NAIST-IS-MT1651040

修士論文

車載ネットワークにおける信号の物理的特徴を利用した攻撃ノード識別手法の提案

北川 智也

2018 年 3 月 15 日

奈良先端科学技術大学院大学
情報科学研究科

本論文は奈良先端科学技術大学院大学情報科学研究科に
修士 (工学) 授与の要件として提出した修士論文である。

北川 智也

審査委員：

藤川 和利 教授	（主指導教員）
林 優一 教授	（副指導教員）
新井 イスマイル 准教授	（副指導教員）

車載ネットワークにおける信号の物理的特徴を利用した攻撃ノード識別手法の提案*

北川 智也

内容梗概

自動車における制御システムの電子制御化が，近年急速に進められている．今日の自動車は多数の Electronic Control Unit (ECU) からなり，ECU 同士が複数の通信プロトコルを用いて，制御に必要な情報をやり取りする車載ネットワークを構成している．車載ネットワークの中でも，Controller Area Network (CAN) が広く用いられている．CAN バスは同じ信号線に全てのノードが接続されるため，不正メッセージの送信が行われた際にメッセージ送信元の識別が困難である．CAN メッセージの送信元識別手法を確立すれば，不正メッセージの検知・無効化に応用できる．既存研究として，CAN ノード毎のクロックのずれから生じる Clock skew を利用して送信元を識別する手法や，CAN バスの電圧を測定し CAN ノードの個体差を特徴付けすることで送信元を識別する手法がある．しかし，Clock skew に着目する手法は周期的ではない CAN メッセージに適用できない．また電圧測定の手法では，通信手順の変更が必要であったり，プロファイル更新に複数回測定する必要がある問題がある．本研究では，CAN バスにおける送信元ノードの識別を目的とし，信号の遅延時間が CAN ノード毎に異なる点に着目し，CAN メッセージの送信元の識別手法を提案する．提案手法のプロトタイプを試作し，実験室と実車で ECU 分類の評価実験を行い，0.97 前後の平均正答率となった．

キーワード

車載ネットワーク, Controller Area Network, ECU 識別

*奈良先端科学技術大学院大学 情報科学研究科 修士論文, NAIST-IS-MT1651040, 2018 年 3 月 15 日.

Attacker identification method using physical characteristics of the signal on in-vehicle network*

Tomoya Kitagawa

Abstract

Modern vehicles are equipped with several dozen Electronic Control Units (ECUs). The Controller Area Network (CAN) is one of the common in-vehicle network protocol. It is difficult to identify the sender of the CAN messages because all CAN nodes are connected to the same wire. Hence, it is significant that establishing a source identification method because that can be applied to Intrusion Detection System (IDS). In earlier study, researchers proposed a method of identifying the sender CAN node by detecting Clock skew. However, the method cannot be applied to non-periodic CAN messages. Some researchers also proposed a method of identifying a sender node by measuring the voltage of the CAN bus and characterizing the individual differences. However, the method has a problem on practicability. In this research, we focus on the fact that the signal delay time for each CAN node are different and propose a method to identify the source of the CAN message. We made a prototype device of the proposed method and evaluated ECU classification accuracy on the CAN bus test bench we made and real vehicle. We obtained a mean accuracy rate of about 97%.

Keywords:

In-vehicle network, Controller Area Network, Fingerprinting ECU

*Master's Thesis, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-MT1651040, March 15, 2018.

目 次

図 目 次

表 目 次

1. 序論

自動車の制御システムは，排ガス規制への対応や利便性･安全性向上のため，電子制御化が急速に進められてきた．自動車に搭載される電子制御ユニット (Electronic Control Unit; ECU) の数は年々増加しており，車種によって大きな差があるものの，2016 年の時点で自動車に 1 台当たり平均 21.6 個の ECU が搭載されている [1]．2025 年には 30.4 個になると予測されている．高級車では 100 個以上の ECU が搭載されるものもある．近年の自動車には，先進運転支援システム (Advanced Driving Assistant System; ADAS) と呼ばれる機能が搭載されるようになった．ADAS の例として，アダプティブ･クルーズ･コントロールや衝突被害軽減ブレーキ，車線逸脱防止システムが挙げられる．アダプティブ・クルーズ・コントロールは，あらかじめ設定した速度内で前方車両との車間距離を維持しながら追従走行するシステムである．衝突被害軽減ブレーキは，前方車両との距離を検知して，システムが衝突を回避できないと判断した際に自動的にブレーキを作動させるシステムである．車線逸脱防止システムは，カメラ画像を基にレーンマーカーを認識し，車両がレーンマーカーに近づいたときに運転者の意図した車線変更でないとシステムが判断した場合に，警報で運転者に知らせるとともに車両を車線内に戻すようにパワーステアリングを制御するシステムである．このような高度に電子化された今日の自動車は，各種センサ･モーター，ステアリング，エンジン，ブレーキといった機器同士がハーネスと呼ばれる多数の電線で接続され，車載ネットワークを構成している．また，コネクティッドカーと呼ばれる通信機能を搭載した自動車の登場により車載ネットワークと外部ネットワークとが接続される機会が増えている．さらに，自動運転技術の採用により，今後ますます自動車の電子制御システムは高度化・複雑化が進むと考えられる．

最新型でない車両に関しても，車載ネットワークと外部ネットワークとが接続される機会がある．例として，運転状況を保険料に反映するテレマティクス保険や，車両情報を故障診断などに活用できるサービスの登場が挙げられる．GMO クラウド株式会社は「LINKDrive byGMO」と呼ばれる，車両に備えられている On-board diagnostics II (OBD-II) コネクタ[1] に接続したデバイスから取得した車

[1]故障診断端子．2.1 節で解説する．

両情報をスマートフォン経由でクラウドに送信し，車両のコンディションを遠隔診断するサービスを提供している [2]．株式会社スマートドライブは「DriveOn」と呼ばれる，OBD-II コネクタに接続したデバイスから取得した車両情報を活用して，安全運転や低燃費運転を支援するサービスを提供している [3]．

一方で，近年車載ネットワークへの攻撃により，自動車が乗っ取られる可能性があることが多数報告されている．Koscher らは，車載ネットワークプロトコルの一つである Controller Area Network (CAN) バスに直接不正なメッセージを送ることで，エンジンやワイパー，ドアロックなどを制御できることを示した [4]．OBD-II コネクタには認証機構がなく，誰でも入手可能な市販ツールを用いて車載ネットワークに攻撃が可能であることを指摘した．Checkoway らは，Koscher らの研究における CAN バスへの攻撃は，車両への物理的な接触を要するという前提は非現実的だとして，車両に接触することなく車載ネットワークに侵入可能であることを実験的に示した [5]．車両で利用される Wi-Fi や Bluetooth といった，様々な無線インタフェースや車載情報端末ユニット (In-Vehicle Infotainment; IVI) で使用される CD や USB メモリなどを経由して攻撃される可能性を指摘した．Miller らは，2015 年に Jeep Cherokee の ECU に脆弱性を発見し，携帯電話網からアクセスできる状態であることを発見した [6]．無改造の車両に対して，携帯電話網を通じて，ECU のファームウェアを書き換えたうえで，車両の制御を遠隔から制御するデモンストレーションを行った．結果的に，メーカーは 140 万台をリコールする事態となった．Keen Security Lab は，2016 年に最新のファームウェアがインストールされた未改造の Tesla Model S を遠隔から乗っ取ることができたと発表した [7]．約 19 km 離れた場所から，走行中の車両のブレーキを作動させるデモンストレーションが行われた．さらに，後付けのデバイスを経由して車載ネットワークに侵入できることも指摘されている．Foster らは Mobile Devices 社製の OBD-II コネクタに接続するデバイスに SMS メッセージを利用して遠隔から車載ネットワークに侵入できる脆弱性があることを報告した [8]．スマートフォンから SMS メッセージを送信して遠隔からワイパーを操作したりブレーキをかけるデモが行われている．このように，後付けのデバイスを用いたサービスに脆弱性が発見されると車載ネットワークへの侵入を許すことになる．我が国に

おける，2017 年 3 月末の乗用車 (軽自動車を除く) の平均使用年数は 12.91 年と長く [9]，設計時には想定されていなかったセキュリティ上の脅威にさらされる問題がある．このように，自動車のセキュリティ対策が急務となっている．また，既存の車両に対しても適用可能な車載ネットワークへのセキュリティ対策が必要となっている．

CAN はマルチマスタかつ，ブロードキャスト型のバスプロトコルである．CAN フレームには，メッセージの送信元を示すフィールドや認証のためのフィールドが存在しない．そのため，なりすまし攻撃や DoS 攻撃[2]が行われた際に，攻撃を行っている送信元ノードを容易に特定できない．

これまでに，CAN メッセージの送信元を特定する研究がいくつかなされてきた．しかし，周期的に送信される CAN メッセージから Clock skew を抽出する方法 [10] は，周期的ではないメッセージに適用できない問題がある．また，CAN バスの電圧から特徴抽出を行い，ECU の識別を行う方法 [11, 12, 13] では，通信手順の変更やが必要であったり，プロファイルの更新に複数回測定をする必要があるといった問題がある．CAN メッセージの送信元を特定する手法を確立すれば，攻撃メッセージを検知・無効化する Intrusion Detection System (IDS) に応用することができる．また，送信元特定手法はインシデント発生時の問題に切り分けへの応用も考えられる．

CAN ノードには，論理レベルと CAN バスレベルの変換を行う CAN トランシーバが搭載されている．CAN トランシーバにおける信号の立ち下がり時の遅延時間と立ち上がり時の遅延時間は等しくない．そのため受信ノードから見ると，理想的なビット時間に対してビットの時間幅にずれが生じる．そのずれは CAN ノードごとに個体差があることに着目した．本研究では，CAN バスにおける攻撃の識別を目的とし，CAN トランシーバの遅延時間の特徴について明らかにした上で，CAN メッセージの送信元を識別する手法を提案する．提案手法の有効性を確認するために，CAN トランシーバの遅延時間を 10 ns の精度で計測するプロトタイプデバイスを試作した．評価実験として，実験室で構築した CAN ノードおよび実車での計測実験を行い，提案手法の有効性を確認した．

[2]Denial of Service 攻撃．サービスを妨害し，通信を不可能な状態にする攻撃．

本論文の構成を以下に述べる．2 章で車載ネットワークとして用いられている通信プロトコルについて概観する．3 章では，CAN バスに対する攻撃への対策の関連研究を IDS と CAN バスの物理的特徴に着目した送信元識別手法に大別して述べる．4 章では，CAN トランシーバの遅延時間を利用した攻撃ノード識別手法を提案する．5 章では，提案手法のプロトタイプの試作について述べる．6 章では，提案手法のプロトタイプの評価を行う．最後に 7 章で本研究についてまとめる．

2. 車載ネットワーク

本章では，車載ネットワークとして用いられているプロトコルについて概観する．特に研究目的で対象とする CAN プロトコルについて，フレームの構造や攻撃モデルについて詳しく述べる．

車載ネットワークの構成例を図 1 に示し，それぞれの車載ネットワークで用いられる通信プロトコルについて簡単に述べる．

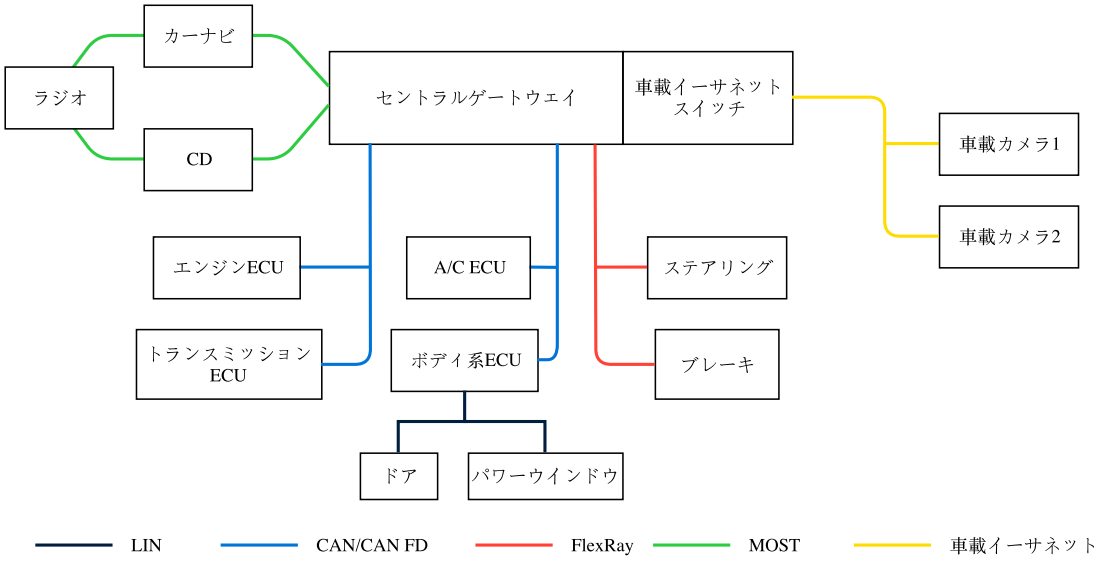

図 1　車載ネットワークの構成例

Controller Area Network (CAN)

CAN はマルチマスタ型のシリアル通信プロトコルである．ドイツの Bosch 社が提唱し [14]，低速 CAN は ISO[3]11519 として，高速 CAN は ISO11898 として規格化されている．1 フレームの最大データ長は 8 バイト，最大通信速度は 1 Mbps となっている．高い耐ノイズ性や強力なエラー検知機能を持ち，ノードの追加・削除が容易にできる特徴がある．

CAN with Flexible Data Rate (CAN-FD)

CAN-FD はより高速・大容量な用途への需要に対応するために CAN を拡

[3]International Organization for Standardization: 国際標準化機構.

