の評価手法・認証の調査・研究)、テーマ③（V2X 通信における署名検証の簡略化の研究)、およびテーマ④（V2X セキュリティに関する海外の仕様や技術動向に関する情報共有)の各テーマの間の連携を図りつつ、全体工程表を策定するとともに、研究管理を実施した。

実施にあたっては、開発における課題を整理するとともに、国内外での技術開発動向や標準化動向も参考に、実施内容や開発費等についての開発全体企画を策定した。

なお、開発計画や実施内容について審議するとともに、研究開発を効率化するための助言をいただくため、外部有識者を含む開発検討会を設置した。

3.5.1　全体工程表の策定

V2X 等車外情報の活用にかかるセキュリティ技術の研究・開発について、本事業期間における全体工程表を策定し、技術課題を整理する。

3.5.2　開発検討会の運営

テーマ①～テーマ④に共通する事項や、各テーマにおける重要な課題などを検討する開発検討会は、2016 年 8 月 2 日、10 月 31 日、2017 年 1 月 30 日の 3 回開催した。開発検討会では、本事業の趣旨、実施内容について理解いただき、事業を進めていく上での課題について議論を行った。

開発検討会では、大規模実証実験を視野に入れた外からの攻撃についても議論された。本事業では車両内部のセキュリティを対象にしているが、「実機を用いた評価(テーマ②d)」において、実車の車両外部からの攻撃に関する評価項目の検討結果が報告された。これは、海外で公開された実車に対する攻撃事例を参考に、攻撃手法を検討したものである。

(1) 第 1 回開発検討会の開催：2016 年 8 月 2 日

・日本自動車研究所（以下、JARI）から今年度実施計画の概要、各研究室から実施内容の概要について説明を実施し、今年度の取組内容について委員の皆様にご理解、ご承認をいただいた。
・事業実施に利用するツール等が業界関係者で利用できるようなものになるように、関係者、特に JASPAR・日本自動車工業会（以下、JAMA）の意見を反映していくことが重要である。また、JASPAR・JAMA で取組んでいることとの整合を図り、今後の事業計画に反映させることも検討する。

(2) 第 2 回開発検討会の開催：2016 年 10 月 31 日

・JARI から今年度事業の全体進捗状況・課題等を報告した。
・共通モデル開発支援ツール、テストベッド、認証研究他、情報共有と連携が必要な項目がたくさんあるので、JASPAR・JAMA と連携を強化して進める。
・各研の研究ターゲットについて、全体のどの箇所をなぜ実施しており、前提条件は何か等についての分かりやすい説明が不足しているので改善をする。

(3) 第 3 回開発検討会の開催：2017 年 1 月 30 日

・JARI から今年度事業の成果の見通し、および、進捗状況・課題等を報告した。
・脅威分析ツールについて、出口として共通利用されることが望ましいので、今後、JASPAR・JAMA とも協議しながら進めることが重要である。
・コンポーネント評価技術は現状のセキュリティ技術を前提にしているが、15 年間以上使用される車において、IT 業界の攻撃者が攻撃対象を自動車へ拡大している現状を踏まえ、厳しく捉える必要がある。評価基準に含めていないが、OTA を考慮すべきである。
・コンポーネント評価技術は、設計者を対象にしたものである。設計者が、アプリケーションノートを参照しつつセキュリティレベルを決定することになる。
・第三者認証も重要なテーマなので、今後もしっかりと実施していく。

3.5.3 その他の会議

JARI、各研究室および関係企業によるワーキンググループとして、評価技術関連を議論するワーキンググループ A と、V2X セキュリティに関する情報共有のためのワーキンググループ B を設置した。ワーキンググループ A は 6 回開催し、研究室間の連携、評価環境の構築、事業参加者での攻撃事例情報の共有などの議論を行った。ワーキンググループ B は 2 回開催し、海外動向調査の対象選定や調査担当について議論し、また、調査した結果の情報共有を行った。

第三者認証に関する調査については、認証研究会を 2 回開催し、認証に関する課題と今後の調査に関する方向性を得た。

また、有識者などが参加する別途設定する自動運転システム研究推進委員会の第 1 回（2016 年 9 月 29 日）、第 4 回（2017 年 2 月 27 日）において研究開発報告を行い、開発計画や実施内容等に関してアドバイスやコメントを頂き、実施内容に反映した。

