(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第5584767号
(P5584767)

(45)発行日　平成26年9月3日(2014.9.3)　(24)登録日　平成26年7月25日(2014.7.25)

(51)Int.Cl.			F I		
H04L	9/32	(2006.01)	H04L	9/00	675A
H04L	9/08	(2006.01)	H04L	9/00	601C
G09C	1/00	(2006.01)	G09C	1/00	640D
			G09C	1/00	640E

請求項の数 14　(全 24 頁)

(21)出願番号　特願2012-531314(P2012-531314)
(86)(22)出願日　平成22年9月8日(2010.9.8)
(65)公表番号　特表2013-506369(P2013-506369A)
(43)公表日　平成25年2月21日(2013.2.21)
(86)国際出願番号　PCT/EP2010/063168
(87)国際公開番号　WO2011/039037
(87)国際公開日　平成23年4月7日(2011.4.7)
審査請求日　平成24年3月28日(2012.3.28)
(31)優先権主張番号　102009045133.1
(32)優先日　平成21年9月29日(2009.9.29)
(33)優先権主張国　ドイツ(DE)

前置審査

(73)特許権者　591245473
ロベルト・ボッシュ・ゲゼルシャフト・ミト・ベシュレンクテル・ハフツング
ROBERT BOSCH GMBH
ドイツ連邦共和国デー－７０４４２　シュトゥットガルト，ヴェルナー・シュトラーセ　１
(74)代理人　100140109
弁理士　小野　新次郎
(74)代理人　100075270
弁理士　小林　泰
(74)代理人　100101373
弁理士　竹内　茂雄
(74)代理人　100118902
弁理士　山本　修

最終頁に続く

(54)【発明の名称】センサデータの操作を防止するための方法及びこのためのセンサ

(57)【特許請求の範囲】
【請求項１】
センサ(A)のセンサデータ(x)の操作を防止するための方法であって、
センサ(A)の認証の枠内で一回だけ使用される数(R_B)が制御装置(B)から前記センサ(A)に送信され、該センサ(A)が、前記一回だけ使用される数(R_B)を使用して暗号学的な認証メッセージを生成し、暗号学的な認証メッセージの第１部分を前記制御装置(B)に送信し、前記センサデータ(x)に暗号学的な完全性保護が設けられ、前記センサデータ(x)に時変パラメータ(c)が追加され、暗号学的な完全性保護及び追加された前記時変パラメータ(c)を有するセンサデータ(x)が、前記センサ(A)から前記制御装置(B)に送信され、
前記時変パラメータ(c)を計算するために暗号学的な認証メッセージの第２部分を使用し、暗号学的な完全性保護を計算するために暗号学的な認証メッセージの第３部分を使用し、
前記暗号学的な認証メッセージの前記第１部分と前記暗号学的な認証メッセージの前記第２部分とが重ならないようにし、前記暗号学的な認証メッセージの前記第１部分と前記暗号学的な認証メッセージの前記第３部分とが重ならないようにすることを特徴とする方法。
【請求項２】
前記時変パラメータ(c)を、それぞれのセンサデータ-トランザクション時に変更する請求項１に記載の方法。
【請求項３】

前記時変パラメータ(c)の変更を段階的な増分に相当させる請求項２に記載の方法
【請求項４】
それぞれのセンサデータ-トランザクション時に暗号学的な認証メッセージの第２部分(R_{a0})から現在のパラメータ(R_a)を計算し、
現在のパラメータ(R_a)と、暗号学的な認証メッセージの前記第２部分(R_{a0})との差から、n回目のセンサデータ-トランザクションにおける時変パラメータ(c)を計算する請求項２に記載の方法。
【請求項５】
暗号学的な完全性保護の計算のために前記現在のパラメータ(R_a)を使用する請求項４に記載の方法。
【請求項６】
チャレンジ-レスポンス方式にしたがって前記センサ(A)の認証を行う請求項１乃至５のいずれか１項に記載の方法。
【請求項７】
メッセージ認証コード(MAC)方式にしたがって前記センサデータ(x)の暗号学的な完全性保護を実施する請求項１乃至６のいずれか１項に記載の方法。
【請求項８】
MAC方式として、OMAC方式又はEMAC方式を用いる請求項７に記載の方法。
【請求項９】
タイムスタンプ、シーケンスカウンタ又は乱数によって、前記時変パラメータ(c)を変更する請求項２に記載の方法。
【請求項１０】
センサ(A)の認証の枠内で、一回だけ使用される数(R_B)を制御装置(B)から受信し、前記一回だけ使用される数(R_B)を用いて暗号学的な認証メッセージを生成し、暗号学的な認証メッセージの第１部分を前記制御装置(B)に送信する手段と、
センサデータ(x)に暗号学的な完全性保護を設け、前記センサデータ(x)に時変パラメータ(c)を追加し、暗号学的な完全性保護及び追加された前記時変パラメータ(c)を有する前記センサデータ(x)を前記制御装置(B)に送る手段と
を備えるセンサ(A)であって、
該センサ(A)が、前記時変パラメータ(c)を計算するために暗号学的な認証メッセージの第２部分を使用し、暗号学的な完全性保護を計算するために暗号学的な認証メッセージの第３部分を使用し、
前記暗号学的な認証メッセージの前記第１部分と前記暗号学的な認証メッセージの前記第２部分とが重ならないようにし、前記暗号学的な認証メッセージの前記第１部分と前記暗号学的な認証メッセージの前記第３部分とが重ならないようにすることを特徴とするセンサ(A)。
【請求項１１】
前記センサ(A)が、センサデータ-トランザクション時に時変パラメータ(c)を変更する手段を備える請求項１０に記載のセンサ(A)。
【請求項１２】
前記時変パラメータ(c)の変更が段階的な増分に相当する請求項１１に記載のセンサ(A)。
【請求項１３】
前記現在のパラメータ(R_a)が、それぞれのセンサデータ-トランザクション時に暗号学的な認証メッセージの第２部分(R_{a0})から計算され、
n回目のセンサデータ-トランザクションにおける前記時変パラメータ(c)が、前記現在のパラメータ(R_a)と、前記暗号学的な認証メッセージの前記第２部分(R_{a0})との差から計算される請求項１１に記載のセンサ(A)。
【請求項１４】
センサ(A)の認証の枠内で、一回だけ使用される数（R_B）を生成し、前記センサ(A)に送

信し、該センサ(A)によって、前記一回だけ使用される数（R_B）を用いて生成した暗号学的な認証メッセージの第１部分を受信し、暗号学的な比較-認証メッセージを自ら生成し、受信した暗号学的な認証メッセージの一部を、前記自ら生成した暗号学的比較-認証メッセージによって評価する手段と、

暗号学的な完全性保護及び時変パラメータ(c)を設けたセンサデータ(x)を比較する手段と

を備える制御装置(B)であって、

該制御装置(B)が、前記時変パラメータ(c)を評価するために、自ら生成した暗号学的な比較-認証メッセージの第２部分使用し、暗号学的な完全性保護を評価するために、自ら生成した前記暗号学的な比較-認証メッセージの第３部分を使用する手段を備え、

前記暗号学的な認証メッセージの前記第１部分と前記生成した暗号学的な比較-認証メッセージの前記第２部分とが重ならないようにし、前記暗号学的な認証メッセージの前記第１部分と前記生成した暗号学的な比較-認証メッセージの前記第３部分とが重ならないようにすることを特徴とする制御装置(B)。

【発明の詳細な説明】

【技術分野】

【０００１】

本発明は、センサデータの操作を防止するための方法及びこのためのセンサに関する。

【背景技術】

【０００２】

出願人による公開されていない文献DE102009002396には、センサデータの操作を防止するための方法が記載されており、この方法は、センサの認証とセンサデータの完全性保護とを結びつけることによりトランザクション認証が得られ、これにより、センサデータの極めて高い操作防止性が確保されている点で優れている。

【先行技術文献】

【特許文献】

【０００３】

【特許文献１】DE102009002396

【発明の概要】

【発明が解決しようとする課題】

【０００４】

センサデータの操作を防止するための方法及びこのためのセンサを提供する。

【課題を解決するための手段】

【０００５】

本発明による方法又は本発明によるセンサは、例えば、サイドチャンネル攻撃による鍵の紛失時にも操作防止が解除されない、センサデータのトランザクション認証を達成することを可能にする。これにより、最小限の追加費用でセンサデータのためのさらに確固とした操作防止性が得られる。

