「マルチメディア，分散，協調とモバイル
(DICOMO2017)シンポジウム」 平成29年6月

特定の CAN メッセージを送信する ECU に対する バスオフ攻撃を利用したなりすまし攻撃

家平 和輝[1] 井上 博之[2,3] 石田 賢治[2]

概要： 車載ネットワークで多く用いられる CAN プロトコルの脆弱性を利用した攻撃や，車載器が乗っ取られたり不正な車載器がつながったりすることによる CAN バスへの物理的なアクセスが問題となっている．CAN バスや ECU に対する DoS 攻撃の一種として，送信エラーを誘発することによりターゲット ECU を CAN バスから離脱させるバスオフ攻撃が提案されている．また，CAN バスへの物理的なアクセスが可能な攻撃者によるスタッフエラー注入を用いたバスオフ攻撃も提案されている．本論文では，バスオフ攻撃を利用して，受信 ECU にも送信 ECU にも検知されないようななりすまし攻撃方式を提案する．提案に基づき FPGA を用いた攻撃装置を実装し，実験室での CAN バスと ECU からなる模擬環境および実車にて本攻撃を検証し，なりすましによる効果と実車の動作を評価した．

A Spoofing Attack Using Bus-Off Attacks to a Specific ECU on the CAN Bus

KAZUKI IEHIRA[1] HIROYUKI INOUE[2,3] KENJI ISHIDA[2]

1. はじめに

車載ネットワークである CAN(Controller Area Network)は共有バス型のネットワークトポロジをとり，メッセージの暗号化の仕組みも標準化されていないため，盗聴やなりすまし攻撃に対して脆弱である．受信 ECU(Electronic Control Unit)では，従来の単純ななりすまし攻撃により注入されたメッセージは正規の CAN メッセージと区別することが原理的に難しいが，周期や内容によって正規のメッセージと区別して異常を検出することは可能である[1]. 送信 ECU では，自身が送信したものでないなりすましメッセージを検出することは可能である[2]．また，正規のメッセージを上書きするなりすまし攻撃では，上書きされたことを送信 ECU で検出することが可能である．そのため，何らかの方法で正規のメッセージの送信を阻止し，送信 ECU に検知されないようになりすましメッセージを注入することができれば，より効果的なりすまし攻撃が可能となる．本研究では，正規の送信 ECU の送受信を阻止し，なりすましメッセージを CAN バスに注入することで，受信 ECU および送信 ECU が異常を検知できないなりすまし攻撃方式を提案し，また実車のトラフィックに適用し評価と考察を行う．

1 広島市立大学情報科学部
Faculty of Information Sciences, Hiroshima City University
2 広島市立大学大学院情報科学研究科
Graduate School of Information Sciences, Hiroshima City University
3 重要生活機器連携セキュリティ協議会 研究開発センター
Connected Consumer Device Security Council (CCDS)

2. 関連研究

2.1 CAN プロトコルの特徴

CAN では，CAN バスを構成する CANL および CANH と呼ばれる 2 本の信号線を用いた差動通信を行っている．信号線に電圧差がなく論理“1”を表す状態をリセッシブ，電圧差があり論理“0”に対応する状態をドミナントと呼ぶ．もし，リセッシブとドミナントが同タイミングで送信されると，ドミナントが優先される．また，CAN では同じ論理のビットが連続して送信されることにより同期が取れなくなることを防ぐためにビットスタッフィングルールを適用しており，同じ論理のビットが 5bit 連続するときは論理を反転したビットを 1bit 挿入することが決められている．CAN メッセージは送信元アドレスを持たず，宛先アドレスである送信先の CAN ID しか持っていない．さらに，CAN メッセージは共有バス上でブロードキャスト通信されるため，送信元を知る手段がない．そのため，攻撃者が正規の ECU に成りすまして不正なメッセージを送信することで，他の ECU に不正な情報を渡すことや制御することが可能である．

CAN では，エラー制御方法として，ECU がエラーを検知すると 6bit のエラーフラグと 8bit のリセッシブから構成されるエラーフレームを送信することで，他の ECU に知らせる．エラーの種類として，送信したビット論理と受信したビット論理が異なるときに発生するビットエラー，ビットスタッフィングルールを無視したことを検出したときに発生するスタッフエラーなどがあり，通常のデータを送信

