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

松本 勉　向達泰希　土屋 遊　中山淑文　吉岡克成

横浜国立大学 大学院 環境情報学府・環境情報研究院

240-8501　神奈川県横浜市保土ヶ谷区常盤台 79-7

tsutomu@ynu.ac.jp,

{kodatsu-taiki-km, tsuchuiya-yuu-tg, nakayama-yoshifumi-vg}@ynu.jp,
yoshioka@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り，送信ノードは自らが送信するデータのビットと，バス上の実際の信号をサンプリングした結果のビットとが一致するかを確認しつつ，不一致で誤り生起を検出する．また受信ノードは，自ら計算したCRC値と，受信したメッセージのCRC値を比較し誤りを検出する．しかし，送信ノードと受信ノードが信号をサンプリングする時点が異なる場合には，CANバスを構成する2線間の電位差をタイミングよく意図的に変えることで，誤ったメッセージを検出されずに転送できる可能性がある．この事実を指摘し，実験により電気的な改ざん攻撃を実証するとともに，この攻撃に対してインテグリティ強化を行う方法について論じる．

How to Enhance Integrity of Controller Area Network Against Electrical Data Forgery

Tsutomu Matsumoto Taiki Kodatsu Yuu Tsuchiya Yoshifumi Nakayama Katsunari Yoshioka

Graduate School of Environment and Information Sciences, Yokohama National University
79-7 Tokiwadai, Hodogaya-ku, Yokohama, Kanagawa, 240-8501 JAPAN
tsutomu@ynu.ac.jp,
{kodatsu-taiki-km, tsuchuiya-yuu-tg, nakayama-yoshifumi-vg}@ynu.jp, yoshioka@ynu.ac.jp

Abstract A transmission node detects errors by comparing the data bits that the node try to transmit and that are sampled from the CAN bus. A reception node also does this by comparing CRC values which are calculated from the received message and those which are written in it. Nevertheless, in case the point which a transmission node samples is different from that which reception node does, there is some possibility of altering messages if you can adjust a difference of electric potential in apposite timing. This paper points out this fact and argues how to enhance the integrity of CAN.

1. はじめに

近年の自動車は，様々な機器をコントロールするECU（Electronic Control Unit）が多数搭載され，それらが相互に通信することによって高度な制御を行っている．多くの自動車のECU同士の通信には，主にCAN（Controller Area Network）というバス型の通信ネットワークプロトコルが利用されており，次世代車載ネットワークとして注目されているFlexRayや車載Ethernetが導入されたとしても，当面は車載ネットワークの一部としてCANが利用され続けると予想される．CANのセキュリティ上の脅威は明らかになっている．文献[1]は自動車の診断用ポートを介して車載ネットワークに侵入し，ECUのファームウェアを改ざんすることによって，自動車を不正に制御できることを実際の自動車を用いて実証している．また，文献[2]はオーディオシステムやBluetooth，携帯電話などを介して車載ネットワークに侵入することによって，遠隔地から自動車の制御を攻撃可能であることが実証している．文献[3]は，デモシステムにおいて攻撃者がメッセージを挿入することによって，パワーウィンドウやエアバッグ制御システムなどを不正に制御できることを実証している．このように多くの文献でCANプロトコルの脆弱性が指摘されている．

CANは2本のバスの電位差の有無によってデータの送受信を行うシリアル通信プロトコルであり，送信ノードは自らが送信するデータのビットと，バス上に現れた実際の信号をサンプリングした結果のビットとが一致しているかどうかを確認しつつプロトコルを進め，不一致であれば誤りの生起を検出する．また受信ノードは，自らが計算したCRC値と，受信したメッセージに含まれるCRC値を比較することにより，誤りを検出する．しかし，送信ノードと受信ノードが信号をサンプリングする時点が異なっている場合には，CANバスを構成する2線間の電位差をタイミングよく意図的に変えることによって，誤ったメッセージを検出されずに転送できる可能性がある．そこで本稿では，この攻撃を実証してみた．また，インテグリティを強化する方法についても簡単に論じる．