【０００６】

さらなる利点及び改良形態が独立請求項の特徴により生じる。

【０００７】

有利な構成では、トランザクション認証のそれぞれの段階で、時変パラメータが変更される。このプロセスはシーケンスカウンタに相当し、認証及びトランザクション認証の方式部分を特に緊密に暗号学的に連携させ、ひいてはさらなる安全性を確保すること可能にする。時変パラメータの変更が段階的な増分に相当する場合、特に簡単に実施可能であり、特に有利である。

【０００８】

特別な実施形態は、認証方法の枠内では暗号学的な認証メッセージからの切捨てによって生成され、センサから制御装置に伝達される暗号学的な認証メッセージの第１部分と、時変パラメータを計算するために使用される暗号学的な認証メッセージの第２部分とが重

ならないということに基づいている。
【０００９】
時変パラメータが、トランザクション認証の各ステップで、暗号学的な認証メッセージから得られた初期パラメータと現在のパラメータとの差から計算され、現在のパラメータが初期パラメータからの段階的な増分により生じる場合、特に有利である。これによって、これまでの利点と比較して、認証方法から起こり得るレスポンス切捨てが弱められることもない。
【００１０】
好ましい実施例では、認証方式として特に安全性が高いことにより優れているチャレンジ・レスポンス方式が使用される。さらに、同様に高い安全性規格を満たすセンサデータの完全性保護のためにMAC方式を使用することも有利であり、極めて安全なEMAC方式及びOMAC方式は特に目的に適っている。
【００１１】
特別な実施形態では、時変パラメータはタイムスタンプ、シーケンスカウンタ又は乱数として構成されている。とりわけこれらの構成は、特に簡単かつ目的に適った形で本発明による方法が実施されるという利点をもたらす。
【００１２】
本発明による実施例を図示し、以下に詳細に説明する。図面は例示的なものにすぎず、一般的な発明思想を制限するものではない。
【図面の簡単な説明】
【００１３】
【図１】チャレンジ・レスポンス方式により制御装置Ｂでセンサ Ａを認証するためのシステムを示す概略図である。
【図２】センサデータの完全性保護のためのシステムを示す概略図である。
【図３】制御装置Ｂでセンサ Ａを認証するためのシステムを示す概略図である。
【図４】簡単な認証時の反射攻撃を説明する概略図である。
【図５】メッセージ認証コードEMAC又はCMACによるセンサデータの完全性保護を説明する概略図である。
【図６Ａ】センサデータのトランザクション安全性のための本発明による認証と完全性保護との結合又は認証とトランザクション認証との結合を説明する例示的なプロトコル全体を示す概略図である。
【図６Ｂ】センサデータのトランザクション安全性のための本発明による認証と完全性保護との結合又は認証とトランザクション認証との結合を説明する例示的なプロトコル全体を示す概略図である。
【発明を実施するための形態】
【００１４】
以下では、本発明を説明するために、RDSセンサ及び自動車における制御装置の実施例を適宜引き合いに出す。しかしながら、これにより、本発明をこの実施例に限定するものではない。なぜなら、説明する安全性コンセプト全体は一般に任意の制御装置とセンサとの間の通信の安全性確保のために設計されているからである。
【００１５】
通信路の安全確保に対する要求を考察する場合、主に次のこと：
（１）［メッセージ認証］データの完全性保護
（２）［実体認証］発信元の信憑性の確認
（３）［機密性］データの機密性保護（随意）
を考察すべきである。これらの基本的要求に加えて、基本的要求によって明確にカバーされていないが、その存在により暗黙に想定される特殊な攻撃タイプを困難にするか、又は防止するさらなる要求が導き出される。ここではとりわけ、
（ａ）［リプレイ防止］完全であるが（１）、もう既に一度送信されたデータのリプレイ防止、

（ｂ）［生きていること（liveliness）］発信元が伝達時点で「生きて」いたこと、すなわち、早い時点で事前に計算された（又は記録されたが送信されなかった）データが使用されなかったことの確認
が挙げられる。

【００１６】

以下では、データが機密ではない、すなわち、（３）が不要であることから出発する。これは、例えば車両分野のセンサの場合である。出願人のDE102009002396は、制限されたリソース（計算時間、通信容量）を考慮して（１）、（２）を（ａ）、（ｂ）と共に達成するためのプロトコルを内容としていた。この場合、基本思想は、完全性保護のためにメッセージ認証コード(MAC)を使用すること、対称的な暗号に基づいたチャレンジ・レスポンス・プロトコルをソース認証のために使用すること、及びトランザクション保護を達成するためにこれら２つの部分の組み合わせを使用することである。２つの個別のプロトコルは、それぞれ１つの固有の対称鍵(K_{mac}又はK_{auth})を使用できる。

【００１７】

次に、本発明によるトランザクション認証方法の実施例を詳述する。はじめに説明した安全性コンセプトの３つのステップは、DE102009002396により既知であるが、本発明による安全性コンセプトの決定的な第４ステップでは重要な相違が生じる。とりわけ図６に基づき説明する第４ステップは、各方法ステップの連携に基づいている。認証と完全性保護との確固たる結合を得るためには、２つのステップの乱数又は以下に説明するように２つのステップの時変パラメータと、乱数又はノンス（number only used once：一回だけ使用される数）とを結び付けることが提案される。認証及び完全性保護の２つの部分ステップをこのように結び付けることにより、２つの部分の緊密な暗号学的な連携が達成され、同時にセンサ側における乱数発生器の実装が省略される。さらに、ここで提案した方法又はセンサにより、例えばサイドチャンネル攻撃によるいずれか１つの鍵の紛失時にも操作防止が中断されないことが達成される。

【００１８】

図１は、チャレンジ・レスポンス方式によって制御装置ＢでセンサＡを認証するためのシステムの概略図を示す。この場合、制御装置Ｂは要求「チャレンジ」をセンサＡに送信し、センサＡは認証のために応答「レスポンス」で応答する。この場合の具体的な方法を概略的に示す。この方法は、高い安全性要求を満たし、上記タイプの攻撃ならびにさらなる安全性脅威及び損傷を暗号学的に確実に検出する包括的な安全性コンセプトの一部である。

【００１９】

第１ステップでは、制御装置ECUでRDSセンサを認証するためのチャレンジ・レスポンス方式を行う。認証の目的は、認証時点でRDSが有効な認証特性、すなわち、暗号鍵の知識が利用可能であるかどうかを確認し、この特性をアイデンティティ、例えばシリアルナンバーに確実に結び付けることである。（見出し：鍵の新しさ、生きていること）

図２は、センサデータの完全性保護システムを示す概略図である。安全性コンセプトのこの第２ステップでは、認証後、すなわち、データの受信相手、つまり制御装置（ECU）Ｂが、認証された送信元、すなわち、RDSセンサＡと通信していることがわかった後に、伝達するデータに完全性保護、有利には時変パラメータ、例えば乱数、シーケンス番号（シーケンスカウンタ）又はタイムスタンプが設けられる。

【００２０】

このことが図２ではセンサＡから制御装置Ｂに送信されたデータに時変パラメータを表す「タグ」をつけることにより示されている。このような時変パラメータは圧力データの処理後に暗号学的にチェックされ、RDSデータの操作が暗号学的に確実に検出される。すなわち、暗号学によるリアルタイムのデータ処理時にさらなる待ち時間が生じることがなく、これにより、操作を検出することができる。

【００２１】

暗号学的な完全性保護を達成するための効率的な方法として、いわゆる「メッセージ認

証コードMAC」が提案される。いわゆる「CBC-MAC方式」では、２つのさらなる暗号鍵を使用し、変更された別の提案、いわゆる「OMAC方式」では、１つのみの付加的な完全性鍵があれば十分である。デジタルPSI5インターフェースを有するRDSセンサのための実施オプションとして、EMACならびにOMACとしても知られているCMACが具体的に提案される。２つの方式は対称的なブロック暗号を使用し、鍵長k = 128 bit、 ブロック幅 n = 128 bit のAES-128 ("FIPS 197, Advanced Encryption Standard (AES), Federal Information Processing Standards Publication 197, November 2001, http://csrc.nist.gov/" 参照のこと）、又は代替的に、鍵長 k = 128、ブロック幅 n = 64 のPRESENT（"A. Bogdanov, L.R. Knudsen, G. Leander, C. Paar, A. Poschmann, M.J.B. Robshaw, Y. Seurin, and C. Vikkelsoe, PRESENT: An ultra-lightweight block cipher, in P. Pallier and I. Verbauwhede, editors, Proceedings of CHES 2007, volume 4727 of Lecture Notes in Computer Science, 450-467ページ, Springer-Verlag, 2007" 参照のこと）が提案される。PRESENTはまだ比較的新しいブロック暗号であり、AESほど円熟しておらず、論理的暗号分析攻撃に関する最初の報告もある。

