CANにおける不正送信阻止が可能となる条件

小林優希† 中山淑文† 松本 勉‡

横浜国立大学
†大学院環境情報学府 ‡大学院環境情報研究院／先端科学高等研究院
240-8501 神奈川県横浜市保土ヶ谷区常盤台 79-7
kobayashi-yuuki-mn@ynu.jp nakayama-yoshifumi-vg@ynu.jp tsutomu@ynu.ac.jp

あらまし CAN（Controller Area Network）は基幹的な車載ネットワークであるが，バス型のネットワーク構造であり認証機能も有していないため，不正に接続されたノードからなりすましメッセージを送信することがそのままでは容易となっている．この問題に対して，自身の使用するIDが他のノードからの送信に使用されたことを検知し，アクティブエラーフレームを用いて送信を阻止する方式が提案されている．本稿では，この方式で不正送信を阻止することができるための仮定や前提を整理し，この方式が有効に機能する条件について評価する．また，この評価を踏まえて，不正送信阻止の方式の無効化を狙う新たな攻撃方法についても考察する．

The Condition Where CAN Unauthorized Data Transmission Can Be Prevented

Yuuki Kobayashi† Yoshifumi Nakayama† Tsutomu Matsumoto‡

Yokohama National University
†Graduate School of Environment and Information Sciences
‡Faculty of Environment and Information Sciences, and Institute of Advanced Sciences
79-7 Tokiwadai, Hodogaya-ku, Yokohama, Kanagawa, 240-8501 JAPAN
kobayashi-yuuki-mn@ynu.jp nakayama-yoshifumi-vg@ynu.jp tsutomu@ynu.ac.jp

Abstract Any node illegally connected to an in-vehicle bus network realizing the ordinary CAN (Controller Area Network) protocol can readily make an unauthorized data transmission because CAN itself has no authentication functionality. However, if a node over the CAN bus detects an appearance of a CAN message headed by a CAN-ID which is assigned only to the node, then the node can find that someone is masquerading the node and can change the message into an error frame by some built-in mechanism of CAN. This paper extensively examines the assumption and condition by which such a method does actually prevent the unauthorized transmission. This examination implies novel attacks targeting the very method of preventing unauthorized data transmission.

1　はじめに

近年製造される自動車には，様々な機器を制御するために多くの ECU（Electronic Control Unit）が搭載されている．それらは，車載ネットワークにより接続され，互いに通信して制御を行っている．CAN（Controller Area Network）というバス型のネットワークは基幹的な車載ネットワークであり，CAN-FD や車載 Ethernet といった次世代の車載ネットワークの導入が進んだとしても，当面は利用されることが想定される．

CAN が有する脆弱性についてはすでに多くの指摘がなされている．例えば文献[1] では，自動車の診断用ポートから車載ネットワークに接続し，不正に ECU のファームウェアの書き換えが行えることが指摘されている．文献[2] では，車内のオーディオシステム，Bluetooth，携帯電話等を通じて車載ネットワークに接続し，不正な制御が行えることが指摘されている．

これらに対して，文献[3, 4] では暗号技術を用いた保護手法が，文献[5] では IDS（Intrusion Detection System：侵入検知システム）やファイアウォールを用いた保護手法が，また文献[6, 7] では CAN のエラーフレームを用いた不正送信阻止手法が提案されている．

従来の保護手法では，CAN の仕様に則ったノードが不正接続された場合や，すでにバスに接続されたノードのファームウェアが書き換えられた場合を想定している．しかし，ノードの不正接続に関して，文献[8, 9, 10] では CAN の仕様と関係なくバスの電位差を自由に変更できるノードを使うことで，より強力な攻撃が行えることが示されている．そこで本稿では，文献[8, 9, 10] の強力な攻撃者と攻撃手法に対して，現状の保護手法がどれだけ有効であるかについて整理する．