本稿の構成は以下のとおりである．2章で関連研究を紹介し，3章でCANについて説明する．次に，4章でCANの電気的データ改ざんについて述べ，5章で実証実験を示す．その後，6章でCANのインテグリティ強化策を述べ，7章でまとめを行う．

2. 関連研究

車載ネットワークのセキュリティ向上に関する先行研究は，暗号技術を用いた保護手法[4, 5]やIDS（Intrusion Detection System：侵入検知システム）やファイアウォールを利用した保護手法[3, 6]，CANのエラーフレームを利用した不正送信阻止方式[7, 8, 9]，送信ECUによる不正送信フィルタリング[10, 11]，CANハブをインテリジェントにすることによる不正CANデータの抑制[11]などが挙げられる．

暗号技術を用いた保護手法として，文献[4]は，車載ネットワークにおいて共通鍵暗号方式と公開鍵暗号方式を組合せて，高速な暗号通信を実現する方式を提案している．文献[5]はCANのデータフレームにMAC（Message Authentication Code）を付加してメッセージ認証を行う方式を提案している．

IDSやファイアウォールを利用した保護手法として，文献[3]は，CANにおいて不正メッセージを検知する手法について3つのパターンを挙げている．また文献[6]は，周期的に送信されるメッセージの頻度を監視することにより異常を検知する方式を提案し，シミュレーションベースでの評価を行っている．

不正送信阻止方式[7, 8, 9]は，CANがブロードキャスト通信であることを利用し，不正メッセージがバス上に送信された場合，なりすましの対象となったECUがエラーメッセージを送信することにより，不正メッセージの送信が完了する前にこれを破棄する．

送信ECUによる不正送信フィルタリングとして，文献[10]はECU内部のCANコントローラにおいて，自らが送信するメッセージをCAN-ID（後述）でフィルタリングすることにより，あらかじめ設定されているCAN-ID以外のメッセージの送信を防ぐ手法である．また文献[11]は，CAN-IDとメッセージの送信タイミング間隔でフィルタリングする手法を提案し，シミュレーション評価とECUを用いた評価を行っている．

文献[11]は，ネットワークを構成するハブに，入力メッセージのCAN-IDと接続機器との対応関係をホワイトリストにより通過制御するコンポーネントを導入して不正データを抑制する方法を提案している．

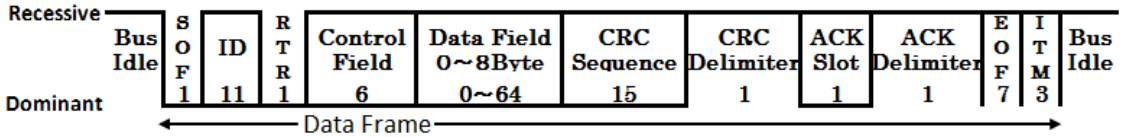

図 1: データフレーム(標準フォーマット)

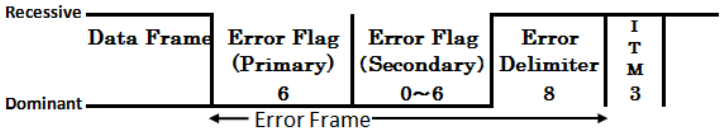

図 2: エラーフレーム

これらの先行研究は, 車載ネットワークに接続された不正 ECU やファームウェアを改ざんされた ECU などによって送信される不正メッセージを検知する方式である. これに対し, 本稿では, CAN バスを構成する 2 線間の電位差を意図的に変えることによって, 誤ったメッセージを検知されずに転送できる可能性を示し, インテグリティ強化方法について論じる.