するメッセージをデータフレームと呼ぶ. CAN コントローラではエラーを検出することで，送信エラーカウント値（TEC)，受信エラーカウント値（REC）の値を増加させる．特に，送信者がビットエラーまたはスタッフエラーを検出した場合は TEC が 8 増加する．このように，増加する値は送信者か受信者であるか，また検知したエラー内容に応じて定められている．CAN コントローラには図 1 に示す 3 つの状態があり， TEC， REC の値に応じて状態が遷移し，バスへのアクセスに制限がかかり，エラーフラグのレベルも変化する．表 1 にアクセスの制限と状態に応じて送信されるエラーフラグを示す．エラーアクティブ状態は通常状態と定められており，制限はかからない．エラーパッシブ状態はエラーを起こしやすい状態と定められており，他の ECU の通信を妨げないために送信待機時間が通常より長く設定されており，さらにエラー検出時もリセッシブのみを送信するため，他の ECU による送信を妨げることはできない．バスオフ状態はバス上の通信に参加できない状態と定められており，送受信すべてを禁止されている．

2.2 CAN におけるなりすまし攻撃

車載 LAN や ECU の情報セキュリティに関する注目が高まるにつれて，車載 LAN に対する攻撃の検証や研究の報告が多く発表されている．CAN の攻撃事例として，初期の報告としては，なりすましメッセージを注入することで，メーターの表示改ざんや，ブレーキ等の不正制御が可能であることが報告されている[3]．CAN バスや OBD-II 診断ポートに対して, CAN メッセージの注入を行うことでなりすまし攻撃を実施し, ECU や車載器が不正な動作を行うような報告がいくつかある[4][5]．CAN バスへ物理的にアクセスできる攻撃者による, 物理層への介入を伴う攻撃として，CANH と CANL を交差して接続するテクニックを用いてエラーにできないフレームを送信する方法が報告されている[6]．しかし，この方法では，正規の送信 ECU もメッセージを受信するため, 異常を検知される可能性がある. また，

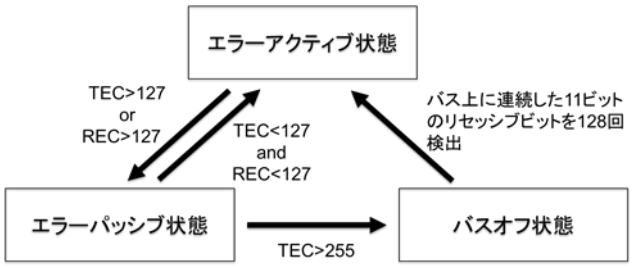

図 1　ECU の状態と遷移条件

表 1　ECU における状態に応じた制限

状態	CAN バスへのアクセス制限	エラーフラグ
エラーアクティブ	なし	6bit のドミナント
エラーパッシブ	8bit 期間の送信待機	6bit のリセッシブ
バスオフ	送受信の禁止	送信不可

送信 ECU と受信 ECU におけるサンプリングタイミングが異なる場合にタイミングよく CANH と CANL の電位差を変えることで，送信 ECU も受信 ECU も異常を検知できないなりすまし攻撃の方式が報告されている[7]．ただし，この方式ではタイミング制約が厳しいため，攻撃難易度が高く実現可能性が低いと考えられる．

2016 年 10 月には CAN に対する新たな DoS 攻撃として，送信エラーを誘発させターゲット ECU をバスオフ状態に遷移させ, CAN バスから離脱させるバスオフ攻撃が報告されている[8]．さらに，スタッフエラーを意図的に起こすことで，送信開始タイミングや周期といった時間的な制約のないバスオフ攻撃も報告されている[9]．従来のなりすまし攻撃では正規のメッセージも送信されるため，完全ななりすましはできない．バスオフ攻撃では送信 ECU をバスオフ状態に遷移させるため，送信 ECU は CAN バスに対しての送受信全ての動作を停止する．停止中に，攻撃者がなりすましメッセージを送信することで，正規の ECU になりすますことが可能となる．本研究では，正規の送信 ECU が送信するメッセージを故意にエラーにすることで，バスオフ状態に遷移させ，正規の送信 ECU の送受信を阻止し，なりすましメッセージを CAN バスに注入することで，受信 ECU および送信 ECU が異常を検知できないなりすまし攻撃方式を提案し，また実車に適用し評価と考察を行う．

