Computer Security Symposium 2017
23 - 25 October 2017

Keep AliveによるCANの攻撃検知手法の提案

倉地 亮[1]　高田 広章[1]　上田 浩史[2]　宮下 之宏[2]

概要：近年，自動車の制御システムに対する攻撃事例が多数報告されており，セキュリティ強化が必要とされている．このため，メッセージ認証子を付与することにより，セキュリティを確保する手法が適用されることが目されている．一方，すべてのメッセージにメッセージ認証子を付与することは転送容量の限界から困難であるため，一部のメッセージにのみ付与することが検討されている．このため，重要なメッセージの 1 つである Keep Alive メッセージを利用した攻撃検知手法について提案する．

キーワード：自動車セキュリティ, 組込みセキュリティ，キープアライブメッセージ，Controller Area Network(CAN)

Proposal of spoofing detection based on Keep Alive Messages for Controller Area Network

Ryo Kurachi[1]　Hiroaki Takada[1]　Hiroshi Ueda[2]　Yukihiro Miyashita[2]

Abstract: Many researches have been reported feasibility of attack against in-vehicle networks. Thus, it is necessary to strengthen security level. It is expected that assigning a message authenticator is applied to in-vehicle networks. On the other hand, it is difficult to assign a message authenticator to all messages, so it is limited to give only some messages. In this paper, we propose a method to protect CAN messages by keep alive messages.

Keywords: Automotive security, Embedded Security, Keep Alive Message, Controller Area Network (CAN)

1. はじめに

現在，車載制御ネットワークでは Controller Area Network (CAN) が広く使われている [1]．一方，近年，攻撃者が CAN メッセージを偽造することによるなりすまし攻撃により，セキュリティの脅威事例が多数報告されている [2], [3], [4]．これらの脅威事例からも，現在販売されている車両の多くはセキュリティ機能が十分に搭載されていないことが課題である．このため，今後販売される車両には様々なセキュリティ強化技術が適用されることが予想されている．

CAN メッセージのなりすまし攻撃に対する強化技術の一つとして，AUTomotive Open System ARchitecture (AUTOSAR) で規定される Secure Onboard Communication(SecOC) 仕様がある [5]．CAN メッセージのペイロードの一部に Message Authentication Code (MAC) を付与することにより改ざんを防止することを目的としており，今後はこの仕様が適用されると予想される．

一方, 快適性や安全性を追求するために, 自動車には様々な機能の搭載が要求されている．このため，搭載される Electronic Control Unit(ECU) の数は増加しており，CAN バスの転送容量はひっ迫している．このため，すべての CAN メッセージに MAC を付与することは難しいと考えられており，CAN with Flexible Data-Rate(CAN-FD)[6] などの転送容量が大きいプロトコルへと移行することが検討されている．しかしながら一方で，車載制御システム全

[1] 名古屋大学大学院情報学研究科
Graduate School of Informatics, Nagoya University
[2] 株式会社オートネットワーク技術研究所/住友電気工業株式会社
AutoNetworks Technologies, Ltd./Sumitomo Electric Industries, Ltd.

体を CAN-FD に移行するには膨大な開発コストが必要となるなどの懸念があり，転送容量が必要となる ECU から徐々に CAN-FD へと移行することが予想されている．このため，今後も CAN バスと CAN-FD バスが混在するネットワーク構成になることが予想されている [7].

1.1 本研究の位置付け

先行研究では，CAN メッセージに MAC を効率的に付与する方法が提案されている．まず，Lin や Xie らにより，CAN メッセージのシグナル単位での組み替えによる最適化手法が提案されている [8], [9].

また，AUTOSAR の SecOC を検証するノードを限定することにより，CAN バス上の 1 つのノードで MAC を検証する手法が提案されている [10].