張したプロトコルである．1 フレームの最大データ長は 64 バイトに拡張され，最大通信速度は 5 Mbps となっている．

Local Interconnect Network (LIN)

LIN は 1 線式のマスタスレーブ型のシリアル通信プロトコルである．最大 20 kbps の速度で通信できる．スレーブでは水晶発振器が不要という特徴があり，CAN と比較して低コストである．ドアロック制御やパワーウインドウ制御などに用いられる．

Media Oriented System Transport (MOST)

MOST はカーオーディオやカーマルチメディアシステムなどのエンターテインメント系の接続に用いられる．物理的な接続には光ファイバーまたはツイストペア線を使用する．MOST には，MOST25，MOST50，MOST150 の 3 種類の規格がある．最初の版である MOST25 は 24.5 Mbps の速度で通信できる．MOST50 は 50 Mbps，MOST150 は 150 Mbps の通信速度で通信できる．

FlexRay

FlexRay は CAN よりも高速・信頼性の高い通信プロトコルである．最大 10 Mbps の速度で通信できる．通信タイミングにタイムトリガ方式を採用しており，一定の周期で発生する固定長のフレームを使ってデータを伝送する．このため，応答性が保証しやすく，ドライブ・バイ・ワイヤやブレーキ・バイ・ワイヤなどの時間的制約が厳しい用途に採用される．

車載 Ethernet

車載 Ethernet は，従来の Ethernet 規格を車載向けに改良したものである．車載 Ethernet 規格として 100BASE-T1 や 1000BASE-T1 が挙げられる．100BASE-T1 は，1 対のツイストペア線で 100 Mbps の全二重通信ができ，Broadcom 社が開発した BroadR-Reach がベースとなっている．100BASE-T1 は IEEE 802.3bw で標準化されている．一方，1000BASE-T1 は，1 対のツイストペア線で 1 Gbps の全二重通信ができ，IEEE 802.3bp で標準化されている．高帯域を活かして車載カメラ映像の伝送などに用いられる．

2.1 OBD-II

OBD-II (On-board diagnostics II) は，自動車の故障診断機能の規格である [15]．OBD-II は，最初の版である OBD-I の機能向上と標準化をはかった第 2 版である．排ガス規制に適合するための機能に支障が生じた場合に運転者に警報するとともに，当該故障情報を保存できるようにすることを規定している．OBD-II の情報出力方法はコネクタの形状，ピン配列ともに ISO15031-3 で規定されており，このコネクタは OBD-II コネクタと呼ばれる．

OBD-II は，1996 年以降に米国内で販売される乗用車と小型トラックに搭載が義務付けられている．我が国では，J-OBDII という名前で制定されている [16]．J-OBDII に適合した装置は，2008 年 10 月以降の新型車 (輸入車等を除く) 及び 2010 年 9 月以降に製造された自動車での搭載が義務付けられている [17]．OBD-II コネクタの例を図 2 に示す．

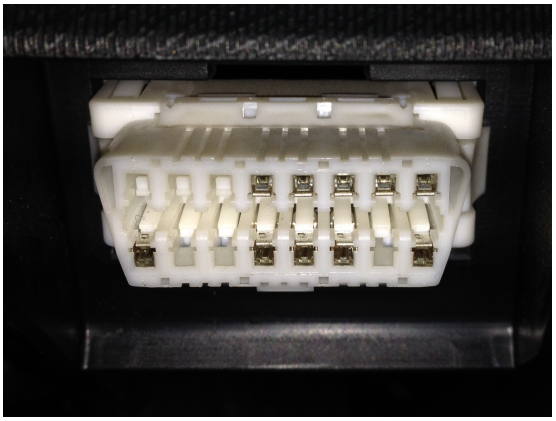

図 2 OBD-II コネクタ

OBD-II 規格において，通信プロトコルは SAE[4] J1850 (1998 年 3 月制定，41.6 kbps)・ISO9141-2 (1994 年 2 月制定，1996 年改正， 10.4 kbps)・ISO14230-4 (2000 年 6 月制定，10.4 kbps)・ISO15765-4 (2005 年 1 月制定，250 kbps または 500 kbps[5]) のいずれかを用いることが要求されている．この中でより新しく制定され，最も通

[4]Society of Automotive Engineers: 米国の自動車技術者協会．

[5]CAN 規格上の最大通信速度は 1 Mbps であるが，OBD-II 規格ではこのいずれかの通信速度を用いることを要求している．

信速度の速い ISO15765-4 (CAN) が広く採用されている．このことから，今日の自動車において CAN は事実上の標準といえる．

ISO15031-3 に規定されている OBD-II コネクタのピン配列を図 3 に，ピン割り当てを表 1 に示す．6 番ピンに CAN-H・14 番ピンに CAN-L が割り当てられている．

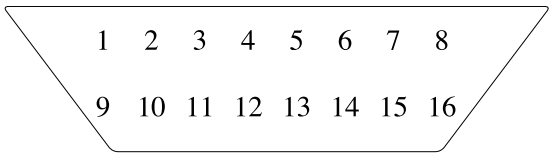

図 3 OBD-II コネクタのピン配列

2.2 CAN (Controller Area Network)

CAN プロトコルには，CAN 2.0A と CAN 2.0B の 2 つのバージョンがある．CAN 2.0A は識別子が 11 ビット長である標準フォーマットのフレームを扱い，CAN 2.0B は識別子が 29 ビット長である拡張フォーマットのフレームを扱う．両者の違いは識別子の長さのみであるため，本節では CAN 2.0A について詳しく述べる．

また，CAN の物理層は，通信速度が 125 kbps–1 Mbps である高速 CAN と通信速度が 10 kbps–125 kbps である低速 CAN の 2 種類がある．それぞれ，ISO11898 と ISO11519-2 で規定されている．本節では広く用いられている高速 CAN について述べる．

典型的な CAN バスの構成を図 4 に示す．典型的な CAN ノードは制御用の Micro Controller Unit (MCU)・CAN コントローラ・CAN トランシーバから構成される．CAN コントローラは CAN プロトコルの処理を行い，CAN トランシーバは CAN バスと CAN コントローラの狭間で論理レベルと CAN バスレベルの変換を行う．図 4 では，CAN コントローラが MCU に内蔵されている例を示しているが，CAN コントローラは外付けのチップとなっており，MCU と SPI[6]通信を行うことで制御する実装となっている場合もある．

[6]Serial Peripheral Interface: MCU と周辺 IC 間のシリアル通信インタフェース．

表 1 OBD-II コネクタのピン割り当て

ピン番号	割り当て
1	任意
2	PWM+
3	任意
4	シャーシグランド
5	シグナルグランド
6	CAN-H
7	K-LINE
8	任意
9	任意
10	PWM-
11	任意
12	任意
13	任意
14	CAN-L
15	L-LINE
16	常時電源

CAN バスは，高い耐ノイズ性を実現するためにノードを 2 本のツイストペア線で接続する．それぞれの線を CAN-L，CAN-H という．バスの両端は，信号の反射を防ぐために 120 Ω の終端抵抗で終端する．そのため，バスの合成抵抗は 60 Ω となる．高速 CAN では，バス上に最大 30 ノードまで接続できる．

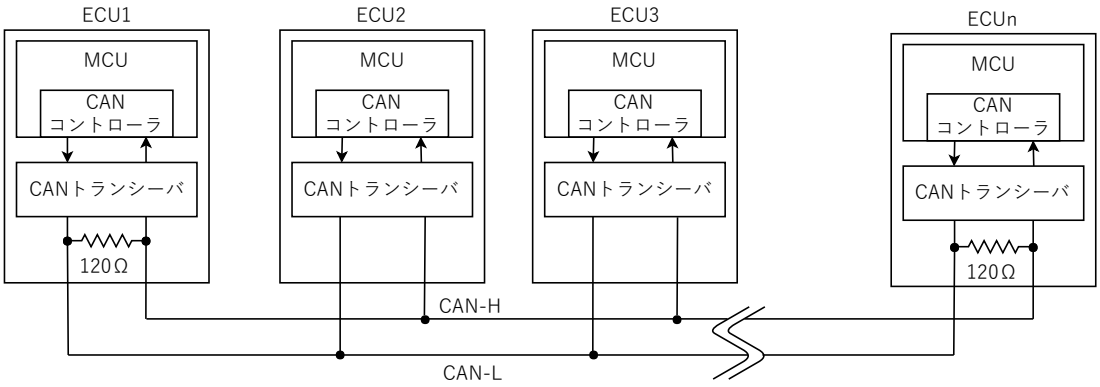

図 4　典型的な CAN バスの構成

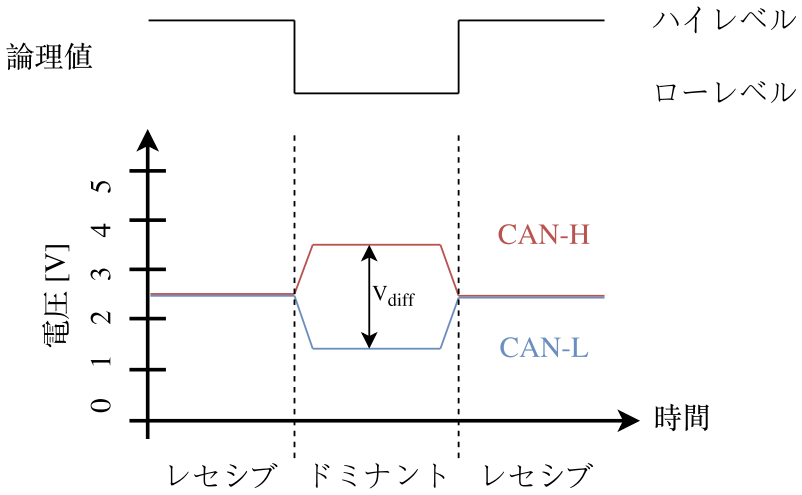

図 5　CAN バスのレベル

グランドと CAN-L ピン間の電圧を V_{CANL}，グランドと CAN-H ピン間の電圧を V_{CANH} とすると差動電圧 V_{diff} は

$$V_{diff} = V_{CANH} - V_{CANL} \tag{1}$$

と定義できる．

CAN バスには，図 5 に示すように，ドミナント (Dominant) とレセシブ (Recessive) の 2 つのバスレベルがある．

ドミナント

V_{diff} が 1.5 V から 3.0 V の状態

レセシブ

V_{diff} が −0.5 V から 0.05 V の状態

CAN プロトコルではビットレベルでは，ローレベルは 0 に，ハイレベルは 1 に対応する．よって，CAN トランシーバは論理値でハイレベルをレセシブ，論理値でローレベルをドミナントへと変換し，CAN コントローラと CAN バス間での信号の受け渡しを行う．複数のノードからドミナントとレセシブが同時に送信された場合は，バスの状態は，ドミナントとなる．CAN プロトコルではこの特徴を利用して，複数のノードが同時にフレームを送信し，信号が衝突したとしても，優先度の高いフレームを送信を中断することなく送信できる．この仕組みはアービトレーションと呼ばれ，詳細は 2.2.2 項で述べる．

2.2.1 フレームの種類

CAN プロトコルでは，データフレーム・リモートフレーム・エラーフレーム・オーバーロードフレームの 4 種類のフレームが規定されている．このうち，データフレームとリモートフレーム以外のフレームは CAN コントローラによって自動的に送信される．

以降，データフレームについて詳細に説明する．データフレームは送信ノードから受信ノードへメッセージを送るためのフレームである．CAN におけるデータフレームの構造を図 6 に示す．各フィールドの下に記載した数字はビット数を表す．データフレームの各フィールドについて簡単に述べる．

スタートオブフレーム (SOF; Start Of Frame)

フレームの開始を表す．1 ビットのドミナントで表される．

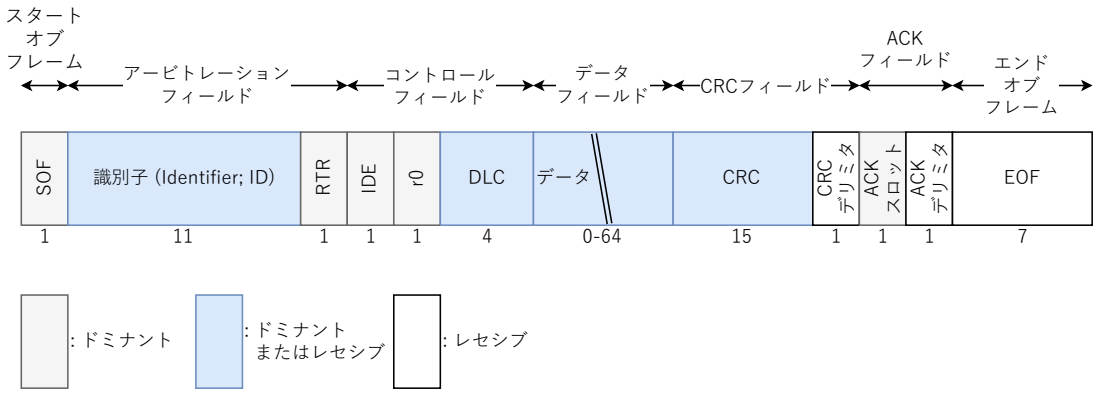

図 6 CAN におけるデータフレームの構造

アービトレーションフィールド

11 ビットの識別子 (Identifier; ID) とフレームの種類を示す RTR が送信される区間で，フレームの優先順位を表す．識別子は，小さい値であるほど高い優先度のフレームとなる．本論文では，識別子を指して CAN ID と呼ぶ．RTR は Remote Transmission Request の略で，データフレームとリモートフレームを識別する．RTR がドミナントであるときデータフレームを表し，レセシブであるときリモートフレームを表す．アービトレーションフィールドとは調停フィールドとも呼ばれる．

コントロールフィールド

IDE，r0 と呼ばれる 2 つの予約ビットとデータ長 (Data Length Code; DLC) が送信される区間である．コントロールフィールドは制御フィールドとも呼ばれる．

データフィールド

データの内容が送信される区間である．この区間は可変長で，CAN では 0–8 バイトのデータを送信できる．

CRC フィールド

フレームの伝送誤りをチェックする区間である．15 ビットの CRC (Cyclic Redundancy Check) と CRC の終了区切りを表す 1 ビットのレセシブ (CRC

デリミタ) からなる．

ACK (Acknowledgement) フィールド

そのフレームを送信しているノード以外の受信ノードが，CRC フィールドまでを正常受信できた場合は，その合図として ACK スロットで 1 ビットのドミナント送信する．

エンドオブフレーム (EOF; End Of Frame)

フレームの終了を表す．7 ビットのレセシブで表される．

2.2.2 アービトレーション

CAN プロトコルではバスアクセスに Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) 方式を採用している．そのため，フレームを送信しようとするノードは，まずバスがアイドル状態であることを確認し，他のノードがフレームを送信中の場合はフレームの送信が完了し，バスがアイドル状態になるまで待つ．

また，CAN ノードは自分が送信したバスレベルと実際のバスレベルを比較し，一致していることを 1 ビット毎に確認している．アービトレーションフィールドにおいて，バスレベルを比較した結果，自分が送信したバスレベルと実際のバスレベルが異なっていることを検出したノードはフレームの送信を停止する．このアービトレーション (調停) の仕組みによって，複数のノードが同時にフレームの送信を開始しても，優先順位が高いフレームを送信を中断することなく送信できる．

ノード 1 とノード 2 がそれぞれ $(123)_{16} = (00100100011)_2$ と $(321)_{16} = (01100100001)_2$ の CAN ID を持つフレームを同時に送信開始した場合を考える．ノード 1 とノード 2 は互いにバスがアイドル状態であることを確認し，SOF を表すドミナントを送信する．このとき，互いにドミナントを送信し，バスレベルがドミナントとなるため，自分が送信したバスレベルと実際のバスレベルは一致し，ノード 1 とノード 2 は次のアービトレーションフィールドの送信に移る．CAN ID が $(123)_{16}$ と $(321)_{16}$ の最上位ビットはどちらも 0 であるため，ノー