【００２２】

安全性コンセプトの第３ステップでは、いわゆる「リプレイ攻撃」を防止するために、参照データ、すなわち、圧力値に対して付加的に、完全性保護のために保護すべきデータに乱数が追加される。これらの乱数は、平文でECUに伝達され、MACによって検証される。

【００２３】

認証は、エンジン始動直後、すなわち、最初の数分以内に行うことが望ましい。しかしながら、始動段階で実施することは不可欠ではない。完全性保護は、認証後にはじめて行われ、それ以前にこのような完全性保護を行っても暗号学的には意味がない。認証は比較的稀に、少なくともそれぞれのエンジン始動後及び複数のパケット又は同期性損失後に行われるが、圧力データには継続して、すなわち、例えばt = 16ブロック毎に完全性タグがつけられる。認証時にECUからセンサに伝達される乱数は、方法ステップを結びつけるために使用することができ、例えば完全性保護時の構成部分、シーケンスカウンタとして使用され、それぞれの(t = 16)×(n = 128)bitブロック毎に増分される。

【００２４】

完全性保護が標準的にそれぞれの圧力パッケージで使用される場合には、不都合であるが認証前にMACデータを無視するか、又はRDSセンサに提供された乱数を使用するべきである。認証が得られた後、調整されたシーケンスカウンタが使用される。

【００２５】

暗号学的安全性目的である送信元及びデータの認証、データの完全性、暗号鍵及びデータの新しさ及びリプレイ攻撃の防止が次の機構によって達成される：

チャレンジ・レスポンス方式による認証（アイデンティティと共にチャレンジの暗号化）；CBCモードに基づいたメッセージ認証コードによる完全性保護ならびに対応したパディング及びMAC強化、すなわち、EMAC及びCMAC；純粋なユーザデータの他に乱数の追加によるリプレイ攻撃の防止；完全性保護のためのシーケンスカウンタとしてチャレンジの一部を使用することにより２つのプロトコル部分を確実に暗号学的に結合；全ての機構のために唯一の基本構成、対称的なブロック暗号のみが必要となり、実施オプションとして、具体的には AES-128又は代替的にPRESENTが提案される；さらに制御装置でのみ２つのプロトコル部分を結合するため、乱数発生器が必要となる。

【００２６】

ＡはRDSセンサ、Ｂは制御装置ECUとする。目的は、Ｂに対するＡの認証である。すなわち、共通の秘密、暗証鍵を認証時点で使用できることをＡはＢに証明する。プロトコルにおける双方の参与者は３つの共通鍵を使用できる：

$K_{AB,Auth}$ - 共通の認証鍵

$K_{AB,MAC1}$ - 共通のメッセージ認証コード鍵１

$K_{AB,MAC2}$ - 共通のメッセージ認証コード鍵２。

【００２７】

CMACアルゴリズムの場合には、共通のメッセージ認証コード鍵１、$K_{AB,MAC1}$があれば十分である。Enc_Kを長さｋ及びブロック幅ｎの鍵Ｋを有する暗号化アルゴリズムとした場合、ほぼEnc_K=AES、k=長さ(K)=128、n=128である。代替的に、例えばEnc_K = PRESENT、k=長さ(K)=128、n=64が考慮される。さらにID_A及びID_Bを参与者のシステム全体に一義的なアイデンティティ、約32bitの長さのシリアルナンバー又はタイプ品番とする。シリアルナンバーの長さは暗号学的に重要ではない。エンジン／車両だけではなくシステム全体の全ての参与者を一義的に特定できることが重要である。以下に説明するように、このことは鍵管理において中心的な役割を果たす。

【００２８】

さらにこの例ではR_A又はR_BはA又はBによって生成された乱数であり、これらの適宜な長さは後に特定される。

【００２９】

以下にDE102009002396に記載の制御装置Ｂに対するセンサＡの例示的な認証を説明する。この場合にステップ１～３を図３に概略的に示す。

１．Ｂは64bitの乱数R_Bを生成する。

２．Ｂはこの乱数R_BをＢのアイデンティティID_Bと共にＡに送信する。

B->A : $R_B \| ID_B$ (64bit ‖ 32bit)

３．Ａはメッセージ$R_B \| ID_B$を暗号化し、すなわち、$Enc_{K\ AB,Auth}(R_B \| ID_B)$、これを$ID_A$と共にＢに戻す：

A->B : $Enc_{K\ AB,Auth}(R_B \| ID_B) \| ID_A$ (128bit ‖ 32bit)

４．Ｂは、２．のプロトコルステップ終了直後に、レスポンス$y=Enc_{K\ AB,Auth}(R_B \| ID_B)$の計算を開始する。３．における応答y'を得た後、Ｂはyを伝達された応答y'と比較する：

$y=y'=Enc_{K\ AB,Auth}(R_B \| ID_B)$といえるか？

さらにＢはＡのアイデンティティを所望のアイデンティティと比較する。すなわち：

$ID_A=ID'_A$といえるか？

２つの比較が成功した場合にのみ、ＡはＢに対して無事に認証されたといえる。

【００３０】

以下にステップ３及び４の適宜な変更を説明する。通信帯域を節約するために、ステップ３では最高値の64bitのみの暗号が伝送され、したがって、４．で比較される。これは64bitのブロック幅による暗号化方法を使用した場合と等価ではない。

３’．A->B : $msb_{127,\ldots,64}[Enc_{K\ AB,Auth}(R_B \| ID_B)] \| ID_A$ (64bit ‖ 32bit)

以下に認証１について方法全体を説明する。

１．Ｂは64bitの乱数R_Bを生成する。

２．Ｂはこの乱数R_BをＢのアイデンティティID_Bと共にＡに送信する。

B->A : $R_B \| ID_B$ (64bit ‖ 32bit)

３．Ａはメッセージ$R_B \| ID_B$を暗号化し、すなわち、$Enc_{K\ AB,Auth}(R_B \| ID_B)$とし、64の最上位ビット（最大値ビット）を$ID_A$と共にＢに戻す：

A->B : $msb_{127,\ldots,64}[Enc_{K\ AB,Auth}(R_B \| ID_B)] \| ID_A$ (64bit ‖ 32bit)

４．Ｂは２．のプロトコルステップ終了直後に、レスポンス$y=msb_{127,\ldots,64}[Enc_{K\ AB,Auth}(R_B \| ID_B)]$の計算を開始する。３．における応答y'を得た後、Ｂはyを伝達された応答y'と比較する。

$y=y'=msb_{127,\ldots,64}[Enc_{K\ AB,Auth}(R_B \| ID_B)]$といえるか？

さらにＢはＡのアイデンティティを所望のアイデンティティと比較する。すなわち：

$ID_A=ID'_A$といえるか？

２つの比較が成功した場合にのみ、ＡはＢに対して無事に認証されたといえる。

【００３１】

このために次のことに留意されたい。

(i)安全性コンセプトにおけるここで提案したステップは、実質的にISO/IEC9798-2 の２パス相互認証プロトコル(two-pass unilateral authentication protocol)である。これ

については、"ISO/IEC 9798-2, Information technology - Security techniques - Entity Authentication - Part 2: Mechanisms using Symmetric encipherment. algorithms ISO/IEC, 1994"を参照のこと。これは、Encの64最上位ビットのみの伝送の変更を除いてあてはまる。
(ii)リプレイ攻撃、すなわち、全ての通信記録及び後のプロトコル経過における復元を防止するために乱数R_Bがプロトコルで使用される。暗号学的には、「ノンス＝一回だけ使用される数(nonce = number used only once)」である必要がある。認証時のリプレイ攻撃を防止するためには、考察した使用シナリオでは64bitの長さで十分であるとみなされる。これよりも短い時変パラメータは過去に、例えばWLAN802.11b, KeeLoqにおけるWEP-Encyption で既に方法の攻撃及び／又は悪用可能な脆弱性につながっている。不可避的な認証のさらなる実施形態及び代替的な可能性を以下に説明する。制御装置Ｂにおいて乱数を生成するために、乱数発生器RNGが必要である。乱数発生器RNGは、決定的なRNGで特性K3レベルを有していることが望ましい。これについては、"Bundesamt fur Sicherheit in der Informationstechnik BSI, AIS 20: Funktionalitatsklassen und Evaluationsmethodologie fur deterministische Zufallszahlengeneratoren, Anwendungshinweise und Interpretationen zum Schema (AIS) Version 1, 2.12.1999, BSI, 2001, http://www.bsi.bund.de/zertifiz/zer1/interpr/ais20.pdfで参照可能"を参照されたい。