しかしながら，いずれの研究においても，CAN の転送容量が十分に使用できることが前提となっており，CAN メッセージの転送容量が十分にない場合は想定されていない．このため，本論では，セキュリティ強化により増加される転送容量を低減することを目的とした攻撃検知手法を提案する．

本研究の貢献は，以下の通りである．

(1) 本論では，車載制御ネットワーク上で使用が想定されるネットワークマネージメント (NM) 手法の中で規定される Keep Alive メッセージを利用して，攻撃検知を行う．これにより，既存するメッセージセットの変更規模を限定できる．
(2) 本論では，ハードウェアの改造をすることなく，実装することが可能である．このため，既存する ECU でも実現することが可能である．

1.2 論文の構成

以降では，まず 2 章にて，MAC 付与に対する課題について説明する．続く 3 章では，提案手法のアイデアとその原理について述べる．その後，4 章にて評価を述べ，最後に 5 章で本論をまとめる．

2. MAC 付与に対する課題

前述するように，自動車業界では，SecOC の仕様を用いてセキュリティ強化する方法が検討されている．2016 年 11 月に発行された AUTOSAR の R4.3.0 では，SecOC の仕様中に MAC 長や再送防止用のカウンタ長を規定したセキュリティプロファイルが定義されている [5]．このプロファイルでは，(1)MAC 生成アルゴリズム, (2) MAC の長さ，(3) 転送する MAC の長さ, (4) 再送攻撃防止用のカウンタの長さ，(5) 転送するカウンタの長さが規定されている．これらのプロファイルの内，本研究では，JasPar[11] で規定されたプロファイルを基準に議論する．この JasPar のプロファイルは，CAN メッセージに付与する MAC の長さは 24 ビット, カウンタの長さは 8 ビットと規定されており，CAN メッセージに付与される認証情報の合計は 32 ビット (4 バイト) となる．

もし既存する CAN メッセージが 5 バイト以上のペイロードを使用している場合には，この JaPar で規定されるプロファイルをそのまま適用することができない．つまり，この場合には，既存する CAN メッセージを 2 つに分割したり組み替えることにより各メッセージに 4 バイトの MAC を付与することになる．このようにメッセージを組み替えてしまうと，既存する ECU 上のソフトウェアの改変が必要となり，開発コストが上がることが問題である．

3. 提案手法

3.1 提案の背景技術と概要

本研究では，CAN のネットワークマネージメント (NM) を利用する攻撃検知手法について提案する．このため，以降では，まず CAN のネットワークマネージメントを説明した上で，提案手法のアイデアとその実現方法について説明する．

3.1.1 CAN のネットワークマネージメント

CAN のネットワークマネージメントは，従来は OSEK，現在は AUTOSAR の CAN NM 仕様で定義されている．OSEK の場合は，明示的な Keep Alive メッセージにより実現される Direct-NM と間接的に周期メッセージを受信することにより実現される Indirect-NM の 2 つの方式により実現されていた．しかしながら，AUTOSAR では Direct-NM のみが言及されており，明示的に Keep Alive メッセージを送信する方式に集約されている．

CAN のネットワークマネージメントでは，マスターとスレーブの 2 つの種類のノードに分類される．一般的に，CAN バス上の 1 つのノードがマスターノードとなり，その他の ECU はスレーブノードとして運用されることが多い．

ネットワークマネージメントの目的は，Keep Alive メッセージによるスレーブノードの生存確認とスリープ時の合意の 2 つが挙げられる．本研究では前者の Keep Alive メッセージによるスレーブノードの生存確認を利用し攻撃検知を行うものとする．

3.1.2 AUTOSAR の CANNM 仕様

AUTOSAR において CANNM が定義されている [12]．前述するように，CANNM では，明示的に Keep Alive メッセージを使用すること及び，このメッセージのペイロードのフォーマットが規定されている．

ペイロードのフォーマットについては，**図 1** に示すように，Byte 0 には送信ノードの ID，Byte 1 は CAN NM を実現する上での制御情報が入ることが規定されている．つづく，Byte2 から Byte7 はユーザーが任意のデータを設定できる．尚，NM 関連メッセージの送信周期については，ユーザーが任意に設定することができる．