ド 1 とノード 2 はドミナントを送信し，送信レベルと実際のバスレベルの一致を確認し，次のビットの送信に移る．次のビットではノード 1 が 0 (ドミナント) を送信し，ノード 2 が 1 (レセシブ) を送信する．前述したように，ドミナントとレセシブが同時に送信された場合，バスのレベルはドミナントとなる．よって，ノード 2 は自分がレセシブを送信したにも関わらず，バスのレベルがドミナントであることを検出する．このとき，ノード 2 はアービトレーション負けの状態と呼び，送信権を失う．この結果，ノード 2 はフレームの送信を自ら停止する．一方，ノード 1 は自分がドミナントを送信し，バスのレベルがドミナントであるため送信権を維持し，送信を継続する．

また，同じ CAN ID でデータフレームとリモートフレームが送信された場合，データフレームの RTR がドミナントであることから，データフレームを送信しているノードが優先され，リモートフレームを送信しているノードがアービトレーション負けとなる．

2.2.3 同期

CAN プロトコルは，Non-Return to Zero (NRZ) 方式で信号を送受信するため，各ビットの始まりや終わりに同期信号を付加しない．また，ビットタイミングを伝送するための通信線も持たない．

CAN ノードは各ノードで独立したクロックを持ち，各ノードのクロック誤差や配線による信号の遅延でビットタイミングがずれることがある．そのため，各ノードではバスレベルがレセシブからドミナントへ変化するときに，ビットタイミングの同期を行う．

2.3 CAN に対する攻撃分析

CAN プロトコルが本質的に攻撃に対して脆弱であることは，多くの研究者が指摘している．Liu ら [18] は，CAN プロトコルが持つ本質的な脆弱性を 4 つの観点から分析し，CAN に対する攻撃手法を 5 種類に分類して分析している．本節では，CAN の脆弱性と CAN に対する攻撃手法について，それぞれ解説する．

2.3.1 CAN の脆弱性

CAN プロトコルが持つ本質的な脆弱性は，ブロードキャスト・無認証・平文通信・ID ベースの優先度の 4 つに分類できる．それぞれについて詳細を述べる．

ブロードキャスト

CAN フレームは，CAN バス上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．各 ECU は受信した CAN メッセージの ID (CAN ID) に基づき動作を決定する．CAN ID が自 ECU に関連する CAN ID であれば対応する動作をし，自 ECU に関係のない CAN ID であれば何もしない．このような仕組みのため，悪意のあるノードは他のノードから送信されたフレームを全て容易に傍受できる．

無認証

図 6 で示したように，CAN フレームにはメッセージの送信元を示すフィールドや認証のためのフィールドが規定されていない．そのため，受信ノードは悪意のあるノードから CAN ID を偽装して送信されたメッセージを区別できない．

平文通信

CAN フレームは，内容が暗号化されていないため，攻撃者は大量の CAN バスのログを容易に解析できる．

ID ベースの優先度

CAN ID は，メッセージの内容や送信ノードを示すためだけではなく，優先度も示す．先述のように，より小さい値の CAN ID はより高い優先度であることを示す．悪意のあるノードが最も高い優先度の CAN フレームを常に送信し続けると，他のノードでは，送信キューが待ち状態のままとなりフレームを送信できなくなる．このように，フレームの優先度が CAN ID で指定できることから，CAN プロトコルが本質的に DoS 攻撃に対して脆弱であることが分かる．

2.3.2 CAN に対する攻撃手法

CAN に対する攻撃手法はフレーム盗聴・フレーム偽装・フレーム挿入・リプレイ攻撃・DoS 攻撃の 5 種類に分類できる．それぞれの攻撃手法について詳細を述べる．

フレーム盗聴

CAN メッセージはバス上の全てのノードに送信されているため，盗聴が容易である．攻撃者は CAN バス上に流れているログの記録・解析により，フレームの詳細を知ることができる．

フレーム偽装

フレームの詳細を把握した攻撃者は，ECU を混乱させるための偽のフレームを作成し，バスに送信できる．例えば，車速情報を偽装したフレームを送信すれば，メーターに実際とは異なる速度を表示できる．これにより，ドライバーに誤った車速情報を認識させ，危険な運転へ誘導する攻撃が考えられる．

フレーム挿入

攻撃者は OBD-II コネクタに接続されたデバイスや，プログラムを書き換えた ECU またはマルウェアに感染したテレマティクスシステムを利用して偽装フレームを送信できる．

リプレイ攻撃

リプレイ攻撃は単純な攻撃手法である．CAN バスには送信ノードの認証メカニズムがないため，受信ノードはフレームの送信元を識別できない．リプレイ攻撃はドアドック解除・エンジン始動・ライト点灯などの攻撃に利用されうる．

DoS 攻撃

CAN バスは DoS 攻撃に対して仕様上脆弱である．CAN ID はフレームの優先度を表すため，攻撃者は最も高い優先度のフレームを連続して送信する

ことで，他のノードがメッセージを全く送信できない状態とすることが容易にできる．

2.4 車載ネットワークへの攻撃経路と対策

自動車の車載ネットワークは，直接または間接的に様々なインタフェースに接続されているため， 攻撃経路は多様である．そのため，車載ネットワークへの攻撃対策の考え方としては，多段的・多層的な防御策を施すことが重要とされている．本節では，階層の例として，田中らが示した車載システムを外部通信機器・車載ゲートウェイ・車載ネットワーク・ハードウェアの 4 階層に分類したモデルを紹介し，各階層におけるセキュリティ対策について述べる [19]．

2.4.1 車載ネットワークへの攻撃経路

車載ネットワークに対する攻撃経路の例を図 7 に示す．

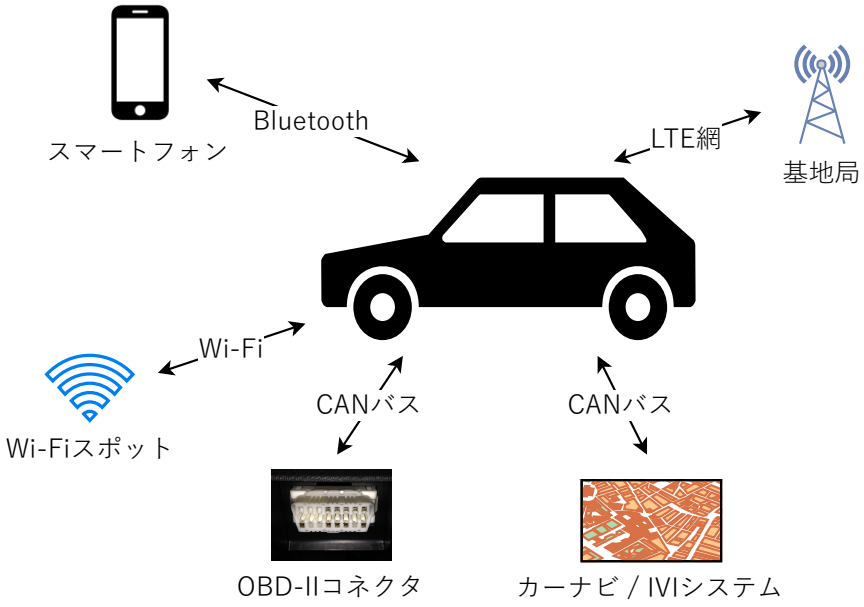

図 7 車載ネットワークへの攻撃経路

攻撃者がOBD-IIコネクタに物理的に接触できる場合は，直接車載ネットワークに侵入できる．特に，OBD-IIコネクタは運転席の分かりやすい位置に配置することが義務付けられているため[16]，攻撃者は悪意のあるデバイスを短時間で追加し，車両に細工をすることが容易である．

無線インタフェースにおいては，Wi-Fi・Bluetooth・LTE網のインタフェースを持つシステムが車載ネットワークに接続されておりかつ，そのシステムに脆弱性がある場合は車載ネットワークに侵入される可能性がある．車載システムの無線インタフェースから攻撃された実例として，セキュリティ研究者のHuntは2016年，電気自動車のエアコンを第三者が遠隔から制御できることを自身のブログで報告した[20]．電気自動車の純正アプリに認証機構が実装されていない脆弱性があり，第三者の車両のエアコンやファンを作動させるデモが行われた．この事例では，1章で述べたJeep Cherokeeの事例のように，車両の運転制御を乗っ取ることはできないものの，電気自動車内のバッテリーを消費させ，走行不能にさせられる可能性を示した．

また，車載ネットワークとカーナビやIVIシステムが接続されている場合，ファームウェアが不正なプログラムに書き換えられ，車載ネットワークに攻撃が加えられる可能性も考えられる．IVIシステムへの攻撃事例として，自動車メーカー純正のIVIシステムにスクリプトファイルの書き込まれたUSBメモリを差し込むと，無認証でスクリプトが実行される脆弱性が発見された[21]．この脆弱性を利用して，通常は走行中にテレビ視聴が不可能となっている設定を解除する方法が公開された．この脆弱性は，ファームウェアのアップデートにより対策された．

2.4.2 車載ネットワークへの攻撃対策

田中らは車載ネットワークにおけるセキュリティ対策を図8に示すような，外部通信機器・車載ゲートウェイ・車載ネットワーク・ハードウェアの4階層に分類して考え，各層において対策を施すことで，多段的・多層的な防御策を施すことが重要であると指摘した[19]．

階層1の外部通信機器は，外部と通信を行う役割を持ち，V2X (車車間・路者間通信) や，Wi-Fi/Bluetoothなどの近接無線機器，あるいはIVIシステムなどが

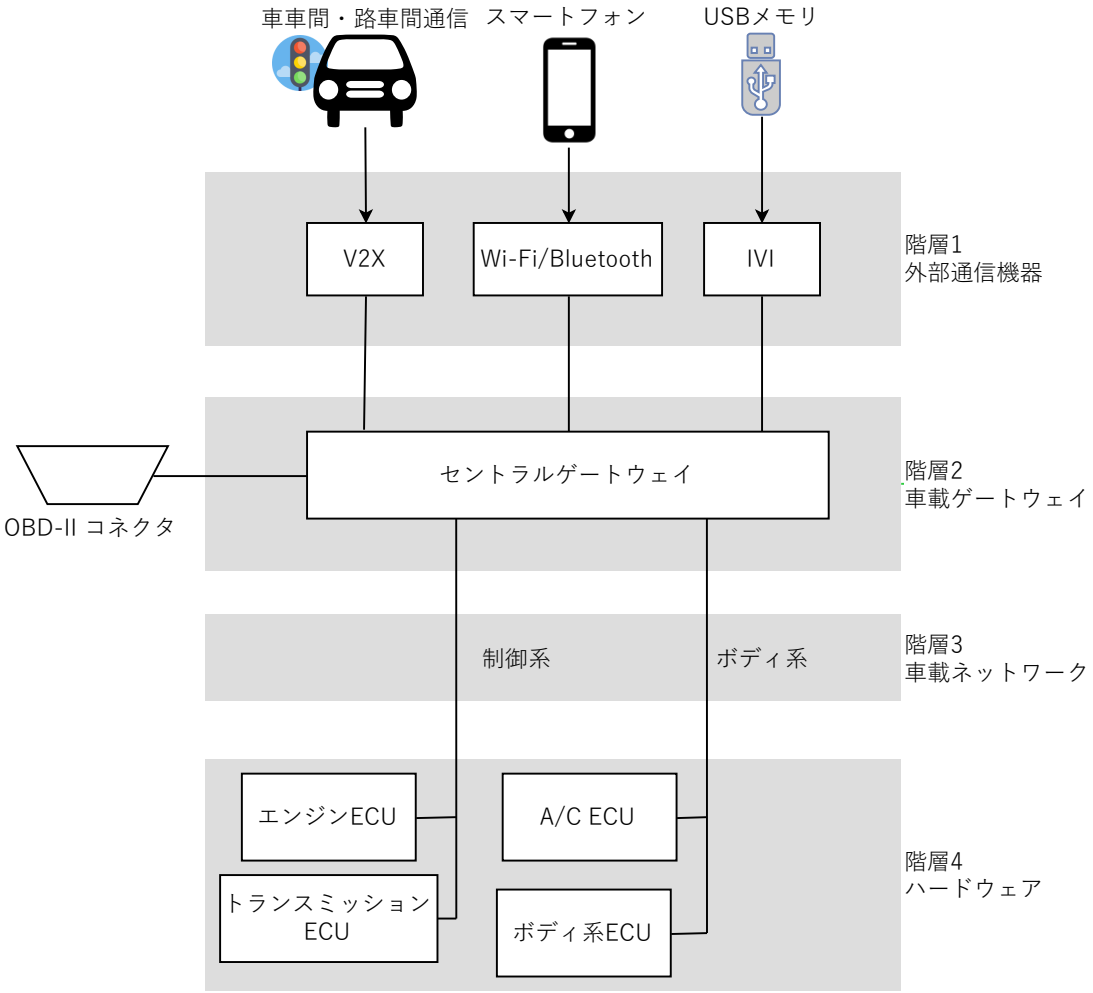

図 8　車載システムの階層分類 ([19] 図 1 を基に一部変更して作成)

該当する．階層 1 におけるセキュリティ対策として，通信路の暗号化による，盗聴や不正メッセージ挿入の防止が挙げられる．

階層 2 の車載ゲートウェイは，車載システム全体の制御を担う．階層 2 におけるセキュリティ対策として，車外システムと車載ネットワークとの間で許可されたメッセージのみを通過させるフィルタリングを行うことや，車載ネットワークに流れるメッセージに異常が発生していないかを監視・検知する異常検知をセントラルゲートウェイで行うことが挙げられる．

階層 3 は車載ネットワークは，ECU 間でやり取りされるメッセージを伝送する役割を担う．CAN や LIN などの異なる通信プロトコルが，用途によって使い分けられる．階層 3 におけるセキュリティ対策として，暗号技術を用いたメッセージ認証符号 (Message Authentication Code; MAC) を付与する方法がある．MAC を付与する方法の具体例として，Secure Onboard Communication (SecOC) と呼ばれる仕様が，車載ソフトウェアの標準規格である AUTOSAR Release 4.2.1 で言及されている [22]．MAC を付与する対策手法は，暗号技術に基づいてメッセージの正当性を検証できるため，確実に不正送信を防ぐことができ，非常に有効である．しかし，送受信双方の ECU を認証機能に対応させる必要があるため，既存の車への適用が難しい問題がある．

階層 4 のハードウェアは，車載ネットワークに接続される ECU が該当する．階層 4 におけるセキュリティ対策として，ECU のプログラムが改ざんされていないことを起動時に検証するセキュアブートが挙げられる．

3. 関連研究

本章では，CAN バスへの攻撃対策の関連研究を Intrusion Detection System (IDS) と CAN バスの物理的特徴に着目した送信元識別手法の 2 つの視点から分類して，それぞれ 3.1 節と 3.2 節で概観する．

3.1 Intrusion Detection System (IDS)

CAN バス上でやり取りされるメッセージの送信周期やメッセージの内容を分析することで，異常を検知する侵入検知システム (Intrusion Detection System; IDS) の研究は過去に多く行われてきた．