文献[10] で提案される手法については，文献[10] の中では，文献[6, 7] で示された不正送信阻止ノードを用いた実験例が記述されてないため，実際に不正送信阻止ノードを用いた実験を行い，この攻撃手法の有効性を確かめた．

さらに，本稿では，強力な攻撃者を仮定した場合に不正送信阻止手法に対して最も有効であると考えられる新たな攻撃手法を示すとともに，不正送信阻止ノードを用いて有効性を検証する．

本稿の構成は以下のとおりである．2 章で関連研究について紹介し，3 章で CAN について説明する．次に 4 章で文献[6, 7, 8, 9, 10] の手法の前提条件等について整理し，5 章で新たな攻撃手法について述べる．6 章でまとめを行い，今後の課題をあげる．

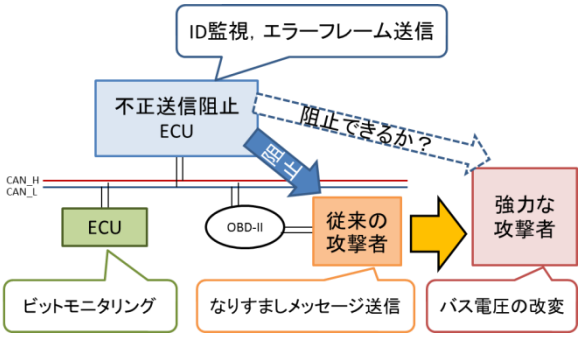

図 1：不正接続ノードによる攻撃

2　関連研究

車載ネットワークのセキュリティ強化手法としてこれまで多くの研究がなされてきた．不正送信阻止方式[6, 7] は，本来自身が送信するはずのメッセージが他のノードから送信されたことを検知し，エラーフレームを用いて不正メッセージの送信を中断させる．この時，不正メッセージの再送とエラーフレームの送信が繰り返されることで，不正メッセージの送信を行うノードはバスオフされる．

サンプリングポイントのずれを利用した電気的データ改ざん[8, 9] は，バスに接続されたノードのサンプリングポイントがそれぞれ違っていることに注目した攻撃手法である．CANでは送信ノードがバスの電位差をサンプルし，自身が送信したビットと異なる電位差を検知した場合にはビットエラーとして扱う仕組みがある．しかし，送信ノードと受信ノードのサンプリングポイントが1ビットの中でずれている場合には，受信ノードのサンプリングポイント付近のみに電気的な改ざんを行うことで，送信ノードに検知されることなく，受信ノードに不正メッセージを受信させることができるとしている．また，送信ノードと受信ノードのサンプリングポイントがずれていない場合でも，CANの機能である再同期を電位差操作により誘発することで，サンプリングポイントのずれを作り出すことができると指摘されている．

強いリセッシブを用いた電気的データ改ざん[10] では，文献[6, 7] の不正送信阻止手法を回避する攻撃手法が提案されている．不正送信

阻止手法では，エラーフレームを用いてなりすましメッセージの送信阻止を行うが，これはエラーフレームを構成する論理0の信号が論理1の信号と衝突した際に，論理0が優先するという性質に由来する．そのため，物理的な細工により，論理1の優先度を引き上げることでエラーフレームに上書きされない強い不正フレームを作ることができると指摘している．

これらの関連研究はそれぞれ想定している攻撃者や攻撃対象が異なっている．そこで本稿では先行研究で前提となっている仮定や条件を整理する．また，その条件下で不正送信阻止方式に対して有効と考えられる新たな攻撃手法について述べる．

3 CAN

CAN は，ISO11898，ISO11519 により標準化されたシリアル通信プロトコルである．2 線間の電位差によって信号を伝える 2 線式差動電圧方式を採用しており，ノイズ耐性を持つため，自動車，産業機器，医療機器などの制御情報の転送に用いられている．CAN はバス型のネットワークを想定しており，送信されたメッセージはバスに接続されているすべてのノードにブロードキャストされる．