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte7	User data 5							
Byte6	User data 4							
Byte5	User data 3							
Byte4	User data 2							
Byte3	User data 1							
Byte2	User data 0							
Byte1	Control Bit Vector							
Byte0	Source Node Identifier							

図 1　CAN NM のペイロードの構成例

3.2　提案手法のアイデア

本論では，前述する AUTOSAR の CAN NM で規定される Keep Alive メッセージに各ノードが送信したメッセージの数や健康状態を付与することにより，各ノードが正しく動作しているかどうかを監視する．より具体的には，以下の**図 2** を用いて説明する．

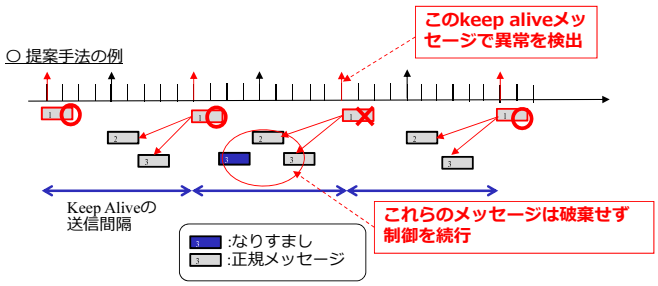

図 2　提案手法の実現例

この例では，幾つかの前提を持つ．まず，ある CAN バス上に接続されるゲートウェイが NM のマスターノード，その他の ECU がスレーブノードとする．このとき，スレーブノードである各 ECU は本提案手法に従い，Keep Alive メッセージを周期的に送信する．この Keep Alive メッセージには，各 ECU 自身が送信する全メッセージの送信カウント数を付与するものとする．また，この送信カウント数は，Keep Alive メッセージの送信ごとに 0 でクリアされるものとする．さらに，少なくとも Keep Alive メッセージは，SecOC で規定されるように MAC を付与することにより改ざんを防止するものとする．つまり，図 1 の Byte 4 から Byte 7 に JasPar で規定されたメッセージに認証情報を付与し，送信カウンタは Byte 2 と Byte 3 に付与されるものとする．

このとき，図 2 に示すように，攻撃者が CAN バス上に任意のタイミングでなりすましメッセージ 3 を送信する．一方，本提案手法を適用する場合には，Keep Alive メッセージには Keep Alive メッセージ間における各 CAN メッセージの送信カウント数が付与される．このため，この攻撃対象となったメッセージ 3 を送信するノードからの Keep Alive メッセージを受信すれば，正規 ECU から送信されたメッセージ 3 の送信カウント数よりも多いために，受信ノードは送信メッセージ数が多かったことを知り攻撃を検知できる．

しかしながら，この提案手法を実現するためには，いくつかの課題が存在する．まず 1 つ目に，Keep Alive メッセージを送信するノードは，確かに自身が送信したメッセージ数を Keep Alive メッセージに付与する必要がある．また，2 つ目に，Keep Alive メッセージの監視ノードは，実際に CAN バス上に送信された各メッセージ数をカウントし，各 ECU の Keep Alive メッセージを受信した際にはカウント数が正しいかどうかを検証する必要がある．これらに 2 つの課題の解決方法は，次節で詳細を述べる．

3.3　提案手法の実現方法

前述するように，本提案手法では，Keep Alive メッセージの送信ノードと Keep Alive メッセージの監視ノードに分類される．それぞれに対して，送信と監視のアルゴリズムが必要になる．

3.3.1　Keep Alive 送信アルゴリズム

まず，Keep Alive メッセージの送信については，以下の制約がある．Keep Alive メッセージを送信する場合，自ノードから送信されたメッセージの個数をカウントしてから送信要求を実行する必要がある．このため，NM メッセージが他のメッセージに追い越されたり，他のメッセージを追い越して送信してしまう場合には，Keep Alive メッセージの情報が誤る．つまり，Keep Alive メッセージの送信ノードは，Keep Alive メッセージが他のメッセージを追い越したり，追い越されたりしないように制御する必要がある．