3. CAN (Controller Area Network)

CAN は, BOSCH 社によって自動車向けに開発され, ISO11898 及び ISO11519 で標準化されている. 2 本のワイヤの電位差によって信号を伝える, 二線式差動電圧方式を採用しており, ノイズへの耐性が強いという特徴がある. 自動車の他に, 飛行機, 産業ロボット, 医療機器, ビルの管理などにおいて, 制御情報の転送を行うシリアル通信プロトコルとして用いられている. バス型のネットワーク・トポロジーを採用しており, ブロードキャストで通信を行うため, バスに接続されている全てのノードがデータを受信する.

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 本のワイヤの電位差が大きい状態をドミナントと呼び, “0”を表す. 電位差の小さい状態をリセッシブと呼び, ”1”を表す. 複数のノードから同時にドミナントとリセッシブがバス上に送信された場合は, ドミナントが優先される仕組みになっている. バスが空いているときは, どのノードもメッセージを送信できるが, 同時に送信が行われた際は, ドミナントが優先され, メッセージ中の CAN-ID と呼ばれるビット列を, 符号なし 2 進数とみなしたときに値が小さい方のメッセージが優先される仕組みとなっている.

3.2 データフレーム

CAN では, フレームに則ったメッセージを送受信することで通信を行っている. フレームにはいくつかの種類があるが, まずデータの送受信に用いられるデータフレーム(図 1)について述べる. 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 シーケンスから構成される.

ID フィールドには, そのメッセージの CAN-ID が記述され, ID フィールドの大きさによって標準フォーマットと拡張フォーマットに分類される. 標準フォーマットでは 11bit, 拡張フォーマットでは 29bit の CAN-ID を記述することができる.

データフィールドには 0～8Byte が格納される.

フレームの SOF から Data Field の終わり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 15 ビットの値が CRC シーケンス部分に格納される.

3.3 エラーフレーム

次に, CAN のフレームの一種であるエラーフレーム(図 2)に関して述べる.

エラーフレームはエラーを全てのノードに伝える機構で, 6～12bit のドミナントによって構成されるフレームである. CAN では, 各 ECU のシステムクロック同期に関係して, 5bit 同じ状態が続いたら反転した状態の 1bit を付加する(例. ドミナントが 5bit 続いたら次の 1bit はリセッシブを送信する), ビットスタッフィングというルールがある.

エラーを検知したノードは, あえて 6bit のドミナント

(この 6bit をプライマリと呼ぶ)を連続して送信し，このビットスタッフィングルールを破ることで，エラーを全てのノードへ伝える．次に，ビットスタッフィングルールが破られたことを検知したノードが 6bit のドミナント(こちらをセカンダリと呼ぶ)を送信し，この時点で全てのノードにエラーが伝わったことになる．

ドミナントの長さが 12bit ではなく，6～12bit で構成される理由は，場合に依って，プライマリとセカンダリが重なる場合があるためである．

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を実装しており，これらのエラーが検出された場合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，自らが送信したビットと，バス上をサンプリングしたビットとの相違をチェックし，相違があればビットエラーとして検出する．
- **アクナレッジチェック**
  送信ノードは，アクナレッジスロットにおいて該当箇所のバス状態がリセッシブであった場合に，アクナレッジエラーとして検出する．
- **CRC チェック**
  受信ノードは，自らが演算した CRC 値と，受信したメッセージに含まれる CRC 値を比較し，一致しない場合は CRC エラーとして検出する．
- **フォームチェック**
  受信ノードは，CRC デリミタ，アクナレッジデリミタ，または EOF において該当箇所のバス状態がドミナントであった場合に，フォームエラーとして検出する．
- **スタッフチェック**
  CANでは，同じ値のビットが5つ続いた場合に，それまで送信されていた状態と反対の状態のビットを 1 つ挿入するビットスタッフィングルールを採用している．受信ノードは，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，同じ値のビットが 6 つ続いた場合はスタッフエラーとして検出する．