3.1 CAN プロトコル

CAN では，NRZ（Non-Return-to-Zero）方式により，ビットエンコーディングを行っている．2 本の線の電位差が大きい状態をドミナント，小さい状態をリセッシブと呼び，ドミナントは“0”を，リセッシブは“1”をそれぞれ表す．複数ノードから同時に送信が始められる場合，ドミナントを送信するノードが優先される．

バスが空いている場合にはどのノードでも送信を行えるが，複数ノードが同時に送信を開始した場合には，メッセージに含まれる CAN-ID と呼ばれるビット列を符号なしの 2 進数とみなした時に値の小さいほうのメッセージが優先して送信される通信調停の仕組みを有する．

3.2 データフレーム

CAN では，フレームと呼ばれる単位でメッセージの送受信を行っている．フレームは使用用途によっていくつかの種類が使い分けられ，データの送信にはデータフレーム(図 2)が用いられる．データフレームは，主に ID フィールド，データフィールド，CRC シーケンスなどで構成される．

ID フィールドには，そのメッセージの CAN-ID が記述される．CAN-ID が 11bit のデータフレームのフォーマットを標準フォーマット，29bit のものを拡張フォーマットと呼ぶ．

送信したいデータは Data Field に 0~8byte の長さで格納される．

また，フレームの SOF から Data Field の最後のビットまでのビット列に対して規定の Cyclic Redundancy Check 符号によるパリティ検査ビットとして求められる 15bit の値が CRC Sequence の部分に格納される．

3.3 エラーフレーム

エラーフレーム（図 3）は各種エラーが発生した際にバスに接続するすべてのノードにエラーを通知するために用いられるフレームである．

CAN では各 ECU のシステムクロックの同期に関係して，同一レベルが 5bit 連続した場合に 1bit の反転レベルを送信するというビットスタッフィングルールを設けている．

エラーを検知したノードはあえてこのビットスタッフィングルールに違反する形で連続した 6bit の同一レベル（プライマリ）を送信する．次にビットスタッフィングルール違反を検知した他のノード

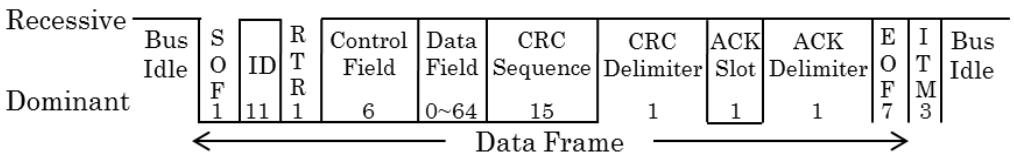

図 2:データフレーム（標準フォーマット）

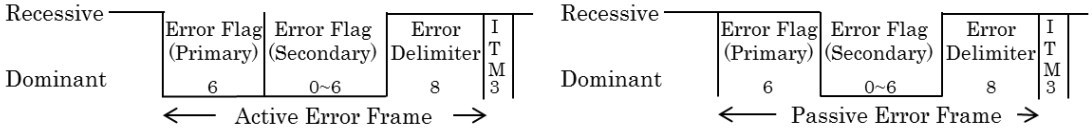

図 3:アクティブエラーフレームとパッシブエラーフレーム

から連続した 6bit の同一レベル(セカンダリ)が送信されることにより，全ノードにエラーが伝わったこととなる.

エラーフラグは送信するノードのエラー状態によって 0 を送信する場合と 1 を送信する場合がある. これによりエラーを起こしやすいノードが，他ノードの送信に対して積極的にエラーフレームによる中断ができないようにしている.

3.4 エラー検出機構

CAN プロトコルでは，以下のエラー検出機構が実装されており，これらのエラーが検出されたときにエラーフレームが送信される.

- ビットモニタリング
  送信ノードは，自身が送信した値とバスのサンプリングにより得られた値を比較し，異なる場合にはビットエラーとして検出する.
- スタッフチェック
  ビットスタッフィングルールが守られているかを監視し，同じ状態が6bit続いた場合にはスタッフエラーとして検出する.