【００３２】

乱数発生器RNGは物理的なRNGにおいて特性P1レベル、好ましくはP2レベルを有していることが望ましい。これについては"Bundesamt fur Sicherheit in der Informationstechnik BSI, AIS 31: Funktionalitatsklassen und Evaluationsmethodologie fur physikalische Zufallszahlengeneratoren, Anwendungshinweise und Interpretationen zum Schema (AIS) Version 1, 25.9.2001, BSI, 2001, "http://www.bsi.bund.de/zertifiz/zer1 interpr/ais3i.pdfで参照可能" を参照されたい。これについては "W. Killmann and W. Schindler, Ein Vorschlag zu: Funktionalitatsklassen und Evaluationsmethodologie fur physikalische Zufallszahlengeneratoren, Technisches Papier zu AIS 31, Version 3.1, 25.9.2001, BSI, 2001, http://www.bsi.bund.de/zertifiz/zer1 interpr/trngkr31.pdfで参照可能" も参照されたい。

【００３３】

(iii)$|R_B|$ = 64の例示的な選択は、単純な２ステップ辞書攻撃に対する$2^{64/2}\times 2^{32}$対($R_B \| msb_{127\ldots 0}Enc_K(R_B)$)(64bit‖64bit)が同時に保存される場合に、$2^{33}=2^{64/2+1}$の通信セッションの攻撃複雑性をもたらす。これについては、例えば "Andrey Bogdanov and Christof Paar, On the Security and Efficiency of Real-World Lightweight Authentication Protocols, In Secure Component and System Identification, Marz 2007, Workshop Record of SECSI, Berlin, Marz 17-18, 2008" を参照されたい。RDSセンサの使用シナリオでは送信元に向けた通信（チャレンジ）は送信元から制御装置への通信（レスポンス）よりも数オーダだけ遅いことに留意されたい。64bitのレスポンス$msb_{127\ldots 0}Enc_K(R_B)$の単純な割合は$2^{64-1} = 2^{63}$の攻撃複雑性を有している。後に複数のセンサを使用することも計画されているので、アイデンティティID_A,ID_Bがシステムで必要とされる。これらは例えば32bitの値であると仮定されるが、大きな問題なしに64bitの長さとすることもできる。データパケット$R_B \| ID_B$が例えば80bitIDの選択により128bitよりも長くなることが望ましい場合、$Enc_K(R_B \| ID_B)$についてはより確実な操作モード（実施モード）、例えばランダムIV及びパディングによるCBC暗号化(CBC-Encryption with random IV and padding)が使用され、単純なECBモードはもはや使用されないことに注目すべきである。すなわち、解決は原理的には可能であるが、コスト高になる。

【００３４】

暗号学の見地からチャレンジの長さのためにより好ましい値は、例えば$|R_B|$ = 128である。この場合、スキャン攻撃（レスポンスの割合）も辞書攻撃も約2^{64}の攻撃複雑性を有する。この場合、メッセージブロックが128bitの限界を超過した場合には、より短い識別子IDは不可能である。

【００３５】
(iv)提案した方法は、「なりすまし」又は「並行セッション攻撃」と呼ばれることもある、いわゆる「反射攻撃」を効果的に防止する。アイデンティティIDのない単純化したプロトコル：
２．B -> A: R_B (64bit)
３．A -> B: $Enc_{K\ AB,Auth}(R_B)$ (128bit)
は、攻撃者Ｏが乱数R_Bをすぐに反射し、Ｂに送り返すことによって攻撃することができる。Ｂはこの並行した第２セッションに、計算した値$Enc_{K\ AB,Auth}(R_B)$で応答し、Ｏはこれを応答として使用する。図４は、単純な認証時のこのような反射攻撃の概略図を示す。
【００３６】
Ａ及びＢの挙動に関するさらなる暗黙の仮定がない場合、簡単な対抗措置は暗号化されたテキスト、すなわち$Enc_{K\ AB,Auth}(R_B \| ID_B)$にアイデンティティを関連付けることである。
【００３７】
(v)暗号化関数Encの代わりに、プロトコルで鍵付きハッシュ関数HMAC、一般のハッシュ関数、例えばSHA-256又はメッセージ認証コード(MAC)、例えばCBC-MACを使用することができる。しかしながらこれらの実施オプションはいずれもより高いコストにつながる。
【００３８】
(vi)チャレンジの認証、すなわち、B -> A: R_B, $MAC(R_B)$は、この実施例ではコストを理由に、(iii)の見解を考慮して省略される。
【００３９】
(vii) ここで例示的に提案したプロトコルは、いわゆる「ノンスを有する２パス一方向認証プロトコル(two-pass unilateral authentication protocol with nonces)」である。暗号学では、常に最も少数のプロトコルパスを有し、それにもかかわらず高い安全性特性を有するプロトコルを探す。一般に、ノンス、乱数の代わりに、タイムスタンプ又は同期カウンタ、シーケンス番号（シーケンスカウンタ）を用いた場合、１つのプロトコルパスで済むことは少ないということができる。タイムスタンプ又はカウンタは、特にＡ、検査装置によって設定又は設定解除してはならない。この場合、全てのプロトコル動作はＡから生じ、したがってＡにより記録することができる。
【００４０】
すなわち、決定的な点は、ＡからＢへの一方向通信しか行うことができず、システム内に同期的に、操作不能に提供されている必要があるタイムスタンプもシーケンスカウンタも使用することができない場合、全てのプロトコル動作はＡにより開始され、Ｂは受動的であり、証明器／要求者Ａは論理的にはそれまでの全てのデータを記録し、入力することができる。したがって、戻り方向の経路を省略し、それにもかかわらず、要求される安全性を例えば暗黙的認証により達成することは原理的に不可能である。
【００４１】
これについては "Walter Fumy and Hans-Peter Riess, Kryptographie, Entwurf; Einsatz und Analyse symmetrischer Kryptoverfahren, R. Oldenbourg Verlag, Munchen, Wien, second edition, 1994" を参照されたい。ここでは、タイムスタンプを使用するには、十分に正確で信頼できるシステム時間を使用できることが必要であることが教示されている。この場合、検証権威者は、割り当てられたタイムスタンプに基づいてメッセージの有効性をチェックすることができる。メッセージ入力時点とタイムスタンプとの間のずれが閾値を上回っていない場合にはメッセージが受信される。許容時間間隔内でもトランザクションの再入力を防止することが望ましい場合、受信したメッセージについて適宜な長さの記録をつける必要がある。これは、もちろんセンサコンテキストでは不可能である。タイムスタンプを実際に使用した場合にある種の困難性が生じるのは、権威者におけるクロックの同期である；しかしながら、タイムスタンプに欠如していることの多い操作安全性が問題となることもある。シーケンス番号の基本思想は、それぞれのシーケンス番号に対する認証機構では１つのメッセージのみが受信されるか、又は所定の論理時間内に１つ

のメッセージのみが受信されることにある。シーケンス番号を使用する場合、それぞれの対応した通信関係について生じるある程度の管理コストが必要となる。最小限の要求は、それぞれの通信方向についてそれぞれ現在のシーケンス番号を記憶することである。例えば、システム故障の場合のために同期機構を設ける必要もある。

【００４２】

暗号学テキストからのこのような引用に基づき、認証時には何らかの形態の双方向通信又は同期的通信を放棄できないことがわかる。このことは、PSI5バスを介したセンサ・制御装置通信の実施例に関してECUとセンサとの間の同期的な双方向のPSI5通信を放棄できるか否かという問いを一義的に否定する。

【００４３】

(viii)完全性のために、さらに対応したISO/IEC 9798-2による "one-pass unilateral authentication protocol with time stamps / sequence counters（タイムスタンプ／シーケンスカウンタを有する１パス一方向認証プロトコル）" を指摘しておく。これについては、"ISO/IEC 9798-2, Information technology - Security techniques - Entity Authentication - Part 2: Mechanisms using symmetric encipherment algorithms ISO/1 EC, 1994" を参照されたい。