3.5 ビットタイミング

CAN メッセージの 1 ビットは 4 つのセグメント，すなわち SS (Synchronization Segment)，PTS (Propagation Time Segment)，PBS1 (Phase Buffer Segment 1)，PBS2 (Phase Buffer Segment 2) から構成されている(図 3)．これらのセグメントは Tq (Time Quantum) という最小単位で構成されている．この構成をビットタイミングという．1 ビットの Tq 数は 8～25 である．

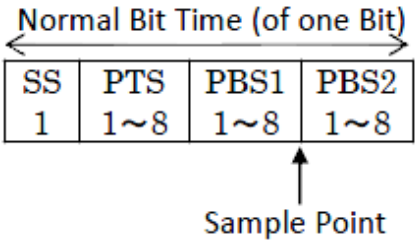

図 3： Sample Point

SS は，バスに接続する複数の ECU がビットの送受信の開始位置とするセグメントであり，Tq 数は 1 である．PTS は，ネットワーク上の各種要因(送信ユニットの出力遅延，CAN バスでの信号伝播遅延，受信ユニットの入力遅延)を補償するセグメントであり，Tq 数は 1～8 である．PBS1，PBS2 は，各ノードの発振子による発振誤差を補償するセグメントであり，PBS1 の Tq 数は 1～8，PBS2 の Tq 数は 2～8 である．誤差に合わせて PBS1 を延ばしたり，PBS2 を縮めたりすることによって同期を合わせるが，Tq 数が SJW (reSynchronization Jump Width) として定められた値以上の誤差が生じた場合は，SJW の Tq 数分しか補正しない．SJW の Tq 数は 1～4 の範囲内で，CAN コントローラで設定される．

ECU が信号をサンプリングする時点(Sample Point という)は，PBS1 の直後にある．

4. CAN メッセージの電気的改ざん

4.1 攻撃の可能性

CAN では，ネットワークの遅延などを考慮して，CAN メッセージの 1 ビットの構成(ビットタイミング)を決定する．遅延や誤差は各セグメントによって補償されるが，ネットワークに接続される複数の ECU のそれぞれの Sample Point が異なる可能性がある．送信ノードと受信ノードの Sample Point が異なるとき，受信ノードの Sample Point のタイミングで CAN バスを構成する 2 線間の電位差を変えることによって，送信ノードのビットモニタリングによるエラー検知を回避しつつ，受信ノードに誤ったメッセージを届けるこ

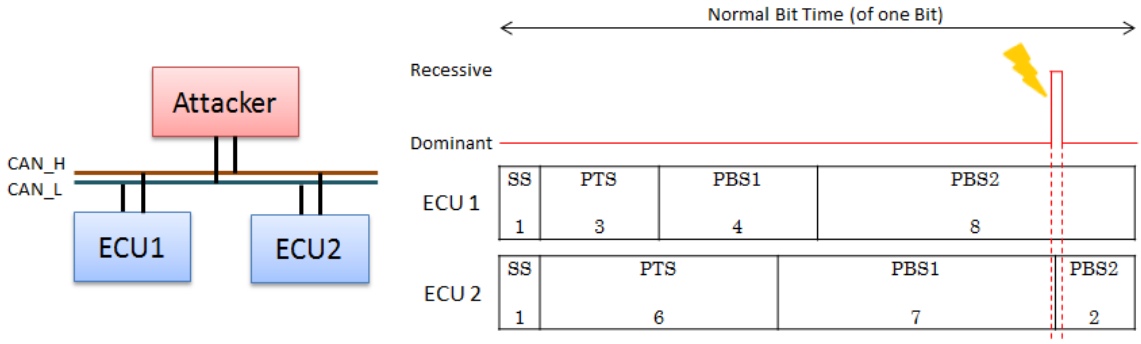

図 4: CAN メッセージの電気的改ざん

とができるはずである. さらに, Sample Point が受信ノードと異なるノードには, 正規のメッセージがバスに流れているように騙すことが可能である.