3. ECU に対するバスオフ攻撃

3.1 ビットエラーを用いたバスオフ攻撃

バスオフ状態に遷移させるためのビットエラーを用いたバスオフ攻撃では，ターゲット ECU のビットエラーを誘発させることにより，ターゲット ECU の TEC を増加させる．攻撃手順を以下に示す．

(1) ターゲット ECU が送信を開始する直前に意味を持たないメッセージを注入し，ターゲット ECU の送信を阻止し，バッファに溜めさせる．

(2) (1)のメッセージを送信している間に攻撃者 ECU のバッファに以下の条件を満たすメッセージを溜める．
- ターゲット ECU と同一の CAN ID であること.
- ターゲット ECU の送信したリセッシブのビットに対して, 攻撃者がドミナントのビットを送信する. ただし, これ以前のビットは全て同一のものとする.

(3) (1)で送信したメッセージの送信が完了したのち, 規定の待機時間後，ターゲット ECU と攻撃者が同時に送信を開始する．

これより，図 2 に示すように，ターゲット ECU と攻撃者が同時に送信を開始する．その後，送信ビットが異なるタイミングでターゲット ECU がビットエラーを検出し, TEC が 8 増加される．そして，このビットエラーを繰り返し発生させることにより, TEC>255 にして, ターゲット ECU を

バスオフ状態に遷移させる．

この攻撃では攻撃者とターゲット ECU がエラーアクティブ状態であるときは再送も同時に行われるため，特に効果的に TEC を増加させることが出来る．このときの TEC の増加量は次の式(1)および(2)で得られる．

ターゲットECUのTEC増加量

$$= \left\lceil \frac{128 - \max(\text{攻撃者の TEC}, \text{ターゲット ECU の TEC})}{8} \right\rceil \times 8 \quad (1)$$

攻撃者ECUのTEC増加量

$$= \text{ターゲット ECU の TEC 増加量} - 8 \quad (2)$$

この攻撃では攻撃者の TEC も増加するため，攻撃者が特殊な CAN コントローラを使用しない限り，連続して攻撃を成功させることは難しい．

3.2 スタッフエラーを用いたバスオフ攻撃

スタッフエラーを用いたバスオフ攻撃は，図 3 に示すように，ターゲット ECU の送信したメッセージに対して，エラーフレームを送信することにより，ターゲット ECU の TEC を増加させ，バスオフ状態に遷移させる．攻撃手順を以下に示す．

(1) CAN バスのアイドル状態の検出
(2) データフレームの SOF の検出
(3) データフレームの CAN ID の読み取りと識別
(4) エラーフレームの送信

これより，ターゲット ECU は攻撃者から送信されたエラーフレームによりビットエラーまたはスタッフエラーを検出し，TEC が 8 増加する．そして，これを繰り返し行うことにより,TEC>255 にして，バスオフ状態に遷移させる．この方法では，バスオフ攻撃中も優先送信制御に従って，他の ECU もフレーム送信ができるため，CAN バスへの影響は少なく，異常状態を検出されにくい．

3.3 1 フレームでのバスオフ攻撃

CAN の仕様より，図 4 に示すように，エラー検出後にドミナントが 14bit 期間続いていると TEC が 8 増加され，さらに続けてドミナントが 8bit 期間続く毎に TEC が 8 増加される．以上より，最短 255bit（エラービット1bit＋14bit＋8bit×30）期間ドミナント状態が続いた場合，TEC>255 となり，ターゲット ECU はバスオフ状態に遷移する[9]．また, CAN ではビットスタッフィングが適用されるため, 6bit 以上連続して同じレベルのビットを送信することはない．これより，260bit 期間以上ドミナントを送信し続ければ，確実にターゲット ECU をバスオフ状態に遷移させることができる．この方法では，バスオフ攻撃中は常にビジー状態になるため，他の ECU のフレーム送信を許さない．そのため，異常状態を検知されやすくなる．