$A \rightarrow B: Enc_{KAB,\ Auth}(TS_A \| ID_B) \| ID_A$

TS_Aは、Aによって生成されるタイムスタンプである。平文で伝送されるID_Aは、直接的な暗号学的意味を有さず、ISO標準には入っていない。

【００４４】

以下に、暗号学的プロトコルのための設計原理を詳述する。暗号学的プロトコルを設計するためには、留意すべき幾つかの原理がある。これについては "Colin Boyd and Anish Mathuria, Protocols for Authentication and Key Establishment, Springer, 2003, 31ページ" を参照されたい。

【００４５】

この文献に示されたプロトコルはプロトコル3.4及び3.5として見られる。77及び78ページを参照されたい。記号$\{\}_K$が鍵Kによって認証された暗号を表す。

$A \rightarrow B: \{T_A, B\}_{K\ AB}$

Protocol 3.4: ISO/IEC 9798-2 １パス一方向認証プロトコル(one-pass unilateral authentication protocol)

1. $B \rightarrow A: N_B$
2. $A \rightarrow B: \{N_A, B\}K_{AB}$

Protocol 3.4: ISO/IEC 9798-2 ２パス一方向認証プロトコル(two-pass unilateral authentication protocol)

文献の80ページに見られる表3.2から、２つのプロトコル3.4及び3.5は望ましい特性、生きていること、鍵の新しさ、Aの実体認証を満たしており、形式的な安全性証明が存在しないにもかかわらず既知の攻撃が存在しないことがわかる。２つのプロトコルの安全性は、過去に既に調査された。プロトコルの形式的な安全性証明は、最近になってようやく規格に対する厳しい要求となっている。生じたプロトコルは技術的理由から古典的な認証プロトコルよりも幾分包括的である。以下に例示的に選択したプロトコル3.5に準拠するプロトコルは、そこに示された全てのプロトコルのうちで最も好ましく、このプロトコルに対してはこれまで攻撃が存在しない。

【００４６】

以下に、センサデータの完全性保護、安全性コンセプト全体の第２ステップを詳述する。センサの交換を防止することの他に、センサデータの変更を防止すること、又はより正確には発見すること及び証明することは極めて有意義である。データを保護するための暗号技術はデータの発信元、すなわち、データ発信元認証(data origin authentication), 送信者の完全性(integrity of sender)及びデータの完全性(integrity of data)、すなわち、データが変更されていないという事実に焦点を合わせる。他の重要な点は、タイムリーであること、データ、場合によっては意図する受信者の順序である。これらの特性は、

著しくアプリケーションに依存している。例えばパッケージ紛失時に、変更されていないデータのリプレイが暗号学的な攻撃であるか否かという問いがなされる。したがって、暗号学的な基本機構自体には時間に依存したステータス（状態）は組み込まれない。

【００４７】

以下では上記符号を用いる。したがって、$x = (x_1, \cdots, x_t)$であり、tは、場合によってはパディング実施後におけるブロック幅nのメッセージテキストブロックである。データの認証又は完全性の目的を達成するために、実質的に３つの暗号学的アクセスを区別することができる：認証コード、デジタル署名及びメッセージ認証コードである。

【００４８】

認証コードは、攻撃に対する成功可能性を明確に算出可能であり、かつ攻撃者の計算能力とは無関係な複合オブジェクトである。認証コードは今日では比較的わずかにしか広まっておらず、それぞれのトランザクションのために新しい鍵が必要となることが多い。

【００４９】

デジタル署名は、対称方式よりも係数100～1000だけ遅い非対称方式である。非対称方式の場合、それぞれのセンサ／参与者Ａは固有の鍵ペアpk_A, sk_A（公開鍵、秘密鍵／非公開鍵）を使用することができる。署名時に参与者Ａはメッセージxのハッシュ値を計算し、非公開鍵sk_Aによる署名生成方式GenSigを用い、得られた署名sをＢに送信する。すなわち：

A -> B: x, s = $GenSig_{skA}(h(x))$

B: $VerSig_{pkA}(x', s)$といえるか？　　承諾又は拒否。

Ｂはメッセージx’及び署名sを受け取り、署名検証方法VerSigによってＡの認証公開鍵を使用して署名をチェックする。現在では、例えば自動車分野では多くの場合RSA PKCS#1v1.5署名を用いる：

1. RSA 1024 bit: GenSig: 1024 $bit^{1024bit}$, VerSig: 1024 bit^{16bit}; 署名、モジュール、秘密鍵がそれぞれ 128 byte、公開鍵 ～ 16 bit
2. RSA 2048 bit: GenSig: 2048 $bit^{2048bit}$, VerSig: 2048 bit^{16bit}; 署名、モジュール、秘密鍵がそれぞれ 256 byte、公開鍵 ～ 16 bit。

ECC(Elliptic Curve Cryptography:楕円曲線暗号)署名は実質的により良好な特性を有している。

3. ECDSA 160 bit - RSA 1024 bitと比較可能な安全性: GenSig: スカラー小数点乗算、VerSig: ２つのスカラー小数点乗算; 署名40 byte。
4. ECDSA 256 bit - RSA 2048 bitと比較可能な安全性: GenSig: スカラー小数点乗算、VerSig: ２つのスカラー小数点乗算; 署名80 byte。

【００５０】

デジタル署名は、完全性保護に対して付加的に、通信相手間の論争において法的に役立ち得る否認防止特性(Non Repudiation Property)を提供する。例示的に説明したセンサコンテキストでは２つの通信相手はシステムである自動車の内部で作動し（エンジン又は制御装置Ｂならびにセンサ Ａ）、制御装置Ｂがセンサ Ａから得たメッセージを否認することは稀なので、否認防止特性はこの具体例では不可欠ではない。

【００５１】

以下にメッセージ認証コードを説明する。既に考察したように、タグ、例えば線形タグ又はハッシュダイジェストの単純な暗号化は、メッセージ認証の問題を解決しない。メッセージ認証コードは：

1. 鍵生成アルゴリズム: 出力: 対称鍵 K、 ビット長さ k、例えば k = 56 ... 256、
2. MAC-生成アルゴリズム: 入力: テキスト x、 鍵K, 出力: MAC_K (x) - 長さmのビット列、例えば m = 24 ... 128、
3. MAC-証明アルゴリズム: 入力: テキスト x, MAC-値 $MAC_K(x)$, 鍵 K, 出力: 承諾又は拒否。

【００５２】

MAC-アルゴリズムの安全性は、以下のように簡単に説明することができ、一般に受け入

れられた安全性モデルである：攻撃者は、いわゆる「適応的選択文書攻撃 adaptive chosen text attack」によって、いわゆる「存在的偽造（existential forgery）」、すなわち、何らかの許容可能な偽造を生成することは実際には不可能（計算時間の面から見て実行不可能：computationally infeasible）である。これは特にMAC-認証問い合わせも含む。例えば"Jonathan Kahtz and Yehuda Lindeil, Introduction to Modern Cryptography, CRC Publishing, 2007, 16ページ" を参照のこと。メッセージ認証コードがこの意味で確実な場合、安全性は特にデータの特殊な暗号化に依存していない。

【００５３】

以下はMACアルゴリズムに対する４つの典型的な攻撃である：

1.ブルートフォース鍵探索 -> 十分に大きい鍵空間、例えば中期安全性(medium term security)ではk $\geq$ 80。

2.MAC値の推測 -> 成功可能性 max (2^{-k}, 2^{-m})。多くのアプリケーションにはk>m及びm=32…64が成り立つ。

3.内部衝突に基づいた一般的偽装。これは特に上記変更して提案された「完全性保護方法」の場合である。

4.暗号分析の弱点に基づいた攻撃。

【００５４】

最後の２つの攻撃シナリオを説明するためには "Bart Preneel, MAC algorithms, In van Tilborg"が引用され、これについては "Henk C. A. van Tilborg, editor, Encyclopedia of Cryptography and Security, Springer, 2005", 365ページ" を参照されたい。

【００５５】

ブロック暗号に基づいたMACと専用のハッシュ関数に基づいたMACとの間でMACを区別する。少し前に国際団体で、MACをいわゆる「ユニバーサルハッシュ関数」によって標準化する作業が始まった。ISO/IEC 9797-3を参照されたい。これらはまだ広まっていない。