例えば, CAN バスに ECU1 という送信ノード(Sample Point:50%)と ECU2 という受信ノード(Sample Point:87.5%)が繋がっているとき, 反転させたいビットの 87.5%付近の電位差を変化させることによって, ECU1 に検知されることなく, ECU2 に誤ったデータが送信される可能性がある(図 4). ただし, メッセージの CRC フィールドも整合性が取れるように変化させる必要がある.

4.2 攻撃の特徴

この攻撃の方法は, 文献[1, 2, 3]などで述べられている単純なりすまし攻撃とは異なり, バスに流れるメッセージをアナログ的に改ざんするものである. このため, たとえば既存の対策手法の一つである, CAN バスに接続されたノードが不正送信を検知する IDS は, 検知する機構の Sample Point が, 受信ノードの Sample Point と異なる場合には, 攻撃を検知できない.

なお, 攻撃経路として一般的な

① OBD-II ポートなどの外部接続用ポートに接続し, 直接的に不正通信を行う.
② CAN バスに不正なノードを直接接続することで, 不正通信を行う.
③ 正規の ECU のファームウェアをマルウェア感染等により不正改ざんし不正通信を行う.

の中で, ③においては行えない. 4.1 節で述べた攻撃は, 攻撃を行う不正ノードに CAN バスを電気的に変える機能が求められるため, 直接 CAN バスに物理的に接続する必要があり, ①②に限定される.

4.3 ビットスタッフィングルールの考慮

メッセージの改ざんを行う際, ビットスタッフィングルールも考慮しなければならない.

例えば, 改ざん後のデータにビットスタッフィングの対象となるビット列が新たに発生した場合, その後のメッセージを全てルールに従うように改ざんを行う必要が生じる. 逆に, 改ざん後のメッセージにはスタッフビットが必要なくなる場合も考えられる. これらの要因によりメッセージの長さが変化した場合は, ACK スロットにも考慮する必要が生じる. 送信ノードでは ACK スロットでドミナントを受信できないと, 正しくメッセージを送信することができなかったとしてメッセージの再送を行う. これを防ぐためには, 送信ノードの Sample Point で電位差を操作し, ACK スロットを改ざんすればよい.

5. 電気的データ改ざん実験

本章では, 前章で述べた CAN の電気的データ改ざん攻撃の実証を試みた結果を示す.

5.1 概要

実験構成図を図 5 に示す.

CAN バスに, CAN-ID : 0x06F, Data : 0x58 のメッセージを 1sec 毎に周期的に送信する実機の ECU を接続し, 送信ノード(Sample Point:50%)とした. 送信ノードは, ECU はマイコン ATmega162, CAN コントローラ SJA1000, CAN トランシーバ TJA1050 を用いて製作した.

ECU およびネットワークの開発ツールである Vector 社の CANoe[13]と CAN インターフェイスである VN1630A を用いて, PC を CAN バスに接続し, 受信ノード(Sample Point: 87.5%)とした.