本研究では，この性質を満たすために，Keep Alive メッセージの送信には，以下の通りの制約を仮定した．

- 各送信メッセージの送信完了時に該当するカウンタをインクリメントすること
- Keep Alive メッセージが他のメッセージに追い越されないために，必ず各 ECU 内で送信される CAN メッセージの中で最も優先度が高くなるように設定した．
- Keep Alive メッセージが他のメッセージを追い越さないために，Keep Alive メッセージの送信要求時は，以下で定義する **Algorithm 1** に従い，他の既に送信要求されているすべてのメッセージの送信が完了したことを確認してから，Keep Alive メッセージの送信カウント数を確定するようにした．

Algorithm 1 をより具体的に説明する．まず，Keep Alive メッセージの送信要求が行われると，既に送信要求されている CAN メッセージがすべて送信完了するまで待つものとした (図中の 2 行目から 5 行目)．その際，Keep Alive メッセージ以外のメッセージの送信要求が行われた場合，それらのメッセージは 5 行目に示されるように一旦送信をペンディングし，Keep Alive メッセージの送信要求

Algorithm 1: TRANSMISSION ALGORITHM OF SECURE KEEP ALIVE MESSAGES ON CAN

Input: Keep Alive Message (KAM_i) の送信要求
Output: KAM_i を優先するために送信要求をペンディングされたメッセージのリスト ($QUEUE_{wait}$)

```
// Step1. すべての送信メールボックスが空になるまで待つ
while Exist messages in Mailbox for transmission do
    // Wait for transmit completion of existing transmission messages
    if Arrive transmit request of any other messages m_j then
        QUEUE_wait ← m_j
// Step2. Keep Alive メッセージを送信メールボックスへ挿入
Mailbox_I^prio ← KAM_i
// Step3. ペンディングされたすべてのメッセージを送信要求
while Exist m_j messages in QUEUE_wait do
    Mailbox_I^{j≠prio} ← m_j
return QUEUE_wait
```

後に送信要求を実行するよう，$QUEUE_{wait}$ に入れる方式を採った．その後，MailBox が空になると，まずそのノードの最高優先度メッセージである Keep Alive メッセージを送信要求し，次にペンディングされていたより低い優先度の送信メッセージを送信要求することにより，他のメッセージが Keep Alive メッセージを追い越さないよう実現した．

3.3.2 Keeep Alive 監視アルゴリズム

まず，Keep Alive メッセージを監視するノードは，Keep Alive メッセージに付与されるすべてのメッセージの CAN バス上への送出カウント数を保持する必要がある．その上で，監視対象となる Keep Alive メッセージを受信した後に，受信したメッセージ数と Keep Alive メッセージに付与される送信カウント数を比較する．その結果，もし実際に受信したメッセージ数と Keep Alive メッセージに付与される送信カウント数が不一致となれば，攻撃検知とみなしアラートを上げる．

このとき，監視ノード側で保持するカウンタの数は，設計により異なる．つまり，Keep Alive メッセージの送信カウンタがどのように付与するかに依存している．より具体的には，ある ECU からの Keep Alive メッセージには，そのノードから送信される全 10 個のメッセージを区別せず 1 つのカウンタとして実装される場合と，各メッセージごとに 10 個のカウンタを付与する場合などがある．このあたりのカウンタの実装方法については，Keep Alive メッセージにより，攻撃対象となったメッセージを特定したいなどの要求がある場合には細かい単位で設計され，あるノードが攻撃されていることだけ知りたい場合には大きい粒度で設計されることを想定する．