これらに加えて，アクナレッジチェック，CRCチェック，フォームチェックでもエラーを検出することができるようになっている.

3.5 エラーカウンタ

CAN プロトコルでは各ノードに送信エラーカウンタ(TEC)と受信エラーカウンタ(REC)が定義されており，エラーが発生するたびにカウンタ値が上昇し，正常に通信が行えるとカウンタ値が減少する. カウンタ値によってノードの状態が遷移するようになっており，頻繁にエラーを起こすノードにより他のノードの通信が妨害されないようになっている. エラー状態は以下の 3 つがある.

- エラーアクティブ
  TEC≦127 かつ REC≦127 のノードの状態で，バスの通信に正常に参加する.
- エラーパッシブ
  127＜TEC≦255 または 127＜REC のノードの状態で，連続送信時に通信制限が行われる.
- バスオフ
  255＜TEC のノードの状態で，バスに対するすべての操作が禁止される.

表 1:攻撃者に関する仮定

	不正送信阻止[6, 7]	サンプリングポイントのずれ[8, 9]	強いリセッシブ[10]	エラーカウンタ値増加攻撃（本手法）
本来のCANコントローラの仕様に準拠	○	×	×	×
エラー状態の有無	有	無	無	無
バス電圧変更能力	CANのメッセージの形式でのみ可能	任意に変更可能	任意に変更可能	任意に変更可能

4 不正送信阻止の条件

ここでは[6, 7, 8, 9, 10]の手法について，それぞれが想定している仮定や前提条件について整理する. その上で不正送信阻止手法が実装された場合に，どの攻撃が最も脅威となりうるかを評価する.

4.1 攻撃者に関する仮定

それぞれの文献で想定している攻撃者に関して，異なる仮定がなされている. 不正送信阻止[6, 7]では，攻撃者が CAN の本来の仕様に準拠した CAN コントローラを用いているが，他の文献ではより強力な攻撃者を仮定している(表 1). ここで表 1 中のエラーカウンタ値増加攻撃は本稿で述べる新たな攻撃手法である.

4.2 被攻撃者に関する仮定

本稿における被攻撃者は[6, 7]で提案されている不正送信阻止方式が実装されたノード(以下不正送信阻止ノード)とする. ここで，不正送信阻止ノードは本来の CAN コントローラの機能に，自身の使用する ID のメッセージが他のノードから送信されたことを検知し，エラーフレームを送信する機能を追加しているものとする. また，不正送信阻止のためのエラーフレーム送信ではエラーカウンタ値が増加しない設定とする.

不正送信阻止が有効に機能するには，不正送信阻止ノードがエラーアクティブ状態であり，攻撃者が CAN の本来の仕様に準拠した CAN コントローラを用いている必要がある.

4.3　関連研究の比較

不正送信阻止ノードに対する各攻撃手法の効果について表 2 にまとめた.

この結果, サンプリングポイントのずれを利用した電気的データ改ざんでは, サンプリングポイント付近のみ改ざんする必要があり, 電圧改変精度が高くなければならず, バスのビットレートが高い場合には攻撃が難しくなるといえる. また, サンプリングポイントの位置やずれ幅を攻撃前に調べる必要があるといえる.

強いリセッシブを用いた電気的データ改ざんは不正送信阻止のアクティブエラーフレームを上書きすることができるが, 不正送信後にも不正送信阻止ノードがエラーアクティブ状態である場合には, アクティブエラーフレームの再送により, 他ノードにエラー発生が通知されてしまうと考えられる. また, 不正送信阻止のためのアクティブエラーフレームに対して改ざんを行っているため, 不正送信阻止ノードの実装の仕方によっては, 不正送信阻止ノードのエラーカウンタ値が増加することが考えられる. しかし, 不正送信阻止ノードは受信ノードであり, REC が増加するものと考えられるため, バスオフすることはないといえる. したがって, 攻撃時には不正送信したメッセージと正規のメッセージがバス上に混在するものと考えられる.