【００５６】

以下に鍵付きハッシュ関数HMACを説明する。鍵付きハッシュ関数の中でもHMAC構造は最も広く知られている。"Mihir Bellare, Ran Canetti and Hugo Krawczyk, Keying hash functions for message authentication, In Neal Koblitz, editor, Proceedings of CRYPTO '96, volume 1109 of Lecture Notes in Computer Science, 1-15ページ, Springer, 1996" を参照されたい。HMAC構造は特にIETF環境、例えばIPsec及びhttp-digestで極めて頻繁に使用される。HMACは、適切なハッシュ関数Hを有する秘密鍵Kを用いてメッセージxを２回ネストしたハッシュに基づいている。

【００５７】

【数１】

$$HMAC_K(x) := H(K \oplus opad \,||\, H(K \oplus ipad \,||\, x))$$

【００５８】

２つの文字列ipad, inner-padding及びopad, outer-paddingは規定された文字列定数である。ブロック暗号に基づいたMACとは反対に、HMAC構造では多数のパディング方式及び随意のプロセスは存在しない。

【００５９】

例えばHMAC-SHA-1、H=SHA-1、m=96、すなわち、160 bit出力のうち96最上位ビットのみ、及びHMAC-SHA-256, H=SHA-256, m = 96, すなわち256 bit出力のうち96最上位ビットのみ、が広く知られている（例えばIPSecを参照）。HMACはセンサ保護の枠内でも考察されるが、最後に具体的な実施例で提案した解決策よりも著しく高価である。

【００６０】

次にブロック暗号に基づいたMACを説明する。x_1, x_2….x_t , を、例えばアルゴリズムEnc=AESのためのブロック長さn=128又はアルゴリズムEnc=DES, Triple DES又はPRESENTのためのブロック長さn=64を有するメッセージテキストのブロックとすると、

$n = |x_1| = \cdots = |x_{t'-1}|$

となり、t’によっていわゆる「パディング」前のブロック数が示され、最後のブロックは、場合によってはnビットよりも少ないビットを有し、すなわち、$0<|x_{t'}| \leq n$である。パディング後にtメッセージテキストブロック$x_1, \cdots, x_t$を受け取り、パディング方式に応じて、以下のことがいえる。

【００６１】

パディング方式１：ブロック長さnの倍数が得られるまで、平文が0bitにより満たされないか、又は複数の0bitによって満たされる。すなわち、t=t'である。これについては "A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, Handbook of Applied Cryptography, CRC Press, 1996, Algorithm 9.29, 334ページ" を参照のこと。

【００６２】

パディング方式２：平文に1bitが追加され、次いでブロック長さnの倍数が得られるまで0bitで満たされる。すなわち、t=t'又はt=t'+1である。これについては"A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, Handbook of Applied Cryptography, CRC Press, 1996, Algorithm 9.30, 334-335ページ" を参照されたい。

【００６３】

パディング方式３：長さブロックＬ、多くの場合64bitで、もとのメッセージの長さをコード化し、場合によっては左揃え0bitで満たす。ブロック長さｎの倍数が得られるまでもとのメッセージを0bitで満たさないか、又は複数の0bitで満たし、長さブロックＬを追加する。すなわち、t=t'又はt=t'+1である。

【００６４】

パディング方式３はメッセージ認証コードに関する規格の1999年版（"ISO/IEC 9797-1, Information technology - Security techniques - Message Authentication Codes (MAC$_S$) Part 1: Mechanisms using a block cipher, ISO/I EC, 1999"を参照されたい。）ではじめて補足され、それ以前の版（"ISO/IPC 9797, Information technology - Security techniques - Data integrity mechanisms using a cryptographic check function employing block cipher algorithm, ISO/IEC, 1994"を参照されたい。）には含まれていない。このパディング方式はハッシュ関数においても使用される。

【００６５】

第１のパディング方式は方式２及び方式３よりも安全特性が劣っている。

【００６６】

簡略化のために、このメッセージテキストデータには、いわゆる「リプレイ攻撃」から保護するために使用される付加的なランダム化データが既に含まれていると仮定され、これについて以下にさらに説明する。

【００６７】

したがって、$x_1, .., x_t$は、パディングされ、場合によってはランダム化データを補足された圧力値、メッセージテキストであるとする。RDSセンサ保護の具体的な実施例では、例えば現在具体的に次のように計画されている：

Enc_K - 鍵Kによる暗号化アルゴリズムAES

k- bitで示す鍵Kの長さ、ここではk = 128

n- 暗号化アルゴリズムのブロック幅、ここではn = 128

t- ブロック数、ここではt = 16。

【００６８】

ブロック暗号に基づいたメッセージ認証コードでは、CBC-MAC構造が最も広まっている。IVを長さnの初期ベクトルとする。CBC-MACではCBC-暗号化とは反対に、値$IV=0^n$を選択してもよい。

$H_0 := IV = 0^n$ 初期化

【００６９】

【数２】

$$Hi := Enc_K(H_{i-1} \oplus x_i)\ i = 1, \ldots, t$$

【００７０】

CBCモード

$MAC_K(x) := g(H_t)$ 出力変換。

【００７１】

以下のことに留意されたい：

(i)規格"ISO/IPC 9797, Information technology - Security techniques - Data integrity mechanisms using a cryptographic check function employing block cipher algorithm, ISO/IEC, 1994"の付録Aにはさらに２つの任意プロセスが定義されている。

【００７２】

任意プロセス１：第１の任意プロセスでは、第２の独立した鍵K_2によって、最後のブロックH_tがもう一度独立して処理される：最後のブロックH_tは、K_2によって復号され、K_1によって暗号化される。これは、$g(H_t) := Enc_{K1}(Dec_{K2}(H_t))$に相当する。このプロセスは、MAC補強と呼ばれ、選択文書存在的偽造(chosen-text existential forgery)攻撃から防御するために役立つ。

【００７３】

任意プロセス２：第２の任意プロセスは、第２の鍵K_2により最終ブロックを再度暗号化することを含む。すなわち、

$MAC := g(H_t) = Enc_{K2}(H_t)$。

【００７４】

(ii)新しい規格（"ISO/IEC 9797-1, Information technology - Security techniques - Message Authentication Codes (MAC_S) Part 1: Mechanisms using a block cipher, ISO/IEC, 1999"を参照されたい。）は、随意のプロセスをもはや含んでおらず、その代わりに６つの異なるMACアルゴリズムを特定しており、これらのアルゴリズムはそれぞれ３つの異なるパディング方式と組み合わせることができる。MACアルゴリズム４～５は新しいパターンである。そこに提案されている方式のうち２つは、2003年の新しい暗号学的知識によりもはや適宜なものとはみなされない。実証された古いアルゴリズムは以下の通りである。

【００７５】

1.任意プロセスなしのCBC-MAC：通常のCBC-MAC、すなわち、$g(H_t) = H_t$である。Bellare, Kilian 及び Rogawayは、基本的なブロック暗号が安全であれば、すなわち、疑似ランダム関数(Pseudo Random Function)であれば、CBC-MACは固定長のメッセージでは安全であることを2000年に示すことができた。（"M. Bellare, J. Kilian, and F.:'Rogaway, The security of cipher block chaining, Journal of Computer and System Sciences, 3(61):362-399, 2000"を参照のこと。）即時攻撃は可変長のメッセージ、例えば

【００７６】

【数３】

$$MAC_K(x \,||\, (x \oplus MAC_K(x))) = MAC_K(x)$$

【００７７】

で生じることに留意されたい。したがって、このような簡単なCBC-MACは実際には使用されない。

【００７８】

2.任意プロセス１によるCBC-MAC：リテール-MAC。$g(Ht) = msb_{63\ldots32}\ Enc_{K1}(Dec_{K2}(Ht))$を有するこのMACは、リテール-バンキング分野でDESのためにはじめて使用されたのでこのように呼ばれ、m = n/2 = 32, n = 64, Enc - DES, K_1及びK_2についてk = 56である

。最後のn-bitブロックg(Ht) 全体がMACとして使用されずに、左揃えのビットのみが使用されるようにCBC-MACを変更した場合、全数鍵攻撃(exhaustive key attack) に対する安全確保に役立つ。実際には、m = n/2である場合が多い。センサデータ完全性のためのこのようなCBC-MAC方式の使用は、2008年10月29日にRDSセンサワークショップで初めて提案された。DESを暗号化アルゴリズムとして使用する場合、ソフトウェアにおいてもハードウェアにおいても暗号化のみを用いるのか、又は暗号化及び復号化を用いるのかで実施の手間に大差はない。しかしながら、AESでは、暗号化に比べて復号化にかかる付帯コストは著しく大きくなる。そこで、次の会合では暗号化のみで済む次の方法が提案される。