Song ら [23] は，CAN メッセージの多くは周期的に送信されることに着目し，特定の CAN ID に対して，フレーム挿入攻撃が行われた際にメッセージの受信間隔が，明らかに短くなることを指摘した．これに基づいた IDS を実装し，閾値を適切に設定することでほぼ 100%の検知精度で攻撃を検知できることを示した．芳賀らは，CAN メッセージの周期性に着目し，外れ値検知アルゴリズムの Local Outlier Factor (LOF) を用いて，異常スコアを算出し，異常を検知する方法を提案した [24]．

車載ネットワーク向けの IDS は既にいくつか実用化されている．シマンテックは「Anomaly Detection for Automotive」と呼ばれるセキュリティソリューションを提供している [25]．機械学習を用いて，正常時の CAN バス上の特性を把握し，正常な範囲を超えると異常状態として検知する．パナソニック株式会社のオートモーティブ&インダストリアルシステムズ社は，機械学習を用いて車載ネットワークへの攻撃をリアルタイムに検知できる侵入検知・防御システムを開発した [26]．車載器から収集したログをクラウドに集約し，監視ルールを更新することで新しい攻撃にも対応できる．また，CAN に加えて車載 Ethernet にも対応している．

これらの先行研究に共通する問題として，通常時においてメッセージの送信間隔が周期的ではない場合には適用できないことが挙げられる．

3.2 送信元ノード識別手法

CAN バスにおいて，メッセージの送信元ノードを識別する研究は過去にいくつかなされてきた．本節では，信号の物理的特徴を用いて送信元ノードを識別する手法と，周期的なメッセージにおける Clock skew を用いて送信元ノードを識別する手法について述べる．前者は信号の計測を行うハードウェアが必要であるが，後者は通常の CAN コントローラのみで実現できる特徴がある．

3.2.1 信号の物理的特徴を用いる手法

信号の物理的な特徴とは，具体的には電圧や信号立ち上がり時間・立ち下がり時間，あるいは信号の安定度などが挙げられる．

CAN バスにおいて，信号の物理的な特徴を用いてメッセージの送信元ノードを識別する手法は，Murvay らによって最初に示された [11]．2 GSa/s[7]のオシロスコープを用いて電圧を測定し，平均二乗誤差と畳み込みの 2 種類の特徴量を計算した．それぞれについて送信元ノードの識別精度の評価を行った．

Choi らは，[12]．CAN フレームの拡張識別子フィールドに固定のビット列を埋め込み，埋め込んだビット列の部分の電圧値を 2.5 GSa/s のオシロスコープを用いて測定した．時間ドメインの特徴量として平均値・標準偏差・平均偏差・歪度・尖度など，8 つの統計量を定義し，周波数ドメインの特徴量として標準偏差・歪度・尖度・ロールオフなど，9 つの統計量を定義し，合計 17 つの特徴量とした．SVM・ニューラルネットワーク・二分決定木の 3 つのアルゴリズムで識別精度を比較した．結果として，送信元 ECU を Murvay らの手法よりも高精度に識別できることを示した．この手法は，拡張識別子フィールドを利用する．標準フォーマットの CAN バスでは，拡張フォーマットの CAN フレームの送受信は不可能であるため，この手法を適用するには，CAN バス上の全てのノードを拡張フォーマットに対応させる必要があるという問題がある．

Cho らは送信メッセージの電圧の違いから攻撃ノードを検知する手法を提案した [13]．Choi らの手法と異なり，既存の CAN メッセージのフォーマットを変

[7]Samples/second の略．

更する必要がないという特徴がある．しかし，計測ノードの電圧サンプル能力が 50 kSa/s であるため，プロファイルの更新に 2-3 メッセージ計測する必要がある．

信号の物理的特徴を用いた IDS は，既に実用化されている．例として，NNG の開発した不正送信阻止システム (Parallel Intrusion Prevention System; PIPS) が挙げられる [27]．これは，CAN バス上の波形を監視し，リアルタイムで攻撃を阻止できる IDS である．

3.2.2 Clock skew を用いる手法

別のアプローチとして，Cho らは周期的に送信される CAN メッセージから，Clock skew と呼ばれる指標に基づいて ECU を識別する手法を提案した [10]．

この手法は，CAN のログデータから Clock skew を抽出する手法であるため特殊なハードウェアを必要としないという利点がある．しかし，周期的ではないメッセージにはこの手法は適用できないという欠点がある．また，Clock skew を利用する IDS に対する攻撃手法として，Sagong らは Clock skew を意図的に偽装し，IDS による検知を回避できることを実験的に示した [28]．

3.3 関連研究のまとめ

CAN メッセージの様々な特徴に着目した IDS が多く提案されているが，CAN メッセージの周期性に着目した手法が多い．多くの IDS に共通する問題点として，通常時においてメッセージの送信間隔が周期的ではない場合には適用できないことが挙げられる．

Choi らの示した CAN バスの物理的特徴に着目した送信元識別する手法 [12] は，Murvay らの示した手法 [11] よりも高い精度でメッセージの送信元ノードを識別できることを示したが，拡張識別子フィールドを利用するため，既存の通信手順を変更する必要があり，実用性の点で問題がある．

Cho らの示した Clock skew を用いて送信元ノードを識別する手法は [10]，特別なハードウェアを必要としないという利点があるものの，周期的ではないメッ

セージには適用できないという問題がある．また，Clock skew を偽装する攻撃手法が Sagong らによって示されている [28]．

4 章で提案する手法では，CAN メッセージの送信周期や内容に依存せず送信元ノードを識別でき，既存の CAN バスにおける通信手順を変更する必要がない．また，A/D 変換器を必要としないという特徴がある．

4. 遅延時間を利用した攻撃ノード識別手法の提案

本研究では，CAN トランシーバの遅延時間が立ち上がり時と立ち下がり時で異なり，また，その遅延時間の差が CAN トランシーバ毎に異なる点に着目して，CAN メッセージの送信元を識別する方法を提案する．

まず，本章に関連する用語を以下のように定義する．

遅延時間 (propagation delay time, t_d)

入力が遷移してハイレベルとローレベルの中間 (50 %) となる時刻から，出力が遷移してハイレベルとローレベルの中間 (50 %) となるまでの時刻までの時間差．特に立ち下がり時の遅延時間を t_{pdHL}，立ち上がり時の遅延時間を t_{phLH} と定義する．

ビット時間 (bit time, t_{bit})

CAN バスのビットレート f_{bit} の逆数．$t_{bit} = \frac{1}{f_{bit}}$ である．

遅延時間 t_{pdHL}, t_{pdLH} の定義を図示したものを図 9 に示す．

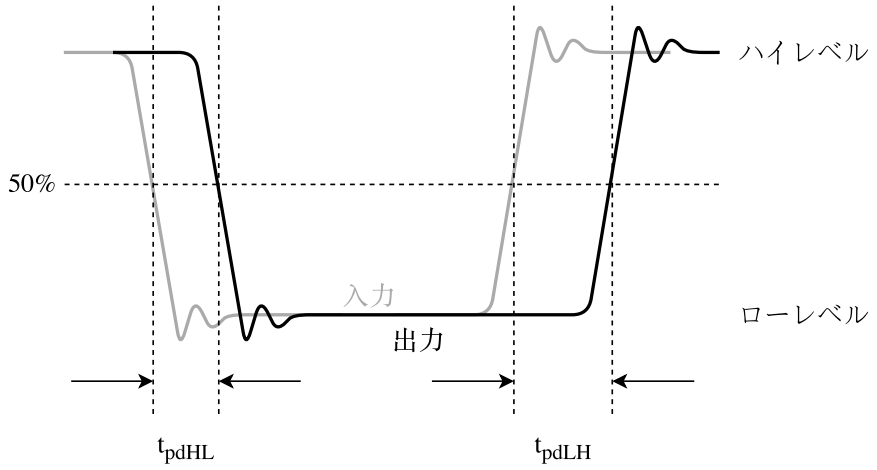

図 9 遅延時間 t_{pdHL}, t_{pdLH} の定義

4.1 CAN トランシーバの送信回路

典型的な CAN トランシーバにおける送信部の等価回路を図 10 に示す [29]．上段は保護ダイオード (D1)，P チャネル FET[8] (Q1) からなり，下段は保護ダイオード (D2)，N チャネル FET (Q2) からなる．上段のダイオード (D1) は，CAN-H ピンの電圧が V_{dd} よりも高くなったときに，V_{dd} への電流の逆流を防止する．下段のダイオード (D2) は，CAN-L ピンの電位がグランドよりも低くなったときに，グランドから CAN-L ピンへの電流の逆流を防止する．R_L はバスの負荷抵抗を表す．2.2 節で述べたように，高速 CAN では 60 Ω である．また，CAN-L と CAN-H には約 2.5 V のバイアス電圧がかけられている．

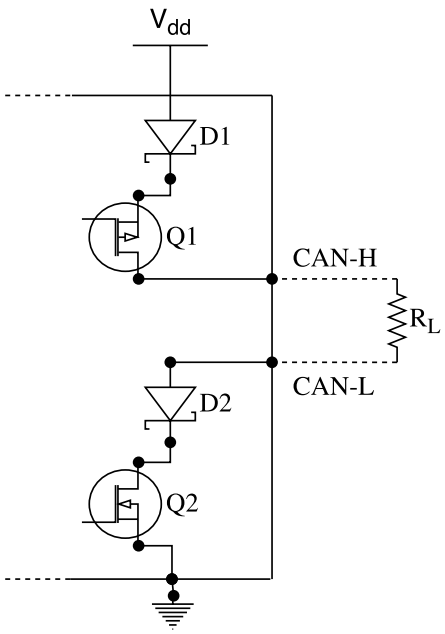

図 10 CAN トランシーバにおける出力部の等価回路

2.2 節で述べたように，CAN バスは，2 つのバスレベルがあり，V_{diff} が 1.5 V から 3.0 V の状態をドミナント，V_{diff} が −0.5 V から 0.05 V の状態をレセシブという．

Q1 のゲートにハイレベル，Q2 のゲートにローレベルが入力されているとき，

[8]Field Effect Transistor; 電界効果トランジスタ.

2 つのトランジスタ Q1, Q2 はオフ状態となる．このとき，トランジスタはハイ・インピーダンス状態であり，V_{dd} からグランドに流れる電流は無視できるほど小さい．よって，R_L における両端の電位差 V_{diff} はほぼ 0 V となり，バスはレセシブとなる．一方，Q1 のゲートにローレベル，Q2 のゲートにハイレベルが入力されているとき，2 つのトランジスタ Q1, Q2 はオン状態となる．このとき，2 つのトランジスタはロー・インピーダンス状態であり，V_{dd} から R_L を経由してグランドに電流が流れる．よって，R_L の両端の電位差 V_{diff} は約 2.0 V となり，バスはドミナントとなる．

CAN バスの信号レベルが変化するときの遅延時間 t_d は，トランジスタのスイッチング時間と出力 C_L の負荷容量が充放電される時間によって決まる [30]．また，一般に立ち上がり時間と立ち下がり時間が長くなればなるほど遅延時間は増加する．負荷容量 C_L の要因として，トランジスタにおけるゲートの出力容量，ゲートの入力容量，配線容量の 3 種類が挙げられる．

FET のスイッチング時間は，一般に立ち上がり時と立ち下がり時で異なり，データシートには区別して記載される [31]．このことから，CAN トランシーバにおける遅延時間は立ち下がり時と立ち上がり時で異なることが分かる．

CAN バスにおける，CAN トランシーバのタイミング図を図 11 に示す．

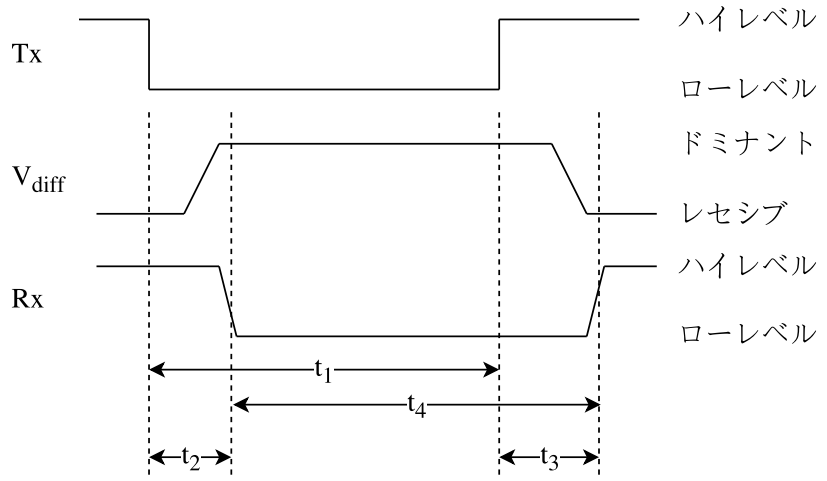

図 11 CAN トランシーバのタイミング図

前述の要因により，送信ノードの CAN トランシーバの Tx ピンが変化してか

ら，差動電圧 (V_{diff}) が発生し，受信ノードの CAN トランシーバの Rx ピンが変化するまでには遅延時間が存在する．送信ノードの送信するビット幅を t_1，送信 CAN コントローラの Tx ピンの電圧レベルが立ち下がりしてから受信ノードの CAN トランシーバの電圧レベルが立ち下がるまでの遅延時間を t_2，同様に Tx ピンの電圧レベルが立ち上がりしてから受信側で立ち上がるまでの遅延時間を t_3，受信ノードの CAN トランシーバで計測されるビット幅を t_4 とする．受信ノードで計測される t_4 は

$$t_4 = t_1 + t_3 - t_2 \tag{2}$$

となり，$t_3 - t_2$ が送信ノードの送信したパルス幅 t_1 との差分として現れる．

ここで，受信ノードで観測できる情報は t_4 のみであり，t_1, t_2, t_3 は未知である．そこで，t_1 と t_{bit} の関係について考える．CAN バスでは，ノードごとに独立したクロックを持ち，2.2 節で述べたような仕組みで同期を取りながら通信する．よって，t_e を水晶発振器の誤差とすると，

$$t_1 = t_{bit} + t_e \tag{3}$$

と表される．高速 CAN ノードでは，周波数許容偏差 (frequency tolerance) の要求から水晶発振器が使用されている．典型的な水晶発振器は $\pm 30\,\mathrm{ppm} - \pm 100\,\mathrm{ppm}$ 程度の周波数許容偏差を持つ [32]．周波数許容偏差を f_{tol} [ppm] とすると，周波数偏差 Δf_e および水晶発振器の誤差 t_e は，

$$\Delta f_e = f_{bit} \times f_{tol} \times 10^6 \tag{4}$$

$$t_e = \frac{1}{f_{bit} + \Delta f_e} - t_{bit} \tag{5}$$

と表される．

例えば，バスの速度が 500 kbps で周波数偏差が大きい場合を考え，$f_{bit} = 500\,\mathrm{kHz}, f_{tol} = +100\,\mathrm{ppm}$ とすると，$t_{bit} = \frac{1}{f_{bit}}$ であるので，

$$\Delta f_e = (500 \times 10^3) \times (100 \times 10^{-6}) \;=\; 50\,\mathrm{Hz} \tag{6}$$

$$t_e = \frac{1}{500 \times 10^3 + 50} - \frac{1}{500 \times 10^3} \;\approx\; -1.9998 \times 10^{-10}\,\mathrm{s} \tag{7}$$