3.4 バスオフ攻撃方式のまとめ

バスオフ攻撃の各方式のメリットとデメリットを表 2 に

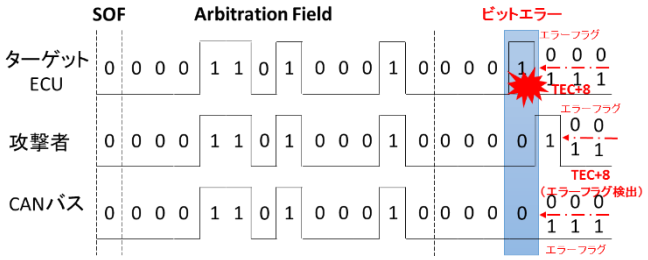

図 2 ビットエラーを用いたバスオフ攻撃

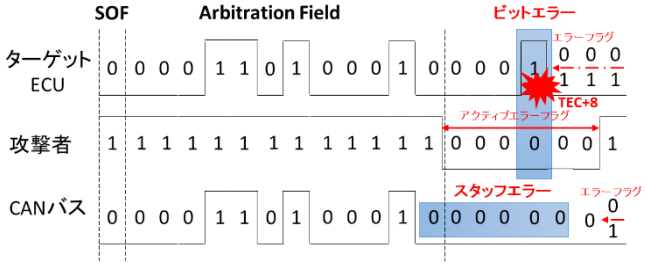

図 3 スタッフエラーを用いたバスオフ攻撃

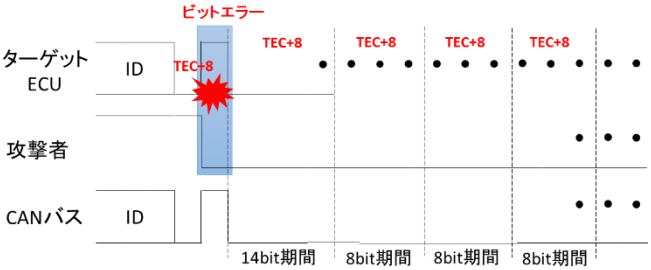

図 4 1 フレームでのバスオフ攻撃

示す．本研究では，攻撃を受信 ECU にも送信 ECU にも検知されないことを目的としている．そのため，確実にバスオフ攻撃を成功させることと，トラフィックへの影響が少ないことが望まれる．ビットエラーを用いたバスオフ攻撃では，連続して攻撃を成功させることが難しい点と意味をもたないメッセージの注入により異常を検知されやすくなることから，なりすまし攻撃との相性はよくない．スタッフエラーを用いたバスオフ攻撃では, 攻撃中も他の ECU の

表 2 バスオフ攻撃の種類と特徴

種類	メリット	デメリット	異常検知
ビットエラーを用いたバスオフ攻撃	CAN バスへの物理的にアクセスする必要がない	連続して攻撃を成功させることが難しい	検知されやすい
スタッフエラーを用いたバスオフ攻撃	攻撃中に他の ECU のフレーム送信を許す	CAN バスへの物理的にアクセスする必要がある	検知されにくい
1フレームでのバスオフ攻撃	短時間での攻撃が可能	攻撃中に他の ECU のフレーム送信を許さない, CAN バスへの物理的にアクセスする必要がある	検知される可能性がある

フレーム送信を許すことから，トラフィックへの干渉が少なく異常検知をされにくいため，なりすまし攻撃との相性がよい．1 フレームでのバスオフ攻撃では，一定期間 CAN バスを占有するため，異常を検知される可能性がある．ただし，確実にバスオフ攻撃を成功させることができる．そのため，なりすまし攻撃との相性は悪くはない．

4. 提案する攻撃方式

4.1 攻撃原理

図 5 に示すように，特定の CAN ID を持つデータフレームを繰り返しエラーにするバスオフ攻撃を用いてターゲット ECU をバスオフ状態に遷移させることで，正規メッセージを送信する ECU にも攻撃者がなりすましメッセージを送信していることを検知できなくなる．また，攻撃者は正規メッセージの周期に合わせて，なりすましメッセージを送信することで，受信 ECU はなりすましであることを検知できない．しかし，なりすましメッセージも特定の CAN ID を持つため，攻撃者 ECU が一般的な CAN コントローラであればバスオフ攻撃の対象になり得る．そのため，攻撃者 ECU の送信データを 6bit 期間サンプリングし，期間中にドミナントが検出されなければ，攻撃者の送信中ではないとし，攻撃を行う．これにより，正規メッセージを送信している ECU はバスオフ状態であるため，なりすましメッセージを受信できない．また，受信 ECU もなりすましメッセージのみを受信するため，受信頻度から異常を検知することができない．