次に，監視ノードの状態としては，**図 3** に示す監視状態を保持する．まず監視を開始するのは，すでに監視対象となっている ECU からの最初の Keep Alive メッセージが届いた後に監視を開始する．このとき，監視ノードでは該当する各メッセージの受信カウンタの値を 0 でクリアする．その上で，非監視モードから監視モードへと遷移し各 CAN メッセージの受信状況を監視する．監視を終了するのは，監視ノードが自ら監視対象 ECU のスリープを要求し，且つ監視対象 ECU からの Sleep 完了を受信した後とする．監視ノードは，このような状態を監視対象 ECU の数だけ保持しているものとする．

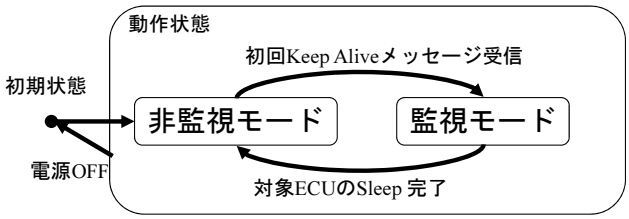

図 3 監視ノードの監視状態

3.4 提案手法の強い点と弱い点

既存手法である各 CAN メッセージに MAC を付与する場合と提案手法の比較を行う．まず提案手法の強い点として，以下の 2 点が挙げられる．1 つ目に，提案手法では，各メッセージに MAC を付与するよりも通信容量が低減できる可能性がある．2 つ目に，提案手法は既存する各メッセージに MAC を付与する手法と併用できる．このため，MAC が付与されるメッセージを Keep Alive 信号でさらに守ることも可能である．つまり，本提案手法を適用する場合には，MAC が付与されるメッセージ A を改ざんしたとしても，Keep Alive メッセージである B も改ざんしない限りは攻撃が成立しないことになる．

一方，提案手法の弱い点としては，以下の 2 点が挙げられる．まず 1 つ目に，提案手法では攻撃検知が可能となるタイミングは，必ず Keep Alive メッセージを受け取るタイミングである．このため，Keep Alive メッセージ間に転送される攻撃メッセージは受信ノードが受け取り制御を実行してしまう．このため，Keep Alive 信号の受信間隔については，例えば数十ミリ秒単位での短い時間に設定することが望ましい．次に 2 つ目に，提案手法を適用する場合，Keep Alive メッセージを送信するノードと監視するノードのソフトウェア処理を変更する必要が生じる．

4. 提案手法の評価

本提案手法を評価するために，2 つの観点で提案手法の有効性を示す．まず 1 つ目に，本提案手法の攻撃耐性について議論する．次に 2 つ目として，本提案手法と MAC を付与した場合のバス負荷率の増減について評価する．

4.1 評価 1: 攻撃耐性の評価

まず提案手法の弱い点について分析する．本提案手法では，NM メッセージに MAC を付与することを想定しており，Keep Alive メッセージや Sleep 制御に用いられるメッセージには MAC を付与するものとする．また，本提案手法では，各 MAC が付与されたメッセージの受信タイミングではなく，Keep Alive メッセージの受信タイミングでしか攻撃を検出することができないことが弱点である．つまり，もしなりすましメッセージを 1 つ受信したのみで制御が実行されるような場合には，攻撃が成功してしまう可能性が高いが，例えば，Keep Alive メッセージの送信間隔が長すぎなければ，制御としては一瞬挙動がおかしくなるものの，それほど大きな影響なく攻撃が検出できる可能性はある．また，Keep Alive メッセージが DoS 攻撃などにより妨害される場合には，Keep Alive メッセージなどが送受信できなくなり，送信カウント数を通知したり受信することができない．この場合，NM の要件より，Keep Alive メッセージがある決められた一定時間内に到達しなければ，当該 ECU から送信される他のメッセージの受信も一旦停止するなどの制御が必要となる．しかしながら，これは通常の NM の要件でもあるため，容易に実現できる．さらに，Keep Alive メッセージのなりすましについては認証情報により保護されているため，容易に偽造することは難しい．また，再送攻撃についても同様に，SecOC で規定されるように再送防止用カウンタを保持しているため，保護されている．このため，本手法を適用するうえで，これらの弱い点を注意深く吟味し，適切な送信周期や認証情報を付与する必要がある．