となる．$-1.9998 \times 10^{-10}\,\mathrm{s} = -0.199\,98\,\mathrm{ns}$ であり，$t_3 - t_2$ は $\pm 80\,\mathrm{ns}$ 程度である．このことから，t_e は $t_3 - t_2$ に比べて十分小さいことが分かる．よって，$t_1 = t_{bit}$ とみなすと，$t_3 - t_2$ は，

$$t_3 - t_2 \approx t_4 - t_{bit} \tag{8}$$

と近似できる．

4.2 CAN トランシーバにおける遅延時間の実測

予備実験として，CAN ノード 1・CAN ノード 2 の 2 つのノードを同じ CAN バス上に配置し，図 12 のようにオシロスコープ (Keysight InfiniiVision 3054A) で計測した．プローブの長さによる伝搬遅延の影響を最小限にするために，同じプローブ (Keysight N2890A) を 2 本使用した．計測に用いたオシロスコープの性能を表 2 に示す．オシロスコープの Ch. 1(橙線) は CAN ノード 1 の CAN トランシーバの送信ピン (Tx) に，Ch. 2(緑線) は CAN ノード 2 の CAN トランシーバの受信ピン (Rx) に接続した．CAN ノード 1 は CAN メッセージを全く送信していない状態，CAN ノード 2 は一定周期で CAN メッセージを送信している状態とした．この状態では，CAN ノード 1 から送信される信号は CAN ノード 2 が送信した CAN フレームに対する ACK のみとなるため，立ち下がりでトリガをかけることで，容易に波形を捉えられる．

表 2 計測に用いたオシロスコープの性能

最高サンプリングレート	4 GSa/s (チャネルあたり 2 GSa/s)
帯域幅	500 MHz
計算された立ち上がり時間 (10–90 %)	$\leq 700\,\mathrm{ps}$

計測条件でのサンプリングレートが 2 GSa/s であるため，時間分解能は

$$\frac{10^{12}}{2.0 \cdot 10^{9}} = 500\,\mathrm{ps} \tag{9}$$

となる．計測した波形の例を，図 13 に示す．図の波形では，Tx ピンの立ち下がりから Rx ピンの立ち下がりまでの遅延時間 t_{pdHL} は 82.0 ns，Tx ピンの立ち上が

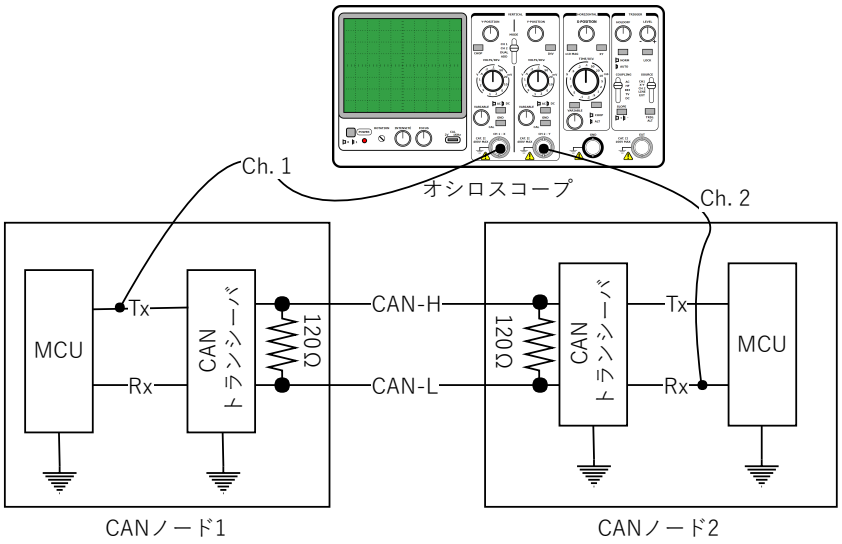

図 12　CAN トランシーバの遅延を計測するためのプローブポイント

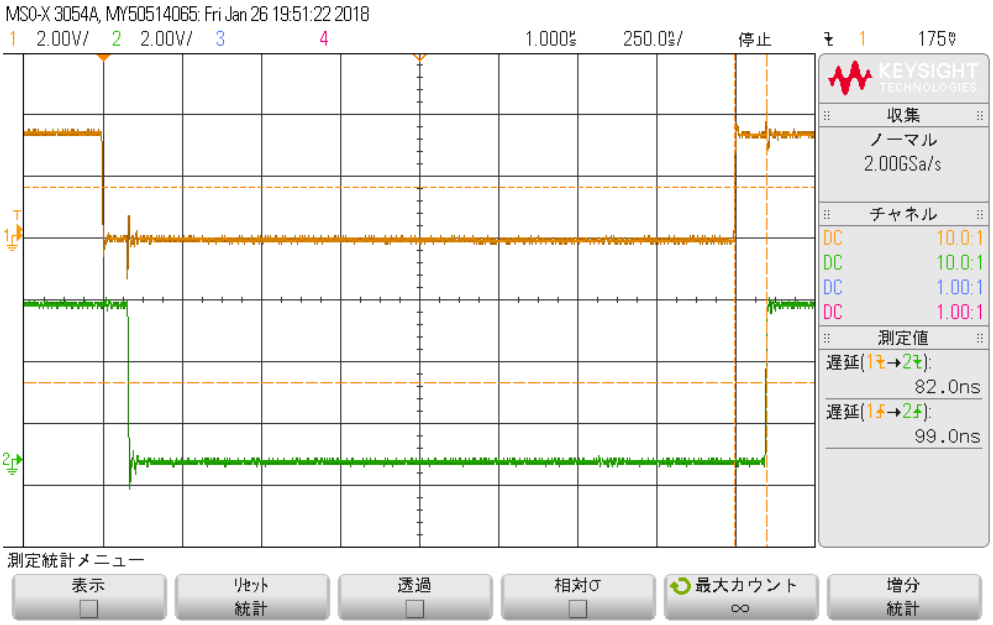

図 13　CAN トランシーバの遅延をオシロスコープで計測した波形の例

りから Rx ピンの立ち上がりまでの遅延時間 t_{phLH} は 99.0 ns である．このことから，立ち下がり時と立ち上がり時の遅延時間の差は

$$t_{phLH} - t_{phHL} = 99.0 - 82.0 = 17.0\,\mathrm{ns} \tag{10}$$

となることが分かる．

同じ条件で，3.0 時間 (92 662 サンプル) 計測したグラフを図 14 に，0–10 秒間 (86 サンプル) を拡大したグラフを図 15 にそれぞれ示す．実験室は空調管理されており，気温は 24.5 °C だった．最小値は 15.0 ns，最大値は 19.5 ns となった．

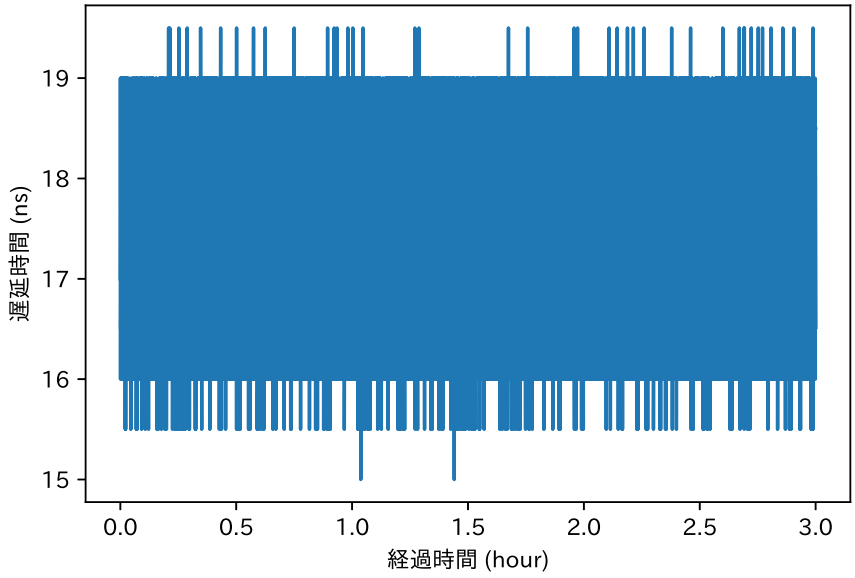

図 14　遅延時間を 3 時間計測したグラフ

図 14 のグラフは短期間の雑音成分 (図 15) を含み，密度が非常に高いため，長時間での変化が捉えにくい．そこで，移動平均フィルタ [33] を適用して平滑化した．

入力信号 $x(n)$ に対して，M_m 点 (M_m は奇数) の移動平均処理を行ったとき，出力信号 $y(n)$ は次の式で表される．

$$y(n) = \frac{1}{M_m} \sum_{l=-L}^{l=L} x(n+l) \tag{11}$$

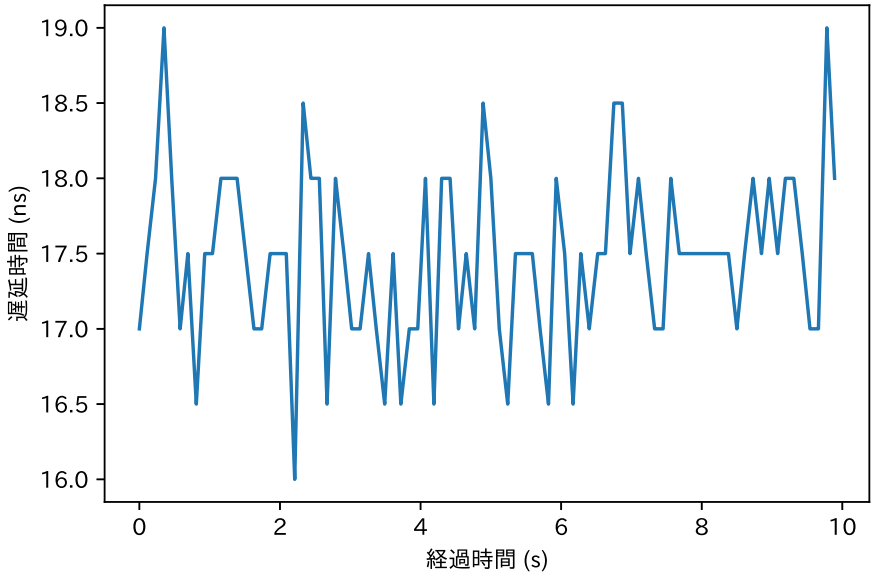

図 15　図 14 の 0–10 秒間を拡大したグラフ

ここで，$L = (M_m - 1)/2$ である．この式より，移動平均は $x(n)$ を中心に $x(n-L)$ から $x(n+L)$ までの合計 M_m 個のサンプルの平均値を求める処理であることが分かる．

$M_m = 701$ として，図 14 に移動平均フィルタを適用したグラフを図 16 に示す．最小値は 17.35 ns，最大値は 17.57 ns だった．遅延時間は短期的な雑音成分を含み (図 14)，長期的にも不規則な変動がある (図 16) ことが分かる．

4.3　計測結果の正規性検定

前節で 3 時間計測した結果のヒストグラムを図 17 に示す．図の分布より，遅延時間の分布は正規分布である考えられる．これを確かめるため，χ^2 分布を利用した適合度の検定 [34] を行い，計測結果の正規性を調べた．

実測による分布および仮説による分布ともに度数分布で表されており，第 k クラスの度数を f_k および $f_k^*(k = 1, 2, \cdots, m)$ とする．このとき，仮説が正しけれ

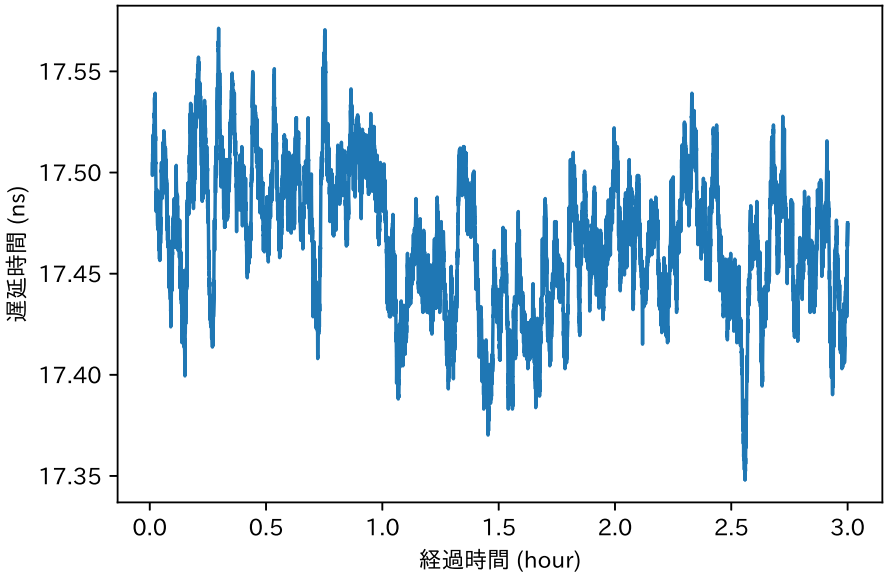

図 16　図 14 に移動平均フィルタ $(M_m = 701)$ を適用したグラフ

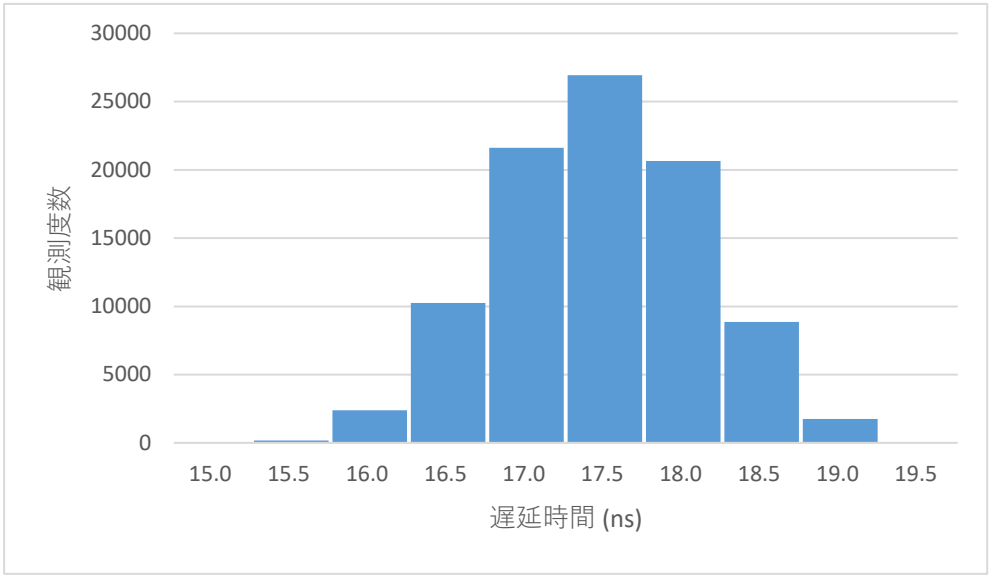

図 17　3 時間計測した結果のヒストグラム

ば，

$$\chi^2 = \sum_{k=1}^{m} \frac{(f_k - f_k^*)^2}{f_k^*} \tag{12}$$

は自由度が $m-1$ の χ^2 分布をすることが知られている．

表 3　3 時間の正規性検定で用いたデータ

k	遅延時間 (ns)	f_{k0}	f_{k0}^*
1	≤ 14.5	0	1.54
2	15.0	2	31.27
3	15.5	166	371.55
4	16.0	2382	2517.79
5	16.5	10254	9753.41
6	17.0	21624	21639.62
7	17.5	26932	27531.35
8	18.0	20653	20093.60
9	18.5	8867	8408.56
10	19.0	1742	2014.87
11	19.5	40	275.92
12	20.0	0	21.54
13	≥ 20.5	0	0.98
	合計	92662	92662