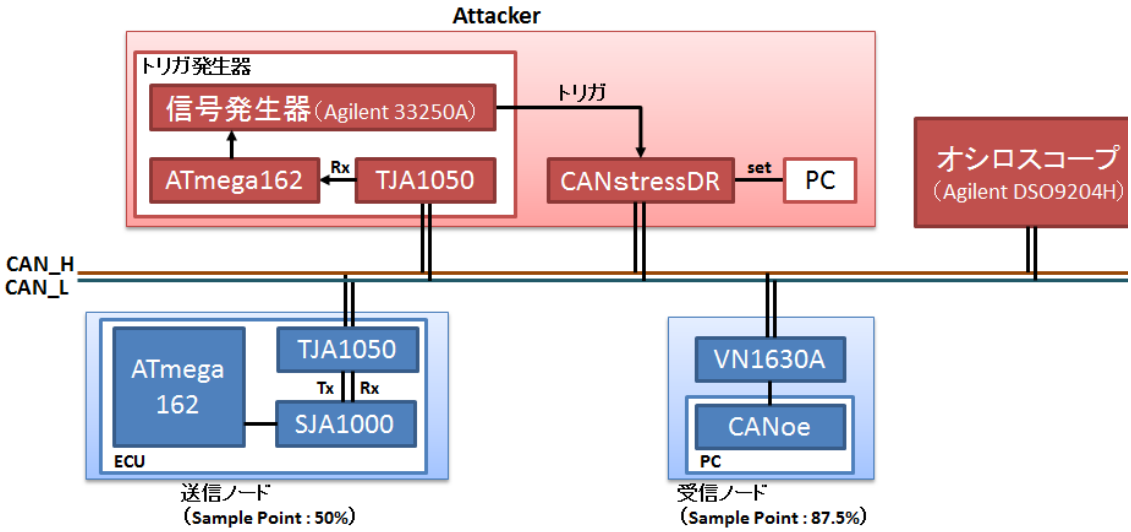

図 5: 実験構成図

CAN バスのボーレートは高速であるほど，1bit の幅が短くなるため，その分攻撃の難易度はあがる．今回の実験では，攻撃の可能性を明らかにするため，比較的低速である 125kbps とした．このネットワークに攻撃を実装する．

攻撃を行うために，バスに流れるメッセージを観測し，攻撃タイミングを特定して，電位差を変える必要がある．そのためのツールとして，マイコン ATmega162，CAN トランシーバ TJA1050，信号発生器（Agilent 33250A）によって構成されるトリガ発生器を作り，CAN バス上に電気的な障害を発生させる CAN ネットワークのテストツールである Vector 社の CANstressDR[14]を使用した．

トリガ発生器は CAN バスの電位差を変えるタイミングを特定し，CANstressDR にトリガを与える．CANstressDR はトリガを受け取っている間，CAN バスの電位差をドミナントからリセッシブへと変化させる．

5.2 攻撃のメカニズム

送信ノードから送信される CAN メッセージのデータ部分を 0x58 から 0x59 へ改ざんする攻撃を行う．CAN バスにトリガ発生器，CANstressDR からなる不正ノードを接続する．トリガ発生器はバスに流れる CAN メッセージの SOF のビットを検知し，データ部分の先頭から 8bit 目の 87.5% 付近で CANstressDR にトリガを与える．さらに，データ改ざん前のメッセージの CRC が 0x2842，データ改ざん後のメッセージの CRC が 0x6ddb であることを予め計算し，CRC の整合性を取るため，データ部分と同様に CRC シーケンスの先頭から 1, 5, 7, 8, 11, 12, 15bit 目のそれぞれの 87.5%付近でトリガを与える．トリガ発生器からトリガを与えられた CANstressDR は，トリガを与えられている間，CAN バスのレベルをドミナントからリセッシブへ電気的に変更する．

この時，CAN バスの電位差を変えることにより受信ノードと送信ノードの同期がずれる．このため，受信ノードが出す ACK のタイミングが通常時より早まり，送信ノードは CAN デリミタのタイミングで ACK を受け取り，フォームエラーとして検出される．今回の実験では，CAN デリミタ部分にも CAN バスの電位差をリセッシブへと変えることによって，フォームエラーを回避している．