4.2 提案方式の実装例

図 6 に提案方式の実装例を示す．ここでは，SOF 検出機能付きサンプリング回路により CAN バスに流れているフレームをサンプリングし，データフレームの CAN ID を検出する．検出した CAN ID を ID フィルタ回路に通し，ターゲット ECU の送信したメッセージの持つ CAN ID であれば，エラーフレーム送信回路によりエラーフレームを送信する．ただし，このとき攻撃者の送信データの最新 6bit 分のサンプリングデータに 1bit 分でもドミナントが含まれていることがサンプリング回路で検出されると，ターゲット ECU でなく攻撃者が送信したデータフレームであると判断しエラーフレームを送信しない．

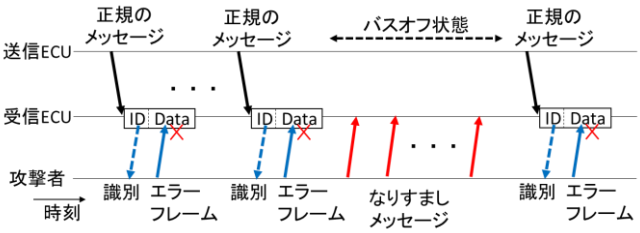

図 5 バスオフ攻撃を用いたなりすまし攻撃

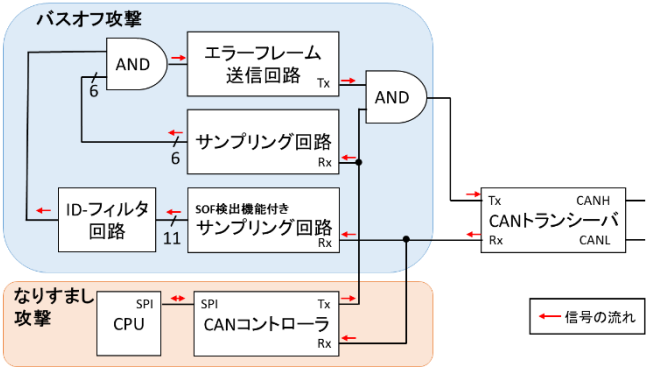

図 6 提案方式の実装例

5. 提案方式の評価実験と考察

5.1 実験環境

攻撃に必要な実験装置を構築するために以下の 4 つのデバイスを使用した．図 6 のバスオフ攻撃部は FPGA を用いてハードウェア実装をした．

1. CAN トランシーバ（MCP2551）
2. FPGA ボード（ZedBoard）
3. SPI 通信可能なマイコン（Arduino）
4. CAN コントローラ（MCP2515）

提案方式の基礎実験を行うために用いた実験環境を図 7 に示す．攻撃デバイスは ZedBoard と Arduino と CAN-BUS シールドから構成され，正規メッセージを送信するターゲット ECU は Arduino と CAN-BUS シールドから構成される．さらに，CAN バスの観測のため，CAN バス解析ツールである neoVI FIRE を用いる．CAN バス上に波形を観測するためのオシロスコープも接続した．CAN 通信は一般的な車載環境と同じ 500Kbps に設定した．今回は，周期 25[ms]である ID:1C4 に対してメッセージのなりすましを行う．また，正規メッセージのデータが“00 00 00 00 00 00 00 00”で，なりすましメッセージのデータを“17 70 00 00 00 00 00 00”としている．

5.2 実験結果

従来方式[5]を利用したなりすまし攻撃として，バスオフ攻撃を行わずに正規のメッセージと同一の周期でなりすましメッセージの注入を行った．図 8(a)に，正規のメッセージとなりすましのメッセージだけを抽出した結果を示す．図 8 では，前のメッセージからの経過時間（周期）と CAN ID および CAN メッセージのデータ部の内容を表示しており，青で囲んだメッセージはなりすましのメッセージを表している．図 8(a)より，正規メッセージとなりすましメッ