これらの他に、民間で構成する「情報セキュリティ研究開発シナリオ検討 SWG」とも、随時、情報共有・交換を行い、実施内容に反映した。

第 4 章　まとめ

今後ますます重要となっていくことが予想される自動車セキュリティに関して、4 年間の計画で以下のテーマの研究･開発に取り組んでおり、平成 28 年度はその 2 年目にあたる。

設定したテーマは大きく 4 つに分かれており、①自動運転の共通システムアーキテクチャの構築とそれに基づく脅威分析の実施、②コンポーネントレベル～車両レベルにおけるセキュリティ評価技術・基準の検討、③V2X 通信における署名検証簡略化の技術検討、④V2X 関連セキュリティ技術の海外動向調査である。平成 27 年度には主に調査を行っており、平成 28 年度は調査の結果を活かし、実際の評価等に取り組んだ。

①では、システムアーキテクチャ、ユースケース、脅威分析手法、リスクアセスメントなどの脅威分析の実施にあたって効率的に行うための脅威分析共通プラットフォームの構築を進めていくためのツールや、ツールの上で扱うデータベースの仕様策定を行った。また、仕様検討にあたっては、将来的に脅威分析共通プラットフォームが広く活用されるように、業界団体等との意見交換を開始した。

②では、コンポーネントレベル～車両レベルにおけるセキュリティ評価技術の検討を、本事業の中で設定した階層 2～4 のそれぞれで、実際の攻撃手法の再現や検討を行った。階層 4 のコンポーネントレベルの評価では、平成 27 年度に実施したリプログラミング手順に対する評価において、実装するセキュリティ技術のレベルを上げた評価対象に攻撃を行い、評価基準の検討にフィードバックした。階層 3 の車内ネットワークのレベルでは、ECU 間での鍵配布の評価のための評価環境を構築し、実機を用いた攻撃評価を実施したほか、車内ネットワークを流れる通信のプロトコルに着目し、これをシミュレーションによって再現し、シミュレータ活用の有効性について示した。階層 2 の車内システムレベルでは、評価環境としての車両模擬システムについての検討を行い、セントラルゲートウェイを組み込んだ評価環境のベースを構築した。また、車内ネットワークに不正な信号が侵入したことを検出する技術としてふるまい検知があるが、ロボカーを用いてふるまい検知技術の評価方法について検討を行った。

③では、平成 27 年度に提案した優先度付きメッセージ検証方式について、シミュレーションによる評価を行い、目標とする性能を達成していることを確認した。

④では、V2X 通信に関するセキュリティ技術の海外動向調査を実施した。米国では、V2V 車載器搭載義務化の規制案がパブリックコメントの段階にあるなど、V2X 車載器の搭載は広がっていくものと見られる。一方で、DSRC と LTE/5G の併用や、5G を V2X 通信にも活用するといった議論もあり、今後とも、こういった技術動向も見ていく必要があると考えられる。

Connected Car と呼ばれ、既に広がり始めている自動車における車外との通信は、自動走行に欠かせないダイナミックマップの情報入手や、車載 ECU の OTA に活用されるものと考えられる。一方で、国際会議等では車外との通信経路を利用したハッキングの研究結果

が発表されており、こうしたハッキングを防御するためにはセキュリティ対策を施すことが重要となっている。セキュリティ対策を考える上では、どこに、どのようなレベルの対策を実施すべきかを、コスト面も考慮した上で検討していく必要がある。

本事業により得られた結果は、今後、セキュリティ対策を適用し、その効果を検証して評価するために重要となる評価手法・評価基準を考えていく上で、非常に有効なものになると考えている。

経済産業省委託

平成 28 年度
戦略的イノベーション創造プログラム（自動走行システム）：
V2X 等車外情報の活用にかかるセキュリティ技術の
研究・開発プロジェクト

報　告　書

平成 29 年 3 月

発　　行　　一般財団法人　日本自動車研究所
東京都港区芝大門 1－1－30
日本自動車会館　12 階
TEL　03（5733）7925