5.3 実験結果

受信ノードに届いたデータを図 6 に示す．通常は Data:0x58 のメッセージが届いていたが，攻撃実行

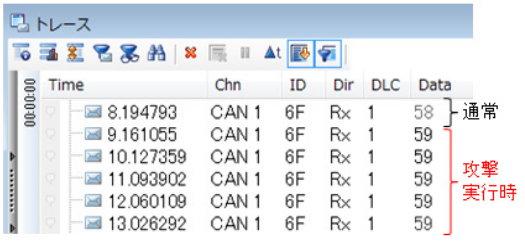

図 6: 受信ノードに届いたデータ

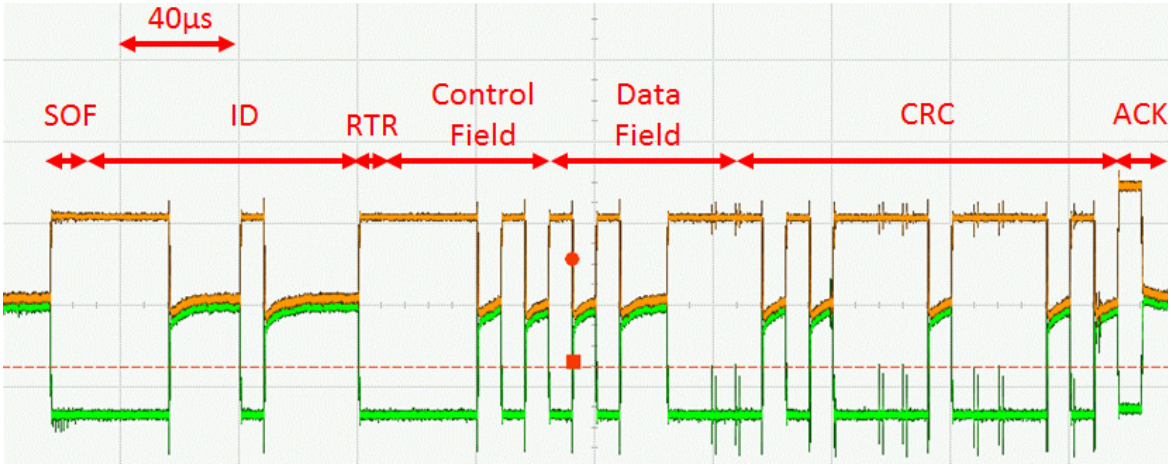

図 7: 通常時の電圧波形

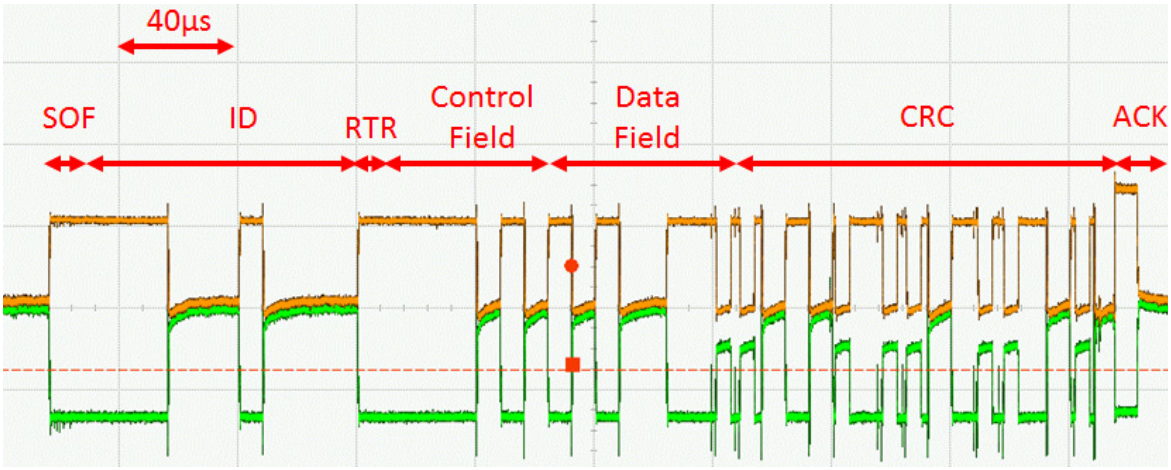

図 8: 攻撃実行時の電圧波形

時には Data:0x59 のメッセージが届いた. オシロスコープ(Agilent DSO9204H)で観測した通常時の波形を図 7 に, 攻撃実行時の波形を図 8 に示す. また, 攻撃実行時に送信ノードで観測されるビット列は表 1 のとおりであり, 受信ノードで観測されるビット列は表 2 のとおりである.