4.2 評価 2: バス効率の評価

本評価では，本論で提案する Keep Alive メッセージを追加する場合 (case1) と既存手法である MAC を追加する場合 (case2) の 2 つのメッセージ追加方法を説明した上で，セキュリティを高めるために増加したバス負荷率について比較し議論する．尚，本評価では実際に自動車メーカーにて使用されるメッセージセットを用いて評価する．本メッセージセットの概要は，ある 1 つの CAN バス上に 14 個の ECU が接続されており，CAN バス上を流れるメッセージ数は 64 個，すべて周期送信メッセージとして定義されており，バス負荷率は 53.27%である．このメッセージセットはセキュリティ強化前のものであり，MAC や Keep Alive メッセージが追加されていない．このため，本メッセージセットをベースにバス負荷の増加率を比較し評価する．

4.2.1 case 1: Keep Alive メッセージの追加による保護

まず，既存するメッセージセットに Keep Alive メッセージを追加する方法について説明する．Keep Alive メッセージを追加する場合，各 ECU に 1 つの Keep Alive メッセージを追加するものとする．このとき，ここでは追加される Keep Alive メッセージが各 ECU 内の送信メッセージの中で最も優先度が高い場合 (KAM-pattern1) と中程度の場合 (KAM-pattern2) の 2 つの場合を議論する．また，追加する Keep Alive メッセージの送信周期は Rate monotonic の原理に従い，優先度に応じて短くなるものとする．つまり，Keep Alive メッセージの優先度が低くなるほど，送信周期が長くなり，バス負荷率は低減できるが，攻撃の検出が遅くなることが懸念される．特に，あまりに優先度を低くしてしまうと，Keep Alive メッセージの到着間隔内になりすましメッセージが連続して送信され制御が乗っ取られる危険性があるため，前述する 2 つの場合を想定した．

ここで，このような前提を用いて割り当てられた Keep Alive メッセージの CAN バスの負荷率について表 2 に示す．この結果より，各 ECU 内で最も優先度が高くなるよう Keep Alive メッセージを追加した場合のバス負荷率の増加は，KAM-pattern1 に示すとおり，17.82%である．一方，各 ECU 内で送信されるメッセージの中程度の優先度として追加する場合には，KAM-pattern2 に示すとおり，6.07%となる．この結果より，最高優先度とした場合でも CAN バス上の負荷率は帯域内に収まっており実現可能性があることを示している．

表 1 KAM(Keep Alive Message) の割り当てによる増加した負荷率

	バス負荷率 (%)	KAM の負荷率 (%)
Base	53.27	0
KAM-pattern1	73.54	17.82
KAM-pattern2	59.34	6.07

4.2.2 case 2: MAC 追加による保護

まず，既存するメッセージセットに MAC を追加する方法について説明する．前述するように，MAC を追加する場合，既存するメッセージがすでにペイロードを 5 バイト以上使用している場合には，2 つの CAN メッセージに分割する必要がある．ここでは，この分割された 2 つの CAN メッセージにそれぞれの MAC を付与して転送することとする．また，MAC を付与するメッセージは，CAN メッセージの最高優先度から上位 25 %に MAC を付与する場合 (MAC-pattern1) と上位 40 %を付与する場合 (MAC-pattern2) の 2 つの場合を想定した．

このときの結果を，表 2 に示す．上位 25 %のメッセージ (16 メッセージ) に MAC を付与した場合の結果は，表中の MAC-pattern1 に示すように，24.06%増加した．このとき，MAC を付与した 16 メッセージのうち，11 メッセージが 5 バイト以上のメッセージであったために，2 つのメッセージに分割される結果となった．また，上位 40%のメッセージに MAC を付与した場合，表中の MAC-pattern2 に示されるように，41.96%バス負荷が増加した．この結果，