これらの関連研究に対して, 同じ攻撃者の能力を仮定した中で, 不正送信阻止手法に対する新たな脅威となりうる攻撃手法, エラーカウンタ値増加攻撃を示す.

5　エラーカウンタ値増加攻撃

4 章までで不正送信阻止手法や関連する攻撃手法についてまとめたが, ここでは, 不正送信阻止手法に対する新たな脅威として考えられるエラーカウンタ値増加攻撃を示す.

5.1　概要

エラーカウンタ値増加攻撃における攻撃者, 被攻撃者は 4.1 節, 4.2 節の仮定に基づいているものとする. また, 攻撃の開始時には不正送信阻止ノードはエラーアクティブ状態であるとする. 攻撃者は, 不正送信阻止ノード(被攻撃者)が送信するデータフレームおよびエラーフレームに改変を行うことで不正送信阻止ノードのエラー状態

表 2:関連研究との比較

	なりすましメッセージ送信	サンプリングポイントのずれ[8, 9]	強いリセッシブ[10]	エラーカウンタ値増加攻撃(本手法)
改変対象	-	被攻撃者が送信するデータフレーム	被攻撃者が送信するエラーフレーム	被攻撃者が送信するデータフレームおよびエラーフレーム
被攻撃者によるエラーの検知	可	不可	可	可
被攻撃者による他ノードへのCANでのエラー通知	可	不可	被攻撃者のエラー状態次第	被攻撃者のエラー状態次第
被攻撃者のエラー状態遷移	無 ・設定次第	無	有 ・RECの増加 ・エラーパッシブ状態まで	有 ・TECの増加 ・バスオフまで
電圧改変精度	-	高	低	低

をエラーパッシブとし, 不正送信阻止が行えないようにする.

5.2　攻撃の流れ

攻撃者の改変と被攻撃者の挙動について順を追って説明する.

I. 攻撃者は, 被攻撃者が送信するデータフレームのデータフィールドに 1bit の改変を行う.
II. 被攻撃者はビットエラーを検知し, TEC が増加するとともに, 直後のビットからアクティブエラーフレームを送信する.
III. 攻撃者は I の改変以降もデータフレームの形を保つように ACK スロットまで改変する. この時, ACK スロットの改変を終えるまでに被攻撃者の TEC が 128 以上になるように改変する必要がある.
IV. 攻撃者が改変を終えた後, 被攻撃者はエラーフレームの再送とメッセージの再送を行う. この時被攻撃者は TEC＞127 のため, 送信するエラーフレームはパッシブエラーフレームとなる.
V. I から IV を繰り返し行うことで被攻撃者の不正送信阻止機構を働かせることなく, 攻撃者はなりすまし攻撃を行うことができると考えられる.

I から V までを同じバス上にある第 3 者ノードから見ると, 不正送信阻止ノードが使っている ID のデータフレームが短い間隔で 2 個送信されたように見え, エラーは検出できない. これは不正送信阻止ノードが最後に送信したパッシブエラーフレームが, 1 個目のデータフレームの ACK デ