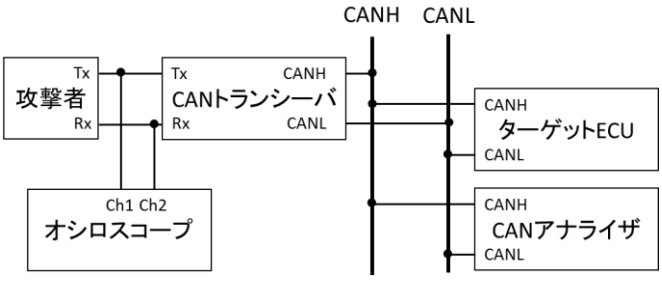

図 7 実験環境

Time (abs/rel)	ArbId	DataBytes
	1c4	
18.789 ms	1C4	00 00 00 00 00 00 00 00
4.318 ms	1C4	17 70 00 00 00 00 00 00
18.831 ms	1C4	00 00 00 00 00 00 00 00
4.274 ms	1C4	17 70 00 00 00 00 00 00
18.873 ms	1C4	00 00 00 00 00 00 00 00
4.230 ms	1C4	17 70 00 00 00 00 00 00
18.925 ms	1C4	00 00 00 00 00 00 00 00
4.178 ms	1C4	17 70 00 00 00 00 00 00
18.967 ms	1C4	00 00 00 00 00 00 00 00
4.136 ms	1C4	17 70 00 00 00 00 00 00

(a) 従来方式によるなりすまし攻撃

Time (abs/rel)	ArbId	DataBytes
	1c4	
23.103 ms	1C4	17 70 00 00 00 00 00 00
23.141 ms	1C4	00 00 00 00 00 00 00 00
382 µs	1C4	00 00 00 00 00 00 00 00
332 µs	1C4	00 00 00 00 00 00 00 00
274 µs	1C4	17 70 00 00 00 00 00 00
22.115 ms	1C4	00 00 00 00 00 00 00 00
274 µs	1C4	17 70 00 00 00 00 00 00
22.798 ms	1C4	17 70 00 00 00 00 00 00
23.120 ms	1C4	00 00 00 00 00 00 00 00
382 µs	1C4	00 00 00 00 00 00 00 00

(b) スタッフエラーを用いたバスオフ攻撃を利用したなりすまし攻撃

Time (abs/rel)	ArbId	DataBytes
	1c4	
23.371 ms	1C4	17 70 00 00 00 00 00 00
22.841 ms	1C4	17 70 00 00 00 00 00 00
23.105 ms	1C4	17 70 00 00 00 00 00 00
23.618 ms	1C4	17 70 00 00 00 00 00 00
22.589 ms	1C4	17 70 00 00 00 00 00 00
23.103 ms	1C4	17 70 00 00 00 00 00 00
23.111 ms	1C4	17 70 00 00 00 00 00 00
23.105 ms	1C4	17 70 00 00 00 00 00 00
23.101 ms	1C4	17 70 00 00 00 00 00 00
23.105 ms	1C4	17 70 00 00 00 00 00 00