全体のバス負荷率が 95.23%となり，これ以上は MAC を付与できないことを示している．

表 2 MAC を追加した場合の増加した負荷率

	バス負荷率 (%)	MAC の負荷率 (%)
Base	53.27	0
MAC-pattern1	77.33	24.06
MAC-pattern2	95.23	41.96

これらの結果より，MAC をすべてのメッセージに付与することは現実的ではないことがわかる．一方，提案手法を用いる場合には，最高優先度のメッセージとして転送する KAM-pattern1 でもすべてのメッセージを対象として保護することが可能である．この結果より，本提案手法の有効性を示した．

5. まとめ

車載制御システムに対する CAN メッセージを利用した攻撃事例が多数報告されている．これに対して，自動車業界では，AUTOSAR などで規定される MAC を利用した CAN メッセージの改ざん保護手法の導入が進められている．しかしながら，一方で，すべての CAN メッセージに MAC を付与することは通信容量がひっ迫しており可能ではない．このため，本研究では，車載制御ネットワークで使用される Keep Alive メッセージを使用した攻撃検知手法について提案した．提案手法の有効性を示すために，セキュリティ分析や通信容量のオーバーヘッドの比較により評価した．この結果，本提案手法が転送容量のオーバヘッドを抑えつつ複数のメッセージを保護できることを示した．

謝辞 本研究開発の一部は，総務省 戦略的情報通信研究開発推進事業 SCOPE 若手 ICT 研究者等育成型研究開発 (152106005) の委託を受けたものです．

参考文献

[1] Leohold, J.. Communication Requirements for Automotive Systems, 5th IEEE Workshop on Factory Communication Systems, 2004.

[2] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, xperimental Se- curity Analysis of a Modern Automobile, IEEE Symposium on Security and Privacy, 2010.

[3] C. Valasek, C. Miller, ”Adventures in Automotive Networks and Control Unit”, http://www.ioactive.com/pdfs/Ioactive_Adventures_in_Automotive_Networks_and_Control_Unts.pdf, 2014

[4] C. Miller, C.Valasek, “Remote Exploitation of an Unaltered Passenger Vehicle” http://illmatics.com/Remote\%20Car\%20Hacking.pdf, 2015.

[5] Specification of Module Secure Onboard Communication AUTOSAR CP Release 4.3.0, 2016.

[6] International Organization for Standardization. Road vehicles - Controller area network (CAN) - Part 1: Data link layer and physical signaling, ISO11898-1, 2015.

[7] O. Esparza, W. Leichtfried, F. Gonzalez, ”Transitioning applications from CAN 2.0 to CAN FD”, https://www.can-cia.org/fileadmin/resources/documents/proceedings/2015_esparza.pdf, 2017.

[8] Lin C W, Zhu Q, Phung C, Sangiovanni-Vincentelli A. Security-aware mapping for CAN-based real-time distributed automotive systems. In: Proceedings of the 2013 IEEE/ACM International Conference on Computer-Aided Design (ICCAD). San Jose, CA: IEEE, 2013. 115?121

[9] Y. Xie, L. Liu, R. Li, J. Hu, Y. Han and X. Peng, ”Security-aware signal packing algorithm for CAN-based automotive cyber-physical systems,” in IEEE/CAA Journal of Automatica Sinica, vol. 2, no. 4, pp. 422-430, October 10 2015.

[10] Kurachi, R., Matsubara, Y., Takada, H., Adachi, N., Miyashita, Y., and Horihata, S., ”CaCAN - Centralized Authentication System in CAN”, Proceedings of the escar 2014 Europe Conference, Hamburg, Germany, Nov 2014

[11] Japan Automotive Software Platform and Architecture (JasPar), https://www.jaspar.jp/, 2017.

[12] Specification of CAN Network Management AUTOSAR CP Release 4.3.0, 2016.