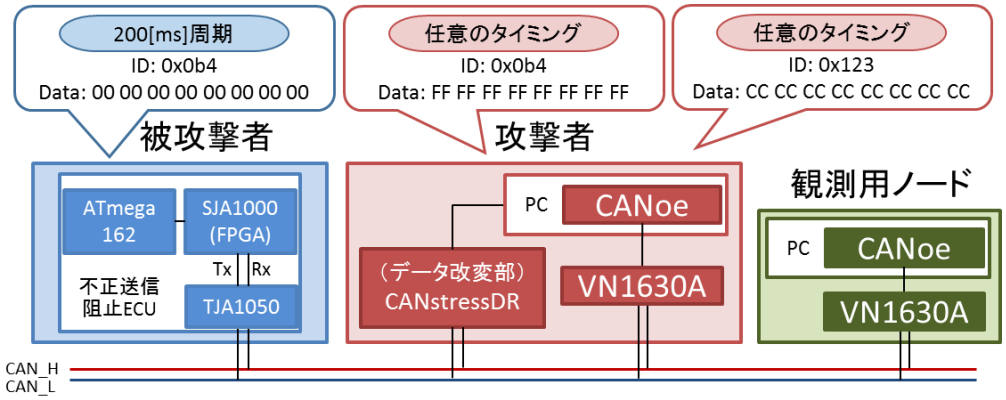

図 4:実験機器の構成

リミタからバスアイドルの 1 ビット目に重なってしまうためである.

5.3 検証実験

5.3.1 実験概要

エラーカウンタ値増加攻撃の実現可能性を検証するために実験を行った. 以下で概要を示す.

5.3.1.1 実験機器構成

実験機器の構成は図 4 に示した. 具体的には, PC 上で動作する Vector 社の ECU 開発ツールである CANoe[11], CAN インターフェースである VN1630A[12], バスの電圧を変化させる CANstressDR[13]によって攻撃者と観測用ノードを構成した. また, 表 3 で示した機器を用いて不正送信阻止ノードを実装した. [6, 7]の実装方法にならい, CAN コントローラは FPGA で構成し, ハードウェア記述言語を用いて, OpenCores[14]の CAN IP Core をベースに不正送信阻止機能のモジュールを追加する形で実装する. この CAN IP Core は市販されている Philips 社の SJA1000[15]という CAN コントローラと互換性があり, 実装後は不正送信阻止方式を導入した SJA1000 として扱うことができる. CAN コントローラ内に不正送信阻止機能のモジュールとして UTPS (Unauthorized Transmission Prevention System)モジュールを追加する. UTPS において, 受信した ID と自身の送信するメッセージの ID のリストとを比較し一致した場合かつ自身は送信を行っていない場合, 不正送信が行われたものとしてエラーフレームの送信を行う. この時, 不正送信阻止のためのエラーフレーム送信ではエラーカウンタ値の増加が 0 になるようにした.

表 3:実装環境・機器

機器	名称
ハードウェア記述言語	Verilog HDL
開発環境	Xilinx ISE 14.7
FPGA	Xilinx Spartan-6
マイコン	ATmega162
CANトランシーバ	TJA1050

文献[6, 7]ではマイコンから送信 ID のリストを設定できるようになっているが, 今回の実装では簡単のために CAN コントローラがあらかじめ送信 ID のリストを持つようにした. 実験時のビットレートは 500[kbps]とした. 攻撃者と被攻撃者から送信されるメッセージも図 4 に示してあり, それらのメッセージを観測用ノードで観測した.

5.3.1.2 不正送信阻止動作確認

まず, 被攻撃者の不正送信阻止機構の動作の確認を行った. 攻撃者から ID:0x123 のデータフレームを送信したところ, 正常に送信することができたが, ID:0x0b4 のデータフレームを送信したところ, アクティブエラーフレームにより送信が中断され, 攻撃者はバスオフとなった. この時, 被攻撃者の周期送信に影響はなく, 不正送信阻止が有効に機能していることが確認できた.

5.3.1.3 エラーカウンタ値動作確認

ここでは詳しく述べないが, 事前実験を行い, 5.2 節の I から III の手順によりメッセージを改変されたノードの TEC が十分に上昇し, エラーパッシブになることが確認できている.

5.3.1.4 実験手順

i. 被攻撃者の ID:0x0b4 のデータフレームに対して, 攻撃者のデータ改変部でを用いてデータの 1bit 目から ACK スロットまでを, データが「FF FF FF FF FF FF FF FF」となるように改変を行う.

ii. 攻撃者のデータ改変部による 1 フレーム分の改変が終わったところで攻撃者から ID:0x0b4 でメッセージの送信を行う.

iii. i の改変を複数のメッセージに対して行う.