(c) 1 フレームでのバスオフ攻撃を利用したなりすまし攻撃

図 8 正規のメッセージとなりすましのメッセージの受信間隔の比較

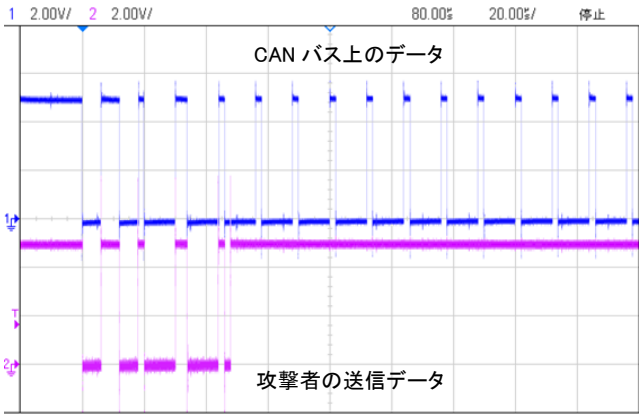

図 9 正規メッセージ送信成功時の波形

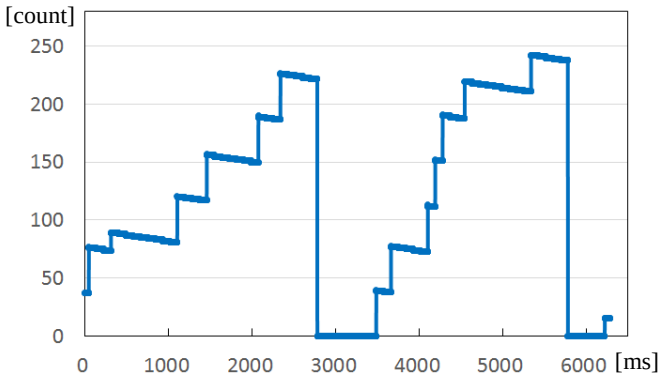

図 10 スタッフエラーを用いたバスオフ攻撃での TEC の変化

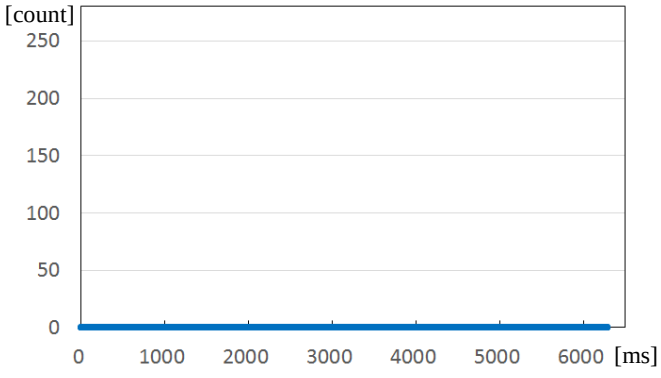

図 11 1 フレームでのバスオフ攻撃時での TEC の変化

表 3 なりすましメッセージの割合の比較

方式	割合
従来のなりすまし方式	50%
スタッフエラーを用いたバスオフ攻撃を利用した方式	65%
1 フレームでのバスオフ攻撃を利用した方式	100%

セージが交互に受信されていることが分かる．

次に，ターゲット ECU に対してスタッフエラーを用いたバスオフ攻撃を行い，同時に正規メッセージと同一の周期でなりすましメッセージの注入を行った．図 8(b)の実験結果より，正規メッセージとなりすましメッセージが入り乱れているのが分かる．特に，正規メッセージが受信された後の受信間隔を確認すると，赤線で示すように，周期が正常時の 50 分の 1 以下になっていることが分かる．また，図 9 の正規メッセージ送信成功時の波形より，攻撃者の送信したデータフレームがビットエラーになった後（図 2 を参照），パッシブエラーフラグを含むエラーフレームが送信されていることが分かる．また，図 10 に攻撃者の TEC 値の変化を示す．これより，攻撃者はパッシブエラー状態・バスオフ状態になっていることが分かる．

最後に，ターゲット ECU に対して 1 フレームでのバスオフ攻撃を行い，同時に正規メッセージと同一の周期でなりすましメッセージの注入を行う．図 8(c)の実験結果より，完全に正規メッセージが除去されたなりすまし攻撃が行われていることが分かる．ここで，図 11 に攻撃者の TEC 値の変化を示す．前述の 3 つの方式によるなりすまし攻撃における，なりすましメッセージの割合を表 3 に示す．これより，従来方式に比べ，提案方式の方がなりすませていることが分かる．特に，1 フレームでのバスオフ攻撃を利用した場合では完全になりすませていることが分かる．

5.3 実車実験

実験室で提案方式の有効性を確認できたので，次に実際の自動車の CAN バスと ECU を用いて評価を行う．自動車としては，国産ハイブリッド自動車を使用した．この実車実験における環境を表 4 に示す．実験室での結果が最も優

表 4 実車実験の環境

CAN 通信速度	500kbps
ターゲット ECU	CAN ID:1C4(エンジン回転数)
正規メッセージ(周期 23ms)	00 00 00 00 00 00 00 00(回転数:0)
なりすましメッセージ(周期 23ms)	17 70 00 00 00 00 00 00(回転数:6000)

れていた 1 フレームでのバスオフ攻撃を用いたなりすまし攻撃を自動車に対して実施した．なりすまし対象はエンジン回転数とし，なりすまし攻撃によりアイドリングストップ時にタコメータに不正な値を表示させる．なお，従来の CAN インジェクションによる単純ななりすまし攻撃[5]では，正規の 0 rpm を表すメッセージと競合することからタコメータの指針は 0 となりすましの値の中間でふらつき，完全になりすましの値にはならない．ふらつきを止めるには，実験を行った自動車の場合，約 50 倍のなりすましメッセージを注入する必要があった．

自動車に対して提案方式による攻撃を行ったところ，エンジン停止中にあるにも関わらず，タコメータの指針がなりすましの値に完全になり，ふらつくこともなかった．これは，正規メッセージが送信されていないため，正規のメッセージと同じ頻度で送信したなりすましメッセージで効果的になりすましが実現できたことを表している．また，実験に使った自動車で特にエラー表示が出ることもなかったため，提案方式による異常は検知されていないと考えられる．