まず，前節で示した 3 時間の計測結果の正規性検定を行う．帰無仮説 H_0 を「3 時間の計測結果は正規分布である.」，対立仮説 H_1 を「3 時間の計測結果は正規分布でない.」とする．有意水準は 5%とする．3 時間の正規性検定で用いたデータを表 3 に示す．f_{k0} は観測度数，f_{k0}^* は平均 17.47 ns・標準偏差 0.65 ns の正規分布を仮定したときの期待度数となっている．よって，統計量 χ_0^2 は

$$\chi_0^2 = \sum_{k=1}^{13} \frac{(f_{k0} - f_{k0}^*)^2}{f_{k0}^*} = 490.50 \tag{13}$$

となる．自由度 $13-1=12$ の χ^2 分布で，p 値は $2.33 \times 10^{-97} < 0.05$ となり，帰

無仮説 H_0 は棄却される．すなわち，3 時間の計測結果は正規分布でないといえる．正規分布でないという結論となったことは，長期的な変動の影響により，遅延時間の平均値が変化したためと考えられる．

変動が起こる前のより短い区間であれば，雑音成分の正規性を確認できると考えられる．そこで，より短い 200 秒間 (1713 サンプル) でも，同様に正規性の検定を行った．

表 4　200 秒間の正規性検定で用いたデータ

k	遅延時間 (ns)	f_{k1}	f^*_{k1}
1	≤ 15.0	0	0.43
2	15.5	3	5.46
3	16.0	42	40.23
4	16.5	169	166.93
5	17.0	384	390.68
6	17.5	505	516.44
7	18.0	411	385.77
8	18.5	167	162.75
9	19.0	32	38.73
10	19.5	0	5.19
11	≥ 20.0	0	0.41
	合計	1713	1713

帰無仮説 H_0 を「200 秒間の計測結果は正規分布である.」，対立仮説 H_1 を「200 秒間の計測結果は正規分布でない.」とする．有意水準は 5%とする．200 秒間の正規性検定で用いたデータを表 4 に示す．f_{k1} は観測度数，f^*_{k1} は平均 17.49 ns・標準偏差 0.65 ns の正規分布を仮定したときの期待度数となっている．よって，統計量 χ^2_1 は

$$\chi^2_1 = \sum_{k=1}^{11} \frac{(f_{k1} - f^*_{k1})^2}{f^*_{k1}} = 10.53 \tag{14}$$

となる．自由度 $11 - 1 = 10$ の χ^2 分布で，p 値は $0.40 > 0.05$ となり，帰無仮説

H_0 は棄却できない．すなわち，200 秒間の計測結果は正規分布でないとは言い切れない．

5. 提案手法のプロトタイプの試作

4 章で述べた，遅延時間を計測するプロトタイプデバイスの実装を行った．本章では，遅延時間計測の実装方法について説明する．

5.1 デバイスの設計

実装に用いたハードウェアの概要を表 5 に示す．MCU には mbed LPC1768 を使用した．CPU のクロック周波数は最高速度である 100 MHz に設定した．また，MCU 内蔵 CAN インタフェースの速度は 500 kbps に設定した．よって，$t_{bit} = \frac{10^9}{500 \times 10^3} = 2000$ ns である．

試作したプロトタイプの外観を図 18 に示す．

表 5 プロトタイプハードウェアの概要

MCU	mbed LPC1768
CPU コア	ARM Cortex-M3
クロック周波数	100 MHz
性能	125 DMIPS
フラッシュ	512 kB
RAM	32 kB
CAN インタフェース数	2

5.2 遅延時間計測の実装

試作したプロトタイプのアーキテクチャを図 19 に示す．CAN トランシーバの Tx ピン・Rx ピンは，それぞれ MCU に内蔵された CAN コントローラの Tx ピン・Rx ピンに接続した．Rx ピンは，タイマのキャプチャピンにも並列に接続している．

32 ビットタイマのクロックは CPU と同じクロックを分周せずに入力し，100 MHz とした．よって，カウンタ値は $\frac{10^9}{100 \times 10^6} = 10$ ns 単位である．MCU に内蔵の 32 ビッ

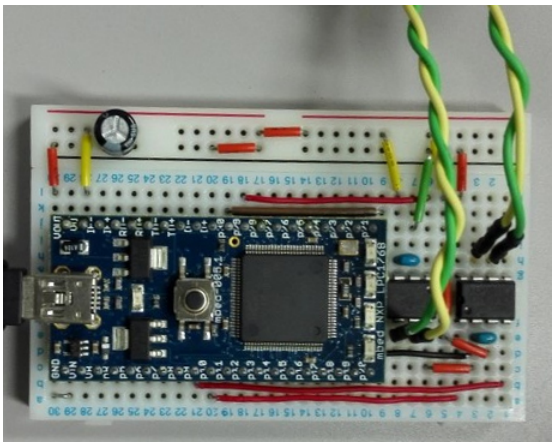

図 18 試作した提案手法のプロトタイプ

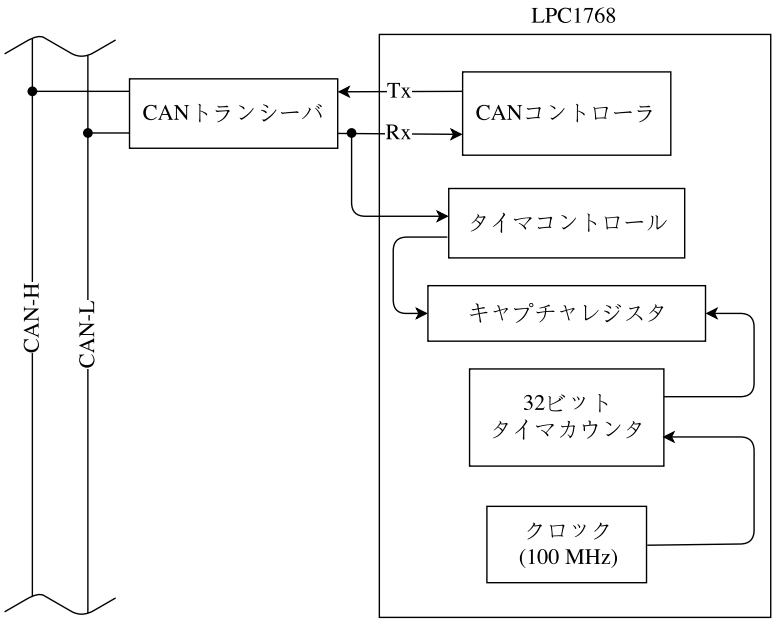

図 19 試作したプロトタイプのアーキテクチャ

トタイマのインプットキャプチャ機能を利用して，CAN トランシーバの Rx ピン変化エッジでのカウンタ値を正確に捉えられる．

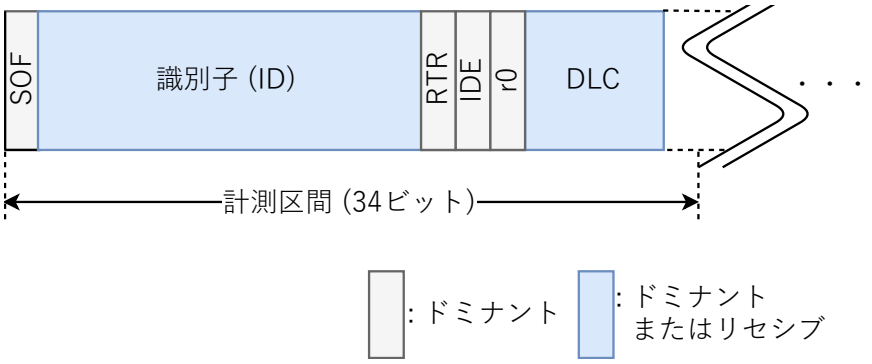

図 20　プロトタイプにおける 1 フレームあたりの計測区間

プロトタイプにおける 1 フレームあたりの計測区間を図 20 に示す．計測区間中は立ち上がりの度にキャプチャされ，キャプチャレジスタに計測値が格納される．2.2 節で述べたように，CAN フレームのデータフィールドは可変長であり，その長さはコントロールフィールドの DLC で示される．CAN コントローラからは，フレームの受信が完了するまで，その CAN メッセージの内容は取得できない，また，DLC をソフトウェア処理でデコードすることは，処理時間の観点から困難である．そのため，CAN フレームの長さが最短 (DLC=0) であっても，確実に送信ノードが送信している区間を計測できるようにする必要がある．そこで，DLC=0 の CAN フレームを考えると，ACK フィールドまでに，SOF (1 ビット)・アービトレーション (12 ビット)・コントロールフィールド (6 ビット)・CRC フィールド (16 ビット) の合計 35 ビット 分の信号が送信されることが分かる．CRC デリミタを含めると ACK スロットの立ち上がりを計測してしまう可能性があるため，1 ビット 引き，SOF から 34 ビット分の時間 (68 µs) を計測区間とする．1 フレームの長さが DLC=0 の時の CAN フレームよりも短くなることはないため，この設定により，確実に送信ノードの信号のみを計測できる．

計測したカウンタ値から遅延時間 t_{delay} を求める方法を述べる．タイマのカウンタ値は 10 ns 単位であるので，SOF からの経過時間 $t_{elapsed}$ (ns) は次のように計

算する．

$$t_{elapsed} = (\text{キャプチャカウンタ値} - \text{SOF のカウンタ値}) \times 10 \tag{15}$$

計測したタイミングでの SOF からの経過ビット数は，$t_{elapsed}$ を 1000 ns 単位で丸め，$t_{bit} = 2000\,\text{ns}$ で割ることで，次のように計算できる．

$$\lfloor \frac{t_{elapsed} + 500}{2000} \rfloor \tag{16}$$

よって，SOF からの理想経過時間 t_{ideal} (ns) は，SOF からの経過ビット数に $t_{bit} = 2000\,\text{ns}$ を掛けることで，次のように計算できる．

$$t_{ideal} = \lfloor \frac{t_{elapsed} + 500}{2000} \rfloor \times 2000 \tag{17}$$

求める t_{delay} (ns) は，

$$t_{delay} = t_{elapsed} - t_{ideal} \tag{18}$$

$$= t_{elapsed} - \lfloor \frac{t_{elapsed}+500}{2000} \rfloor \times 2000 \tag{19}$$

となる．

計測プログラムの動作を述べる．

1. SOF の立ち下がりエッジでキャプチャされ，キャプチャ割り込みが起こる．ここでキャプチャレジスタに格納されている値を SOF の基準とする．立ち上がりエッジでキャプチャするようにタイマを設定する．その後，68 µs 後にタイマ割り込みを設定する．

2. 立ち上がりエッジ毎にキャプチャ割り込みが起こる．式 19 を用いて遅延時間を計算し，記録する．

3. SOF から 68 µs 経過すると，タイマ割り込みが起こり，計測を終了する．

4. CAN フレームの受信が完了すると，CAN コントローラから受信完了割り込みが発生する．ここで初めて受信したフレームの CAN ID が分かるので，CAN ID と計測結果を共に出力する．

5.3 制約

本プロトタイプ実装の制約として，すべての CAN メッセージを処理できないことが挙げられる．これは，フレームを受信し終えてから計測結果を出力し終わるまでは，新しい CAN フレームの送信が開始されても処理できないデッドタイムとなるためである．これは，ソフトウェア処理で実装していることに起因する．FPGA などを用いてハードウェア実装とするか，より高いクロック周波数で動作できる MCU を用いることで解決できると考えられる．

6. 評価実験

本章では，提案手法の有効性評価を目的として行った実験の結果を述べる．ECU 分類手法として，最も単純な分類アルゴリズムである k-Nearest Neighbor (kNN) アルゴリズムを用いた．kNN アルゴリズムにおいて $k = 5$ とした．まず，研究室内で構築したテストベンチおよび実車に対して，試作したプロトタイプデバイスを用いて遅延時間計測の実験を行った結果について述べる．次に，実験結果に関する考察を行い，最後に今後の課題について述べる．

6.1 テストベンチでの実験

研究室内で CAN ノードを 6 個配置して評価実験を行った．実験で構築したテストベンチでは，23 個の異なる CAN ID のメッセージが送信されている．CAN バスの速度は 500 kbps とした．提案手法の有効性の評価と時間に対して，計測結果がどのように変化するかを調べた．

CAN メッセージ毎に計測結果の平均値と標準偏差を計算した．計測は 60 秒間行い，7205 の CAN メッセージを観測した．横軸を遅延時間 (ns)・縦軸を標準偏差 (ns) として，ノード別に色分けして，CAN メッセージ毎にプロットしたグラフを図 21 に示す．ELM327 2 とコンビネーションメーター 2 間および，ELM327 2 と PiCAN2 間で点の重なりが見られるものの，ノード毎に遅延時間の差が明確に出ており，CAN ノードを識別するために必要な分解能があることが分かる．

標準偏差のコンビネーションメーター 1・実車エンジン ECU 以外の点は，5.0 ns 付近に分布しており，標準偏差のみを用いてノードを分類することは難しいと考えられる．よって，識別に必要な要素としては，横軸の遅延時間が大きいと言える．

kNN アルゴリズムによる分類を行った結果の混同行列を図 22 に示す．ラベルの番号は 1 から 6 まで順に図 21 における，ELM327 1，ELM327 2，PiCAN2，コンビネーションメーター 1，実車エンジン ECU，コンビネーションメーター 2 に対応している．10 分割交差検証によってモデルの評価を行った結果，平均正解率は 0.9738 となった．