【００７９】

3.任意プロセス２によるCBC-MAC：EMAC:この図式は第２鍵K_2による出力変換：

【００８０】

【数４】

$$g(H_t) = Enc_{K2}(H_t) = Enc_{K2}(Enc_{K1}(H_{t-1} \oplus x_t))$$

【００８１】

として使用される。EMAC構造は、初めにRIPE組合によって1995年に提案された。Petrank/Rackoff（"E. Petrank and C. Rackoff, CBC MAC for real-time data sources, Journal of Cryptology, 3(13):315-338, 2000"を参照のこと）は、可変長の入力時にCBC-MACの安全性を証明することにこの図式ではじめて成功した。この図式は、センサ安全確保のための実施オプションとして提案され、具体的にはEnc - AES, 鍵K_1及びK_2についてk = 128、n = 128、m= m/2 = 54である。

【００８２】

(iii)BlackとRogawayによるさらなる最適化はパディングによる付帯コストを低減する。XCBC又は３鍵MAC(Three-key MAC) （"J. Black and P. Rogawa, CBC MACs for arbitrary-length messages, In Mihir Bellare, editor, Advances in Cryptology - CRYPTO 2000, number 1880 in Lecture Notes in Computer Science, 197-215ページ, Springer-Verlag, 2000"を参照のこと）はk bitの鍵K_1をブロック暗号のために使用し、２つのn bitの鍵K_2及びK_3をいわゆる「鍵ホワイトニング」のために用いる（XOR-Encrypt-XOR アプローチの基本思想に関する"T. Schutze, Algorithmen fur eine Crypto-Library fur Embedded Systems, Technical report, Robert Bosch GmbH, CR/AEA, August 2007, Internal document, Version 1.0, 2007-08-07, 46 pages"を参照のこと）。XCBC-MACは最後の暗号化及びパディングを、パディング前後のブロック数が同じ、すなわち、t = t'となるように修正する:

$|x_{t'}| = n$ の場合、

【００８３】

【数５】

$$x_t = x_{t'} \oplus K_2.$$

【００８４】

となる。別の場合には、'1' - bit 及びj = n - $|x_{t'}|$-1'0'- bitsを追加し、

【００８５】

【数６】

$$x_t = (x_{t'} \| 10^j) \oplus K_3.$$

【００８６】

となる。

【００８７】

(iv) もちろん、もう１つの別の鍵K_3はそれほど有利ではない。IwataとKurosawaによるOMACアルゴリズム（"T. Iwata and K. Kurosawa. OMAC: One key CBC MAC, In T. Johann

son, editor, Fast Software Encryption, number 2887 in Lecture Notes in Computer Science, 129-153ページ, Springer-Verlag, 2003"を参照のこと）は、特にK_2 = ′2′ × $EnC_{K1}(0^n)$及びK3 = ′4′ × $Enc_{K1}(0^n)$を選択することによって鍵にかかるコストを低減している。

【００８８】

′2′及び′4′はガロア体GF(2^n)の２つの素子であり、×は体の乗算を示すことに言及しておく。

【００８９】

NISTがこのアルゴリズムをCMACの名称で標準化することが期待される。これについては "Morris Dworkin, Cipher Modes of Operation: The CMAC Mode for Authentication, NIST Special Publication 800-38b, National Institute of Standards and Technology NIST, May 2005, http: //csrc.nist.gov/publications/nistpubs/800-38b/sp800-38b.pdf, 2- 3ページ" を参照されたい。

【００９０】

この実施例では、このアルゴリズムOMAC又はCMACをセンサ安全確保のためのメッセージ認証コードとしてEMACと並行して調査することが提案される。

(v) CBC暗号化では、当然ながら全ての暗号化されたブロックが出力され、CBC-MACでは最後のブロックのみが出力される。このことは技術的理由からのみではない。全てのH_iを出力するCBC-MAC方式は不確実だからである（"Jonathan Kahtz and Yehuda Lindeil, Introduction to Modern Cryptography, CRC Publishing, 2007, 126ページを参照のこと）。

【００９１】

次に、提案された全体性保護方法を説明し、要約する。これまでのように、x = (x_1,… , x_t ,)はブロック長さnのメッセージテキスト、圧力値のブロックを示し、Enc = {AES, PRESENT}は、対称的なブロック暗号、すなわち、k = 128及びn = 128又はn = 64を示す。パディング後、パディング方式１又は２が選択され、t個のメッセージテキストブロックx = (x_1,… , x_t)を得る。具体的な数を表すことができるように、次のようにRDSセンサで現在計画された実施形態が選択される：

t = 16 ブロック

Enc AES, k = 128, n = 128

パディング方式１。

以下にEMACを説明する：

２つの 128 bit MAC-鍵$K_{AB,MAC1}$ 及び$K_{AB,MAC2}$、m =n/2 = 64, すなわち、MAC 出力の切り捨て、及び出力変換$Enc_{KAB,MAC2}$ (H_t)により、

H_0 := IV = 0^{128} 初期化

【００９２】

【数７】

$$H_i := Enc_{K\,AB,MAC1}(H_{i-1} \oplus x_i)\ i = 1, \ldots, t$$

【００９３】

CBC モード

$MAC_K(x)$:= $msb_{127...64}$ [$Enc_{K\ AB,MAC2}$ (Ht)] 出力変換

となる。

【００９４】

現在考察した変化態様では、ビット数2048はブロック長さ128によって分割可能である。すなわち、パディング方式１はいずれのパディングとも等価ではない。この安全性コンセプトを一般的に使用する場合、パディング方式２が緊急に推奨され、さもなければ、この方式は実際に固定長の入力についてのみ使用すべきである。EMAC方式のための安全性証明は "E. Petrank and C. Rackoff, CBC MAC for realtime data sources, Journal of Cryptology, 3(13):315-338, 2000" に見られる。すなわち、可変入力長さに対する適応的

選択文書における存在的偽造は不可能である（"existentially unforgeable under adaptive chosen-message attacks for variable input length"）。この方式は、"ISO/IEC 9797-2, Information technology - Security techniques - Message Authentication Codes MACs - Part 2: Mechanisms using a dedicated hash function, ISO/1 EC; 2002"で標準化されている。手間は、実質的にブロック暗号のt+1回の呼び出しにかかる。

【００９５】

次にCMAC/OMACを考察する。K又は$K_{AB,MAC1}$が128bitのMAC鍵であるとすれば、K_1, K_2は導かれた128bitの２つの鍵を示す。さらに通常のように$x_1,\cdots,x_{t-1}$, x'_tはブロック長さn(AESについてn = 128) のメッセージテキストのブロックである。CMAC方式では、t' = tである。すなわち、最終ブロックの後処理（「パディング」）前後のブロック数は等しく、$|x_t| = |x'_t|$が成り立つ。すなわち、最終ブロックのビット数は等しいが、$x'_t = x_t$は成り立たない。最終ブロックは、むしろXCBC構造によって修正される。

【００９６】

次にCMACにおけるサブ鍵生成のためのアルゴリズムを説明する。

入力：鍵K、出力：サブ鍵K_1, K_2

S1: $L := Enc_K(0^{128})$;

S2: (rnsb(L) = 0)ならば

【００９７】

【数８】

$$K_1 := (L \ll 1)$$

【００９８】

さもなければ

【００９９】

【数９】

$$K1 := (L \ll 1) \oplus R_b;$$

【０１００】

S3: ($msb(K_1) = 0$)ならば

【０１０１】

【数１０】

$$K2 := (K1 \ll 1)$$

【０１０２】

さもなければ

【０１０３】

【数１１】

$$K_2 := (K_1 \ll 1) \oplus Rb;$$

【０１０４】

ブロック幅128のブロック暗号、すなわち、例えばAESでは定数$R_b := 0\times0\cdots087$となる。64bitのブロック長さ暗号、例えばTripleDESでは$R_B := 0\times0\cdots01B$となる。$|x'_t|=n$、すなわち、最後のブロック$|x'_t|$が完全な場合、