5.3.1.5 実験結果・考察

i, ii を行ったところ, 攻撃者から送信した ID:0x0b4 のメッセージは不正送信阻止されることなく送信された. これは被攻撃者がエラーパッシブ状態に遷移した影響であると考えられ, その後, 被攻撃者の周期送信が行われるごとに TEC の減少が起こり, ノード A がエラーアクティブ状態に戻ったところで攻撃者から ID:0x0b4 で送信を行うと不正送信阻止された.

また, iii を行った場合, 6 メッセージ分改変を行ったところで被攻撃者からの周期送信が止まった. その後攻撃者から ID:0x0b4 で送信を行ったところ, 不正送信阻止されることなく送信された. これは被攻撃者がバスオフとなり通信に参加できなくなったためと考えられる.

5.3.2 研究[10]の追実験

同様の機器構成で強いリセッシブを用いた電気的データ改ざん[10]の追実験を行った.

5.3.2.1 実験手順

i. バスアイドル時に, 他のメッセージと衝突しないよう十分留意し, 攻撃者のデータ改変部でバスの電圧を改変する. 改変の仕方は, ID:0x0b4, データ「FF FF FF FF FF FF FF FF」のデータフレームの SOF から ITM までの形にする.

ii. 攻撃者のデータ改変部による 1 フレーム分の改変が終わったところで攻撃者から ID:0x0b4 で送信を行う.

iii. i の改変を 50, 100, 300 回行う.

5.3.2.2 実験結果・考察

i で 1 メッセージ分の改変を行うと, 改変を終えた直後にバス上にアクティブエラーフレームが送信された. また, ii で攻撃者から ID:0x0b4 で送信を行うと不正送信阻止された.

同様に, iii で 50, 100, 300 個分の改変を行ったが, いずれの場合も改変を終えた直後にバス上にアクティブエラーフレームが送信された. また, ii で攻撃者から ID:0x0b4 で送信を行うと不正送信阻止された.

このことから, 強いリセッシブを用いた電気的データ改ざんでは, 被攻撃者はエラーアクティブのまま遷移することなく, バス上にアクティブエラーフレームが出てしまうことがわかった. この原因は不正送信阻止のためのエラーフレーム送信ではエラーカウンタ値が増加しない設定であることが影響していると考えられる. しかし, たとえエラーカウンタが増加する場合であっても, REC が増加すると考えられるため, 被攻撃者がバスオフすることはないと考えられる.

また, 実験中に被攻撃者の周期送信が止まることはなく, 正規のメッセージと不正メッセージが混在することも確認された.

6 まとめ

ここまでにあげた保護手法について, それぞれどの攻撃のどの範囲まで保護できるのかについて表 4 にまとめ, 考察する. 表 4 の中で, ”メッセージの改変”はバス上に流れているメッセージの改変を意味し, ”ビットモニタリング”は送信ノードが行っているビットモニタリングを, ”通報用別チャネル”は CAN 以外のチャネルで攻撃の通報ができることを意味する. エラーカウンタ値増加(準備)は不正送信阻止ノードのエラーカウンタ値を

表 4:各保護手法の効果

保護手法	なりすましメッセージ送信			メッセージ改変			サンプリングポイントのずれ			強いリセッシブ			エラーカウンタ値増加(準備)			エラーカウンタ値増加(なりすまし)		
	検出	通報	阻止	検出	通報	阻止	検出	通報	阻止	検出	通報	阻止	検出	通報	阻止	検出	通報	阻止
暗号技術	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
IDS	△	△	-	△	△	-	△	△	-	△	△	-	△	△	-	△	△	-
ビットモニタリング	-	-	-	○	○	○	×	×	×	-	-	-	○	×	×	-	-	-
不正送信阻止	○	○	○	○	○	○	×	×	×	○	○	×	○	×	×	○	×	×
通報用別チャネル	×	×	-	○	○	-	×	×	-	○	○	-	○	○	-	○	○	-