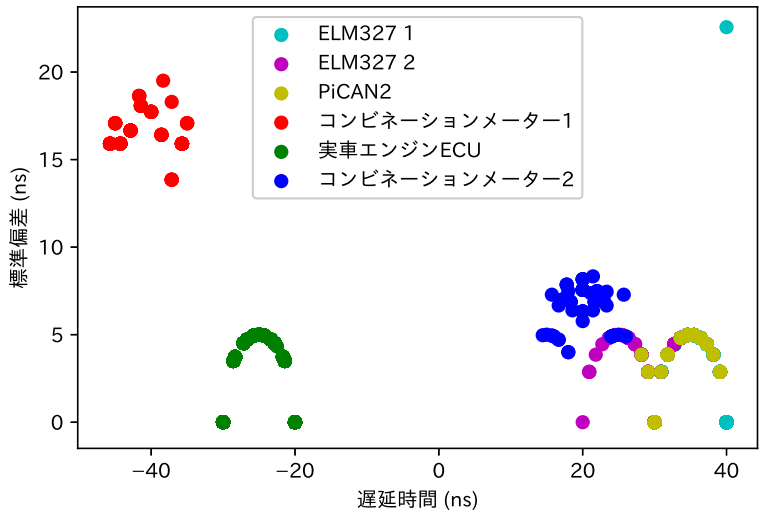

図 21　6 個の CAN ノードを配置したテストベンチでの散布図 (12 月 22 日)

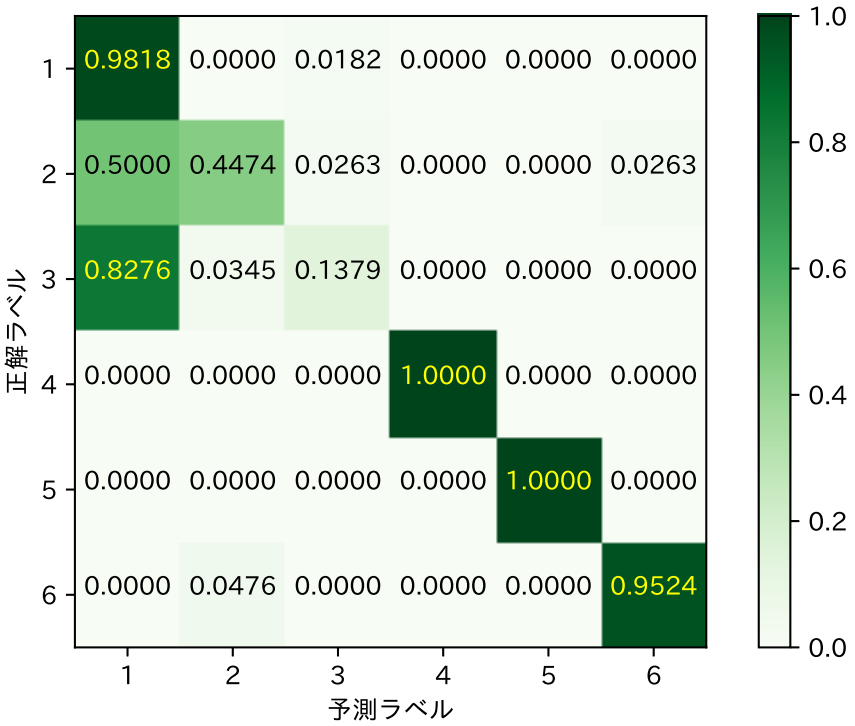

図 22　テストベンチでの kNN アルゴリズムによる分類結果

6.2 実車を用いた実験

提案手法が，実車においても有効かどうか検証・評価を行った．

2007 年式の乗用車 1 で実験を行った．CAN バスへの接続は OBD-II コネクタを利用した．実験はすべて，エンジンをかけて車両を静止した状態で行った．

6.2.1 CAN ID と送信元 ECU の対応付け

CAN プロトコルでは，同じ CAN ID を持つメッセージを複数の ECU から送信することを禁止している．そのため，各 CAN ID の送信元 ECU は 1 つである．しかし，どの ECU がどの CAN ID のメッセージを送信するかどうかはメーカーや車種によって異なり，一般にその情報は公開されていない．

そこで，車両の電気配線図集や CAN バスのログデータの内容から送信 ECU を手動で推測し，CAN ID と送信 ECU の対応表を作成した．作成した対応表は付録 A に掲載した．本論文では，これを正解データとして用いる．

乗用車 1 の CAN バスにおける ECU の配置を図 23 に示す．乗用車 1 では CAN バスは 1 本のみの構成であり，エンジン ECU・A/T[9] ECU・ボディ系 ECU・ABS[10] ECU・EPS[11] ECU・コンビネーションメーターの 6 つの ECU から 16 種類の CAN ID を持つメッセージが送信されている．終端抵抗はエンジン ECU とボディ系 ECU に，それぞれ内蔵されている．バスの速度は 500 kbps である．CAN バスは OBD-II コネクタに接続されている．

6.2.2 乗用車 1 における計測実験 (2007 年式)

6.1 節と同様に，CAN メッセージ毎に計測結果の平均値と標準偏差を計算した．計測は 175.88 秒間行い，17961 の CAN メッセージを観測した．横軸を遅延時間 (ns)・縦軸を標準偏差 (ns) として，ECU 別に色分けし，CAN メッセージ毎にプロットしたグラフを図 24 に示す．

[9] Automatic/Transmission の略.
[10] Antilock Brake System の略.
[11] Electric Power Steering の略.

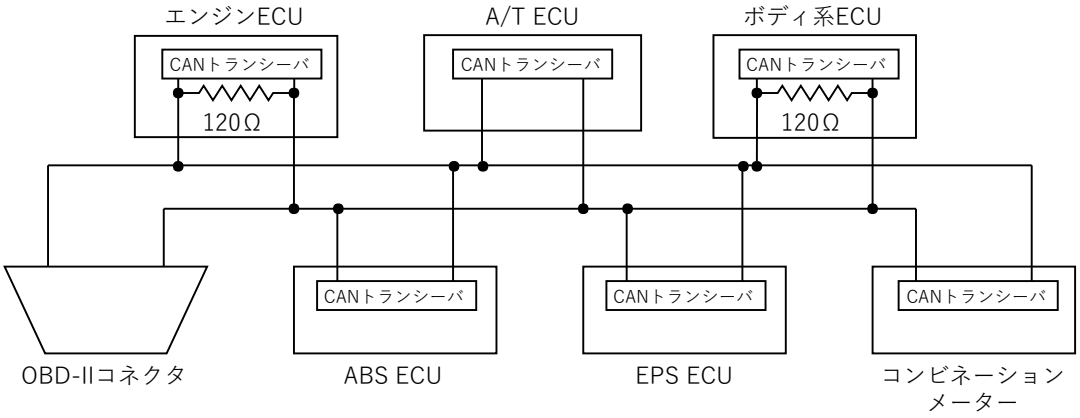

図 23　乗用車 1 の CAN バスにおける ECU の配置

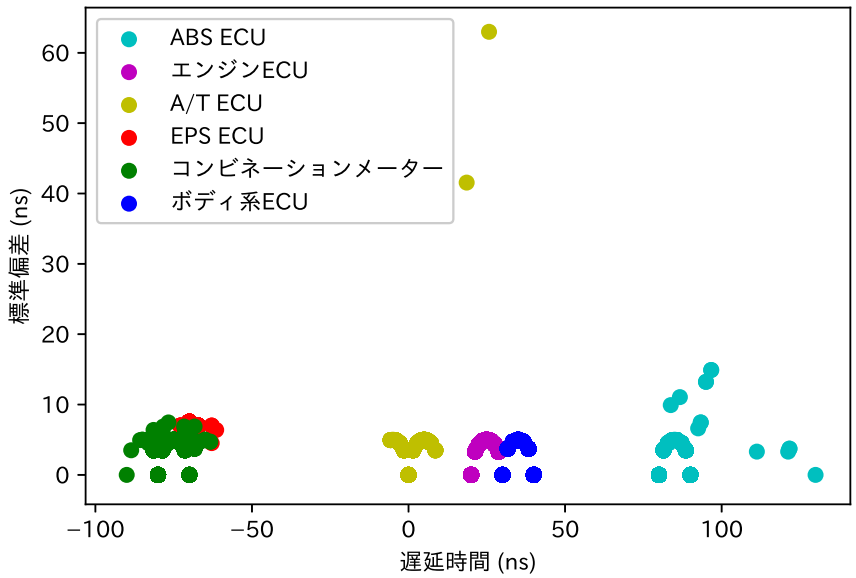

図 24　乗用車 1 での散布図

コンビネーションメーターと EPS ECU 間で点が重なっているが，それ以外の ECU に関しては，識別するために最低限必要な分解能があることが分かる．

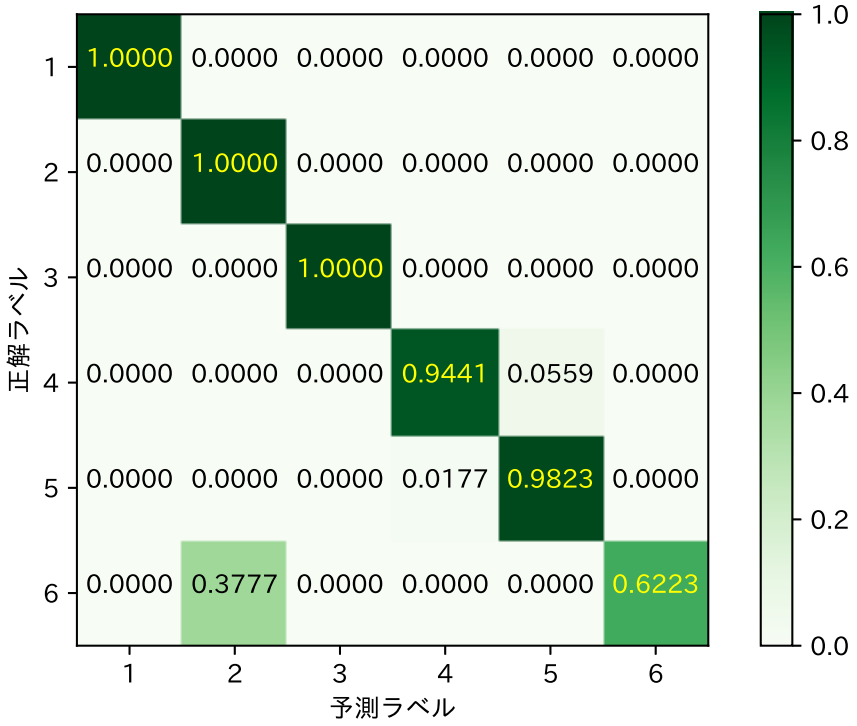

図 25　乗用車 1 での kNN アルゴリズムによる分類結果

6.1 節と同様に，kNN アルゴリズムによる分類を行った結果の混同行列を図 25 に示す．ラベルの番号は 1 から 6 まで順に図 24 における，ABS ECU，エンジン ECU，A/T ECU，EPS ECU，コンビネーションメーター，ボディ系 ECU に対応している．10 分割交差検証によってモデルの評価を行った結果，平均正解率は 0.9635 となった．

6.2.3　偽装攻撃下での実験

乗用車 1 において，エンジン回転数を示す CAN メッセージ (CAN ID = 0x308) を送信する実験を行った．OBD-II コネクタに攻撃用の CAN ノードを追加し，正

規のメッセージを上回る周期で送信を行った．攻撃中は，エンジンがアイドリング状態であるにも関わらず，タコメーターの針は偽装された，異常に高い値を指す．

この状態で，6.1 節と同様に遅延時間の計測を行い，CAN メッセージ毎に計測結果の平均値と標準偏差を計算した．横軸を遅延時間 (ns)・縦軸を標準偏差 (ns) として，ECU 別に色分けし，CAN ID 毎にプロットしたグラフを図 26 に示す．

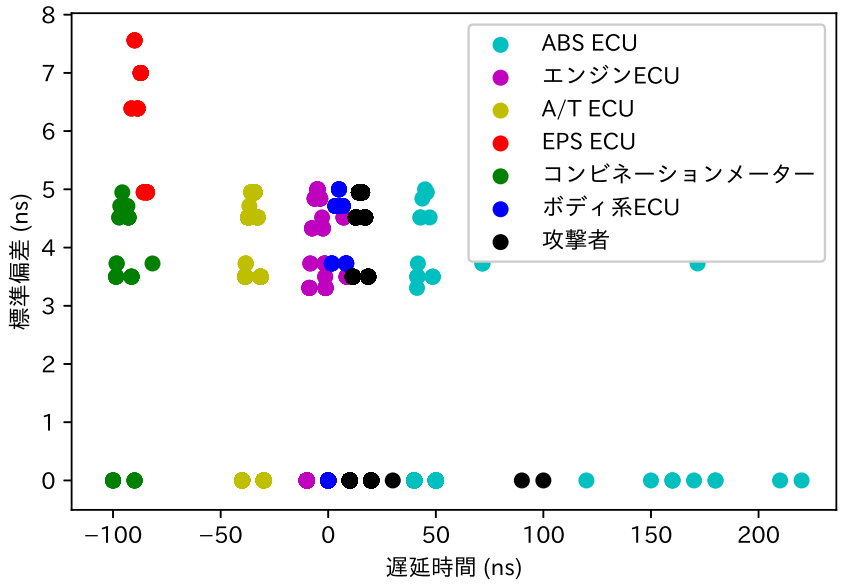

図 26　乗用車 1 におけるエンジン回転数偽装攻撃下での散布図

エンジン回転数を示す正規の CAN メッセージは，エンジン ECU から送信されている．紫色の点は正規のエンジン回転数を含むエンジン ECU から送信された CAN メッセージを示し，黒色の点は攻撃ノードから送信された偽装エンジン回転数の CAN メッセージを示す．

図より紫色の点と黒色の点は，線形分離可能なことが分かる．そこで，6.2.2 項の実験で構築した kNN モデルを用いて偽装攻撃を検知できるかどうか評価を行った．エンジン回転数を示す CAN ID のメッセージ関して，6.2.2 項の実験で構築した kNN モデルを用いてラベルを予測し，予測ラベルがエンジン ECU 以外であれば異常と判定する．評価に用いた異常検知ルールを表 6 に示す．

表 6 異常検知ルール

正解ラベル	予測ラベル	判定	分類
攻撃	エンジン ECU 以外	異常	True Positive
攻撃	エンジン ECU	正常	False Negative
エンジン ECU	エンジン ECU 以外	異常	False Positive
エンジン ECU	エンジン ECU	正常	True Negative

表 7 偽装攻撃下での攻撃検知の評価結果

正解ラベル		予測ラベル: Positive	予測ラベル: Negative
	Positive	1.0000	0.0000
	Negative	0.0036	0.9964

評価結果を表 7 に示す．True Positive Rate は 1.0000，True Negative Rate は 0.9964 となったため，正解率は 0.9982 となる．また，False Positive Rate は 0.0036 となった．

6.3 実験のまとめと考察

4.2 節の予備実験の結果から，CAN トランシーバにおける遅延時間の変化幅は 4.5 ns 程度である．テストベンチおよび実車における計測結果に基づく kNN アルゴリズムを用いた分類を行った．それぞれの平均正答率は 0.9738，0.9635 となり，どちらも 0.97 前後の平均正答率となった．