5.4 考察

スタッフエラーを用いたバスオフ攻撃を利用したなりすまし攻撃では，正規メッセージも送信されていた．原因として，図 9 より，攻撃者とターゲット ECU が同時に送信開始しており，さらに攻撃者がビットエラーを検出し，パッシブエラーフラグ（6bit のリセッシブ）を送信していることが挙げられる．このとき，ID 検出時に攻撃者も送信を行っているため，バスオフ攻撃は行われない．また，攻撃者がパッシブエラーを送信しているため，少なくとも攻撃者はパッシブエラー状態であることが分かる．次に，同時に送信開始していることとパッシブエラー状態では送信待機時間が増加することから，ターゲット ECU もパッシブエラー状態であること確率が高いことが分かる. これより，条件“攻撃者，ターゲット ECU の両方がパッシブエラー状態”とビットエラーを用いたバスオフ攻撃の条件を満たすとき, 正規メッセージの送信を許すことが分かる. さらに，スタッフエラーを用いたバスオフ攻撃ではターゲット ECU が再送を行うため，攻撃者と送信タイミングが重なる確率が高くなっていたことより，条件を満たしやすくなっていたことが分かる．すなわち，攻撃者をターゲットとしたビットエラーを用いたバスオフ攻撃が行われていたと言える．また，条件を満たされるまでにターゲット ECU の TEC が増加している可能性が高いため，式(1)に従い，図 10 のように階段状に攻撃者の TEC が増加したと考えられる．条件を満たしたときにバッファに溜まっているデータフレームが連続して送信されたのは，正規メッセージの送信が成功した際に，本来の周期の 50 分の 1 以下の周期で正規メッセージが連続して送信されていたためと考えられる．

また，1 フレームでのバスオフ攻撃を利用したなりすまし攻撃では，ターゲット ECU に再送を行わせないため，送信タイミングが重なる確率が低く, 攻撃者の TEC が増加する確率が低いため，正規メッセージの送信が成功することがなかったと考えられる．

6. おわりに

特定の CAN メッセージを送信する ECU に対するバスオフ攻撃を用い，送信 ECU にも受信 ECU にも気づかれないなりすまし攻撃方式を提案し，評価実験を行った．実験結果より，完全になりすますことが可能であることが確認できた．今後は提案方式を用いた際の CAN バス上のトラフィックへの影響について評価を行うとともに，影響を最小限に抑えた攻撃方式の考案を行う予定である．

謝辞

本研究の一部は，広島市立大学特定研究費により行われた．ここに記して謝意を表す．

参考文献

[1] 伊達友裕，手柴瑞基，江崎貴也，井上博之: 車載 LAN のセキュリティゲートウェイにおける機械学習を用いた動的ルール生成，SCIS2016，pp.1-6，Jan. 2016.
[2] 畑正人，田邉正人，吉岡克成，大石和臣，松本勉: 不正送信阻止：CAN ではそれが可能である，CSS2011，pp.624-629，Oct. 2011.
[3] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, and S. Savage: Experimental Security Analysis of a Modern Automobile, Proc. 2010 IEEE Symposium on Security and Privacy, pp.447-462, May 2010.
[4] C.Miller, and C.Valasek: Remote Exploitation of an Unaltered Passenger Vehicle, Black Hat USA 2015, Aug. 2015.
[5] Takaya Ezaki, Tomohiro Date, and Hiroyuki Inoue: An Analysis Platform for the Information Security of In-vehicle Networks Connected with the External Networks, Proc. The 10th International Workshop on Security, pp.301-315, Aug. 2015.
[6] 菅原健，佐伯稔，三澤学: 強いリセッシブを用いた CAN の電気的データ改ざん，電子情報通信学会技術研究報告 ICSS，vol. 114，no.489，pp.67-72，Feb. 2015.
[7] 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データ改ざん，SCIS2015，pp.1-8，Jan. 2015.
[8] Cho, Kyong-Tak and Shin, Kang G.: Error Handling of In-vehicle Networks Makes Them Vulnerable, Proc. 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS2016), pp.1044-1055, Oct. 2016.
[9] 亀岡良太，久保田貴也，汐崎充，白畑正芳，倉地亮，藤野毅: ラズベリーパイからのスタッフエラー注入による CAN ECU へのバスオフ攻撃，SCIS2017，pp.1-8，Jan. 2017.