○:保護手法で可能, ×:保護手法で不可能, △:方法次第, -:保護手法の対象外

上げるための改変を意味し，エラーカウンタ値増加（なりすまし）は準備後のなりすましメッセージ送信を意味する．

この表から読み取れることとして，暗号技術を用いた場合にはいずれの攻撃も防ぐことができること，IDS ではどういった手法で侵入を検知するかによって結果が異なることがあげられる．不正送信阻止手法では，”サンプリングポイントのずれ”，”強いリセッシブ”，”エラーカウンタ値増加”の 3 つに対しては，”強いリセッシブ”の検出と通報，”エラーカウンタ値増加”の検出しかできず，いずれも阻止できないことがわかる．ただし，通報用別チャネルを併用した場合には，”エラーカウンタ値増加”の通報は行えるようになる．

これらのことから，現状では暗号技術を用いた保護手法が最も有効であると考えられるが，鍵管理やトラフィックの増加などの課題も多いため，より有効な防御手法について今後も検討していく必要がある．

謝辞 本研究の一部は，文部科学省国立大学改革強化推進事業の支援を受けて行われた．

参考文献

[1] Karl Koscher, Alexei Czeskis, Franziska Roesner , Shweatak Patel, Tadayoshi Kohno, Stephen Checkoway , Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham , and Stefan Savage , “Experimental security analysis of a modern automobile,” IEEE Symposium on Security and Privacy 2010, pp. 447 - 462, 2010.

[2] Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, Stefan Savage, Karl Koscher, Alexei Czeskis, Franziska Roesner, Tadayoshi Kohno, “Comprehensive Experimental Analyses of Automotive Attack Surfaces” , USENIX Security, August 10 - 12, 2011.

[3] M. Wolf, A. Weimerskirch, and C.Paar, “Secure In-Vehicle Communication,” Embedded Security in Cars — Securing Current and Future Automotive IT Applications, 2006.

[4] D. K. Nilsson, U. E. Larson, E. Jonsson, "Efficient In-Vehicle Delayed Data Authentication Based on Compound Message Authentication Codes," Vehicular Technology Conference VTC 2008, 2008.

[5] Tobias Hoppe, Stefan Kiltz, and Jana Dittman, “Applying Intrusion Detection to Automotive IT-Early Insights and Remaining Challenges,” Journal of Information Assurance and Security (JIAS), pp.226 - 235, 2009.

[6] 畑 正人，田邉正人，吉岡克成，大石和臣，松本 勉,“不正送信阻止：CAN ではそれが可能である,” 情報処理学会コンピュータセセキュリティシンポジウム CSS2011, pp.624-629, 2011.

[7] 畑正人，田邉正人，吉岡克成，松本勉，“CAN における不正送信阻止方式の実装と評価,”電子情報通信学会技術研究報告 ISEC2012-72, pp.15 - 22, 2012.

[8]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,”情報処理学会コンピュータセキュリティシンポジウム CSS 2014 論文集, pp. 635 - 642, 2014.

[9]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,” 電子情報通信学会暗号と情報セキュリティシンポジウム SCIS2015, 2015.

[10]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 ICSS2014-74, pp.67 - 72, 2015.

[11] Vector, CANoe https://vector.com/vj_canoe_jp.html (last visit 2015/08/12).

[12] Vector, VN1600 , https://vector.com/vj_vn1600_jp.html (last visit 2015/08/12).

[13] Vector, CANstressD, CANstressDR https://vector.com/vj_canstress_jp.html (last visit 2015/08/12).

[14] OpenCores, http://www.opencores.org/projects,can/

[15]Philips, SJA1000 Stand-alone CAN Controller, 2000. http://www.nxp.com/documents/data_sheet/SJA1000.pdf