しかし，特定の ECU について局所的に分類精度が悪い結果となった．テストベンチにおける分類結果では，ELM327 2 を ELM327 1 と誤分類した割合が 0.5000，PiCAN2 を ELM327 1 と誤分類した割合が 0.8276 となった．図 21 における ELM327 1，ELM327 2 および PiCAN2 でプロットされた点が完全に重なっている部分があるため，計測デバイスの時間分解能を改善しても分類精度は向上しないと考えられる．よって，電圧値などの他の特徴量を併用することで分類精度を向上させられると考えられる．また，乗用車 1 における分類結果では，ボディ系 ECU をエンジン ECU と誤分類した割合が 0.3777 となった．これは，図 24 におけるボディ系 ECU とエンジン ECU の遅延時間の差が非常に小さいためと考えられる．そのため，この場合計測デバイスの時間分解能を向上させることで分類精度を向上させられると考えられる．

エンジン回転数偽装攻撃下での実験では，6.2.2 項で構築した kNN モデルを用いて異常検知性能の評価を行い，正解率は 0.9982，False Positive Rate は 0.0036 となった．

また，各実験の散布図では縦軸に標準偏差をプロットしているが，同じノードであっても標準偏差は大きくばらつく結果となった．よって，識別に必要な要素として標準偏差は関係ないと考えられる．

6.4 今後の課題

実験では，提案手法がCANノードの識別に対して有効であることを確認できた．本章ではkNNアルゴリズムを用いた分類を行い，10分割交差検証によってモデルを評価を行ったが，他の分類アルゴリズムを用いることで分類精度を向上できると考えられる．また，4.2節の予備実験において，CANトランシーバの短時間での正規性を確認したため，特定のCAN IDに対応する遅延時間の平均値・標準偏差が分かっていれば，正規分布を仮定して遅延時間の出現確率を計算することができる．よって，確率密度関数を利用したCANノードの識別実験を行い，正答率を評価することも今後の課題として挙げられる．

また，分類性能を向上させるための方法として，

1. よりクロック周波数の高いMCUを使用する．

2. 電圧値を併用する．

ことが挙げられる．1に関しては，クロック周波数を高くすることで，時間分解能が向上するため，分類能力が向上すると考えられる．2に関しては，遅延時間を測定すると同時にA/D変換器を用いてCANバスの電圧値を測定し，新たな特徴量として用いる方法である．時間ドメインと電圧(振幅)ドメインで，異なる特徴量が得られ，分類能力が向上すると考えられる．

7. 結論

本研究では，近年急速に電子制御化が進められている自動車の車載ネットワークに対する攻撃事例をいくつか挙げて，自動車におけるサイバーセキュリティ対策が課題となっていることを述べた．

車載ネットワークの構成例と各車載ネットワークプロトコルについて簡単に説明した．特に，事実上の標準であり，本研究で対象とする CAN プロトコルについて詳細に述べ，CAN の持つ本質的な脆弱性とそれを利用した攻撃手法について整理した．また，車載ネットワークへの攻撃として考えられる攻撃経路について，攻撃事例を交えながら紹介した．さらに，車載ネットワークへの攻撃対策として，車載システムを 4 階層に分けて考え，各層において対策を施し多段的・多層的に防御策をとるモデルを紹介した．

CAN バスへの攻撃対策の先行研究を IDS と送信元ノードの識別を目的としたものの 2 点に分類して紹介し，それぞれの手法の特徴と問題点について述べた．その上で，本研究の提案手法が，CAN メッセージの送信周期や内容に依存しないこと，既存の CAN バスの通信プロトコルを変更する必要が無いこと，A/D 変換器を必要としないことを特徴として述べた．

CAN バスにおける立ち上がり時と立ち下がり時の信号の遅延時間を利用した CAN ノードの識別手法を提案した．CAN トランシーバの送信部分の回路から，CAN ノードの遅延要因について述べた．また，予備実験として CAN トランシーバにおける立ち上がり時と立ち下がり時の遅延時間の差が時間とともにどのように変化するかを実測により確かめ，遅延時間の分布は，短期的には正規分布に従うことを確かめた．

提案手法の有効性を確認するため，CAN バスの信号の遅延時間を 10 ns の精度で計測するプロトタイプデバイスを試作した．試作したプロトタイプを用いて実験室内で構築した CAN バスのテストベンチおよび，実車で遅延時間の計測実験を行い，k-Nearest Neighber アルゴリズムを用いた ECU 分類性能の評価を行った．その結果，平均正答率はテストベンチにおいて 0.9738，実車実験において 0.9635 となった．

今後の課題として，他の分類アルゴリズムを用いた分類評価を行うこと，また，

識別性能を向上させるためによりクロック周波数の高い MCU を使用することや，電圧値を併用することが挙げられる．

謝辞

主指導教員であり，本研究を進めるにあたり，終始適切なご指導を頂いた本学情報基盤システム学研究室の藤川和利教授に深謝いたします．本研究の方向性について，貴重なご意見をくださいました副指導教員であり，本学情報セキュリティ工学研究室の林優一教授に深く感謝いたします．研究の方針から論文の執筆，学生生活まで数々のご助言と熱心なご指導をいただきました本学情報基盤システム学研究室の新井イスマイル准教授に心より感謝いたします．研究の方針や論文執筆について，貴重なご意見をいただきました本学情報基盤システム学研究室の垣内正年助教，油谷曉助教に心より感謝いたします．研究の方針について，貴重なご意見やご助言をくださいました東京電機大学の猪俣敦夫教授に深く感謝いたします．研究面や生活面において，様々なアドバイスをいただきました本学博士後期課程修了生であり，現東京工業大学 学術国際情報センターの石井将大助教に心から感謝いたします．また，日々の苦楽を共にし，日常の議論を通じて多くの知識や示唆を頂いた本学情報基盤システム研究室の皆様，サイバーレジリエンス構成学研究室の皆様に感謝いたします．いつも暖かい笑顔で励まし続け，様々な面から研究活動を支援くださいました元本学総合情報基盤センターの鷲尾多佳子さん，本学総合情報基盤センターの辻元理恵さん，中野彩子さんに感謝いたします．

最後に，私の意思を尊重してくださり，経済面や生活面で惜しみない援助と励ましを続けてくださいました家族に心から感謝いたします．

参考文献

[1] 株式会社富士キメラ総研. 『車載電装デバイス＆コンポーネンツ総調査 2017（下巻）』まとまる（2017/5/17 発表 第 17042 号）. https://www.fcr.co.jp/pr/17042.htm (閲覧日: 2018 年 1 月 30 日).

[2] GMO クラウド株式会社. 自動車向け IoT ソリューションを提供国内初、車とアプリの連携で車両コンディションの自動解析・遠隔診断を実現. https://www.gmo.jp/news/article/?id=5741 (閲覧日: 2018 年 1 月 16 日), 2017.

[3] 株式会社スマートドライブ. DriveOn. https://driveonapp.com/ (閲覧日: 2018 年 1 月 16 日).

[4] Karl Koscher, Alexei Czeskis, Franziska Roesner, Shwetak Patel, Tadayoshi Kohno, Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, and Stefan Savage. Experimental Security Analysis of a Modern Automobile. In *2010 IEEE Symposium on Security and Privacy*, pp. 447–462, 2010.

[5] Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, Stefan Savage, Karl Koscher, Alexei Czeskis, Franziska Roesner, and Tadayoshi Kohno. Comprehensive Experimental Analyses of Automotive Attack Surfaces. In *Proceedings of the 20th USENIX Conference on Security*, SEC'11, Berkeley, CA, USA, 2011.

[6] Charlie Miller and Chris Valasek. Remote Exploitation of an Unaltered Passenger Vehicle. In *DEFCON 23*, 2015.

[7] Keen Security Lab. Car Hacking Research: Remote Attack Tesla Motors. https://keenlab.tencent.com/en/2016/09/19/Keen-Security-Lab-of-Tencent-Car-Hacking-Research-Remote-Attack-to-Tesla-Cars/ (閲覧日: 2018 年 1 月 23 日), 2016.

[8] Ian Foster, Andrew Prudhomme, Karl Koscher, and Stefan Savage. Fast and Vulnerable: A Story of Telematic Failures. In *9th USENIX Workshop on Offensive Technologies (WOOT 15)*, Washington, D.C., 2015.

[9] 一般財団法人自動車検査登録情報協会. わが国の自動車保有動向 - 車種別の平均使用年数推移表. http://www.airia.or.jp/publish/file/r5c6pv000000g7wb-att/r5c6pv000000g7wq.pdf (閲覧日: 2018 年 1 月 16 日), 2017.

[10] Kyong-Tak Cho and Kang G. Shin. Fingerprinting Electronic Control Units for Vehicle Intrusion Detection. In *25th USENIX Security Symposium (USENIX Security 16)*, pp. 911–927. USENIX Association, 2016.

[11] Pal-Stefan Murvay and Bogdan Groza. Source Identification Using Signal Characteristics in Controller Area Networks. *IEEE Signal Processing Letters*, Vol. 21, No. 4, pp. 395–399, 2014.

[12] Wonsuk Choi, Hyo Jin Jo, Samuel Woo, Ji Young Chun, Jooyoung Park, and Dong Hoon Lee. Identifying ECUs Using Inimitable Characteristics of Signals in Controller Area Networks. *CoRR*, 2016.

[13] Kyong-Tak Cho and Kang G. Shin. Viden: Attacker Identification on In-Vehicle Networks. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, pp. 1109–1123. ACM, 2017.

[14] Robert Bosch GmbH. CAN specification Ver. 2.0B, 1991.

[15] State of California. On-Board Diagnostic II (OBD II) Systems - Fact Sheet / FAQs. https://www.arb.ca.gov/msprog/obdprog/obdfaq.htm (閲覧日: 2018 年 1 月 23 日), 2015.

[16] 国土交通省. 道路運送車両の保安基準の細目を定める公示【2016. 10. 31】 別添 48. http://www.mlit.go.jp/common/001184853.pdf (閲覧日: 2018 年 1

月 23 日), 2016.

[17] 国土交通省. 排出ガス測定方法及び車載式故障診断装置の基準等を改正しました。 http://www.mlit.go.jp/kisha/kisha06/09/091101_2_.html (閲覧日: 2018 年 1 月 26 日), 2006.

[18] Jiajia Liu, Shubin Zhang, Wen Sun, and Yongpeng Shi. In-Vehicle Network Attacks and Countermeasures: Challenges and Future Directions. *IEEE Network*, Vol. 31, No. 5, pp. 50–58, 2017.

[19] 田中政志, 高橋順子, 大嶋嘉人. クルマのサイバー攻撃対策技術. NTT 技術ジャーナル, Vol. 29, No. 2, pp. 14–17, 2017.

[20] Troy Hunt. Controlling vehicle features of Nissan LEAFs across the globe via vulnerable APIs. https://www.troyhunt.com/controlling-vehicle-features-of-nissan/ (閲覧日: 2018 年 1 月 26 日), 2016.

[21] Catalin Cimpanu. You Can Hack Some Mazda Cars with a USB Flash Drive. https://www.bleepingcomputer.com/news/security/you-can-hack-some-mazda-cars-with-a-usb-flash-drive/ (閲覧日: 2018 年 1 月 30 日), 2017.

[22] AUTOSAR. Specification of Module Secure Onboard Communication. AUTOSAR Release 4.2.2, 2015.

[23] Hyun Min Song, Ha Rang Kim, and Huy Kang Kim. Intrusion detection system based on the analysis of time intervals of CAN messages for in-vehicle network. In *2016 International Conference on Information Networking (ICOIN)*, pp. 63–68, 2016.

[24] 芳賀智之, 岸川剛, 鶴見淳一, 松島秀樹, 高橋良太, 佐々木崇光. 機械学習による車載ネットワーク攻撃検知システム. *Panasonic Technical Journal*, Vol. 63, No. 1, pp. 16–21, 2017.

[25] Symantec Corp. Symantec Launches New IoT Solution to Help Carmakers Protect Against Zero Day Attacks. https://www.symantec.com/about/newsroom/press-releases/2016/symantec_0608_01 (閲覧日: 2018 年 1 月 17 日), 2016.

[26] パナソニック株式会社. サイバー攻撃に対抗するオートモーティブ侵入検知・防御システムを開発. http://news.panasonic.com/jp/press/data/2017/10/jn171010-2/jn171010-2.html (閲覧日: 2018 年 1 月 31 日), 2017.

[27] NNG. Automotive Cyber Security Reaches New Milestone. https://www.nng.com/newsroom/press-releases/automotive-cyber-security-reaches-new-milestone/ (閲覧日: 2018 年 1 月 16 日), 2017.

[28] Sang Uk Sagong, Xuhang Ying, Andrew Clark, Linda Bushnell, and Radha Poovendran. Cloaking the Clock: Emulating Clock Skew in Controller Area Networks. *CoRR*, 2017.

[29] John Griffith. Learn the inner workings of a CAN bus driver and how to debug your system. https://e2e.ti.com/blogs_/b/industrial_strength/archive/2016/01/26/the-inner-workings-of-a-can-bus-driver (閲覧日: 2018 年 1 月 18 日), 2016.

[30] 岩田穆. CMOS 集積回路の基礎. 科学技術出版, 2000.

[31] ローム株式会社. MOSFET の特性. http://www.rohm.co.jp/web/japan/tr_what5-j (閲覧日: 2018 年 1 月 27 日).

[32] 村田製作所. 水晶振動子/発信器. https://www.murata.com/~/media/webrenewal/support/library/catalog/products/timingdevice/crystalu/p79.ashx?la=ja-jp (閲覧日: 2018 年 1 月 27 日), 2017.

[33] 岩田彰. ディジタル信号処理. オーム社, 2013.

[34] 宮川公男. 基本統計学 [第 3 版]. 有斐閣, 1999.

発表リスト

1. 北川 智也, 垣内 正年, 新井 イスマイル, 猪俣 敦夫, 藤川 和利. 車載の社外品ドングルに対する近接攻撃の検証. コンピュータセキュリティシンポジウム 2017 論文集, pp. 1481-1485, 2017.

2. Tomoya Kitagawa, Ismail Arai, Masatoshi Kakiuchi, Atsuo Inomata and Kazutoshi Fujikawa. Fingerprinting of ECUs using difference of delay time on Controller Area Networks. ISCIS Security Workshop 2018, 2018.

付録

A. CAN IDと送信元ECUの対応表

6.2節で用いた乗用車におけるCAN IDと送信元ECUの対応表を表8に示す.

表8 乗用車1におけるCAN IDと送信元ECUの対応

CAN ID	送信元ECU	周期 (ms)
0x200	ABS ECU	20
0x215	ABS ECU	20
0x231	ABS ECU	20
0x210	エンジンECU	20
0x212	エンジンECU	20
0x308	エンジンECU	20
0x312	エンジンECU	10
0x415	エンジンECU	20
0x608	エンジンECU	100
0x218	A/T ECU	20
0x317	A/T ECU	20
0x418	A/T ECU	20
0x2F2	EPS ECU	100
0x408	コンビネーションメーター	100
0x412	コンビネーションメーター	100
0x424	ボディ系ECU	40