このことから送信ノードと受信ノードの Sample Point に差がある場合に, それに応じて, CAN バスの電位差をタイミングよく変えることによって, データの改ざんを行えることが実証できたといえる.

6. インテグリティ強化策の検討

電気的データ改ざんに対して, CAN のインテグリティをどのように強化すべきかを検討する.

- **Sample Point の調整**

  受信ノードと送信ノードの Sample Point を同時にすることによって, 送信ノードのビットモニタリングが攻撃を検知する. しかし, CAN の伝送遅延などから, 完全に一致させることは困難であると推測する.

- **MAC の付加**

  CAN のメッセージに MAC (Message Authentication Code)を付加させることによるメッセージ認証を行った場合には, 攻撃者は MAC に用いられる鍵を奪取している場合や, MAC を生成する要素にタイムスタンプやシーケンス番号が含まれずメッセージの再送攻撃ができてしまう場合を除き, 改ざん後のデータの MAC が計算できないため, 受信側でデータ改ざんを検知することが可能である. CAN への MAC の導入検討は自動車業界で進められていると聞く. 多くの自動車の CAN に MAC 導入が進むことを期待したい.

表 1： 送信ノードで観測されるビット列

SOF	0
ID	00001101111
RTR	0
Control Field	0000101
Data Field	01011000
CRC	0101000010000101
ACK	01

表 2： 受信ノードで観測されるビット列

SOF	0
ID	00001101111
RTR	0
Control Field	0000101
Data Field	01011001
CRC	1101101110110111
ACK	01

7. まとめ

本稿では，多くの自動車の車載ネットワークとして利用されている CAN に対して，送信ノードと受信ノードのビットを取得するタイミングの違いを利用して，送信ノードのエラーチェックを回避しつつ，受信ノードに誤ったデータを転送できる可能性について述べた．さらに，実証実験によって電気的データ改ざんが可能であることを示した．また CAN のインテグリティ強化方法について述べた．

今後は，インテグリティ強化策の詳細なアーキテクチャの考案と実装による評価を行う必要があると考えている．

参考文献

[1] K. Koscher, et al., “Experimental Security Analysis of a Modern Automobile,” IEEE Symposium on Security and Privacy 2010, pp.447-462, 2010.

[2] S. Checkoway, er al., “Comprehensive Experimental Analyses of Automotive Attack Surfaces,” the 20th USENIX Security Symposium, 2011.

[3] Tobias Hoppe, Stefan Kiltz and Jana Dittmann, "Security Threats to Automotive CAN Networks -Practical Examples and Selected Short-Term Countermeasures," the 27th international conference on Computer Safety, Reliability, and Security, SAFECOMP '08, pp. 235 –248,2009.

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

[5] 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.

[6] 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.

[7] 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 CSS 2011, pp.624-629, 2011.

[8] T. Matsumoto, M. Hata, M. Tanabe, K. Yoshioka, K. Oishi, “A Method of Preventing Unauthorized Data Transmission in Controller Area Network,” the 75th IEEE Vehicular Technology Conference, 2012.

[9] 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.

[10] 田邉正人, 畑 正人, 吉岡克成, 松本 勉, “CAN コントローラにおける不正送信フィルタリング,” 電子情報通信学会 SCIS 2013, pp.624-629, 2013.

[11] 関口大樹, 畑 正人, 田邉正人, 吉岡克成, 松本 勉, ”不正 CAN 通信阻止のための ECU 内蔵監視機構,” 電子情報通信学会技術研究報告 IT2012-62, pp.203-210, 2013.

[12] 関口大樹, 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,” 電子情報通信学会 SCIS 2014, 2014.

[13] Vector, CANoe, https://vector.com/vj_canoe_jp.html

[14] Vector, CANstressD および CANstressDR, https://vector.com/vj_canstress_jp.html