【０１０５】

【数１２】

$$xt := K_1 \oplus x'_t,$$

【０１０６】

さもなければ

【０１０７】
【数１３】

$$xt := K_2 \oplus (x'_t \| 10^j)$$

【０１０８】
を設定し、j = n-| x ’ t|-1とする。最終ブロックにおいてのみオリジナルデータに対して変更したこのようなメッセージブロックx1,… ,xtにより、いまや規格-CBC-MACアルゴリズムが実施され、必要に応じて出力は64bitまで分割することができ：
$H_0 := IV = 0^{128}$ 初期化
【０１０９】
【数１４】

$$H_i := Enc_{K\,AB,MAC1}(H_{i-1} \oplus x_i)\ i = 1, ..., t$$

【０１１０】
CBCモード
$MAC_K(x) := msb_{127...64}(Ht)$ 出力変換
となる。
【０１１１】
CMAC方式では、EMACとは反対に、ブロック長さによって分割不可能なビット数の場合でさえもさらにブロックを追加する必要はない。128bitのMAC鍵$K_{AB,MAC1}$のみが必要である。この方式の説明及び安全性については、"T. Iwata and K. Kurosawa, OMAC: One key CBC MAC, In T. Johannson, editor, Fast Software Encryption, number 2887 in Lecture Notes in Computer Science, 129-153ページ, SpringerVerlag, 2003" 及び "Morris Dworkin, Cipher Modes of Operation: The CMAC Mode for Authentication, NIST Special Publication 800-38b, National Institute of Standards and Technology (NIST), May 2005、さらにhttp://csrc.nist.gov/publications/nistpubs/800-38b/sp800-38b.pdf" を参照されたい。この方式は現在NISTで標準化されている。手間は、ブロック暗号のt回の呼び出し、$Enc_K('0')$の一回のみあらかじめ計算すること、ならびにGF(2^{128})又はシフト操作にある。
【０１１２】
CMAC方式を実際にEMACよりも速く、とりわけスペース又は記憶場所を節約して実現することができるのか否かは、CMACにおけるブロック暗号及び複合的な付帯コストの実施に関係している。詳細は、様々な実施テスト後にはじめて報告することができる。MAC計算又は検証はセンサ側（Ａ）及び制御装置側（Ｂ）で実質的に同時に行われる。
【０１１３】
図５は、メッセージ認証コードEMAC又はCMACによるセンサデータの完全性保護を説明するための概略図を示す：
A: それぞれ長さnのt個の入力ブロックx=(x1,…,x_t)から64bitのMAC値を共通の鍵K = $K_{AB, MAC}$、EMACでは$K_{AB, MAC} = K_{AB, MAC1}$, $K_{AB, MAC2}$ 又はCMACでは$K_{AB, MAC} = K_{AB, MAC1}$ のみによって計算せよ：
【０１１４】
【数１５】

$$MAC_K(x) := \begin{cases} EMAC_K(x) \\ CMAC_K(x) \end{cases}$$

【０１１５】
MAC値をメッセージテキストx = (x_1, …, x_t)と一緒にBに送信せよ。
A -> B: $x_{1,...,}x_t$, $MAC_K(x)$ txn bits, m bits = 2048 bits, 64 bits

B:メッセージテキストx_1,…, x_tを平文で受け取り、これをすぐにさらなる処理のために遅延なしに転送し、$MAC_K(x)$を計算し、次いで伝送されたAのMAC値と比較する。等しくない場合にはエラー信号が生成され、エラー信号は適宜処理される。

【０１１６】

次に、安全性コンセプト全体のさらなる部分として、リプレイ攻撃を防止するための方法を説明する。メッセージ認証コードは常にデータの意味論とは無関係であることが望ましいので、好ましくはMACの構造には時間に関係した状態は組み込まれない。したがって、データ$x = (x_1, \cdots, x_t)$が上記プロトコルで純粋な圧力データあった場合、攻撃者／チューナーは、自分にとって関心のあるエンジン状態又は圧力経過（フルスロットル）を記録し、後に再入力（リプレイ）する。このシナリオを防止するために、データに加えて偶発的な要素（乱数、時変パラメータ）などがメッセージ内容に挿入される。すなわち、完全性保護方法、MAC自体は変更されないままである。記録された圧力データを備える局所的データバンクの可能性について効率を考慮し、推測するために、例えば、少なくともr = 32 bit の任意データR_Aをそれぞれのt×n = 16 × 128 = 2048 bitのブロックに挿入することが提案される。暗号学的な見地から、場合によってはr = 48 bitの任意データが推奨される。伝送される実際の圧力データはまだある程度のエントロピーを有していると仮定されるので、次にこのような妥協が続く。実際の分割、すなわち、どの箇所に圧力データがくるのか、及びいつ任意ビットがくるのかは暗号学的に重要ではない。実施例のRDSセンサのための提案は次のとおりである: 9 bitの圧力データ×244 + r = 32 bit 乱数（R_A）= 2048 = 128×16 = t×n。これにより、完全性保護時のリプレイ攻撃が効果的に防止される。攻撃者は少なくとも$2^{32+エントロピー(x)}$のプロトコル経過を完全に記録しなければならない。

【０１１７】

次に本発明による安全性コンセプトにおける決定的な第４ステップで２つの方法を巧みに結びつけることによって、いわゆる「トランザクション認証」をどのようにして得るかを示す。センサ安全性確保のためには、純粋なメッセージ認証だけでなく、特異性（uniqueness）及び新しさ（freshness）も必要とされる。"A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, Handbook of Applied Cryptography, CRC Press, 1996, 表9.10, 362ページ" を参照されたい。この文献箇所に表示された認証タイプの表は以下のとおりである。

【０１１８】

【表１】

	送信元の特定	データ完全性	適時性又は特異性	定義箇所
メッセージ認証	はい	はい	-	§9.6.1
トランザクション認証	はい	はい	はい	§9.6.1
実体認証	はい	-	はい	§10.1.1
鍵認証	はい	はい	望ましい	§12.2.1

【０１１９】

完全性確保（１行目）のためのMACによって新しさは得られない。このためには下から２行目の「実体認証」と呼ばれる認証が用いられる。トランザクション認証が最良であることがわかる。しかしながら、これにはそれぞれのパケットでメッセージ認証及び適宜な長さの時変パラメータが必要となる。したがって、例えば圧力データに関するMAC、十分な長さのタイムスタンプ又は（同期した）カウンタである。センサには標準的にはクロックも同期カウンタも設けられていないので、３行目（実体認証）と１行目（メッセージ認

証、完全性保護）との組み合わせが選択された。

【０１２０】

ここで、２つのプロトコルの決定的な結合が生じる。認証時に制御装置Ｂは、例えば乱数R_BをセンサＡに送信する。完全性保護時には、センサは乱数R_Aを制御装置に送信する。ドイツ国特許第102009002396号明細書では、R_AについてはじめにR_Bの32最下位ビットが使用され、それぞれのステップで、すなわち、t×n = 2048 bit毎にR_Aの値が１だけ上位に数えられる。これにより、２つのプロトコルは暗号学的に確実に結合される。シーケンスカウンタとして使用されるIsb(R_B)を知っていても、それぞれの認証において制御装置によって新たに選択されるので攻撃者には役に立たない。さらに、全ての乱数は制御装置Ｂにより生じるのでセンサ側Ａでは乱数発生器の実装が不要となる。しかしながら、センサ装置には今日では既に決定論的なRNGが存在する。

【０１２１】

このような実装のための重要な問題は、どのような頻度でシーケンスカウンタR_Aを繰り返すことなしに完全性保護プロトコルステップを実施することができるかである。32bitカウンタは単純にオーバフローすると仮定される（unsigned int wrap：符号なし整数のラップアラウンド）。すなわち、2^{32}×(2048-32)bitの圧力データを伝送することができる。毎秒8000×9bitのデータ伝送率では、カウンタの繰返しは、

【０１２２】

【数１６】

$$\frac{2^{32}\times(2^{10}-2^{5})}{8000\times 9s^{-1}}\approx 5{,}9\times 10^{7}s\approx 684$$

【０１２３】

日後に生じる。

【０１２４】

遅くともこの時間後には再認証を行うべきである。制御装置Ｂは自ら乱数R_Bをセンサに送信し、これにより、シーケンスカウンタR_A= $Isb_{31\cdots 0}(R_B)$のための開始値を算出することもできる。それぞれのデータパケット、すなわち、t×n bit毎に制御装置は新しい値R'_Aを受信する。伝送された値R'_Aが[R_A, R_A+ △R_A]の所定の窓内にある場合、例えば△R_A:=3である場合、制御装置はこのシーケンスカウンタを承諾し、$R_A = R'_A + 1$を設定する。センサがシーケンスカウンタによってリセットできないことが重要である。KeeLoqにおける劣悪な設計解決方法を参照されたい。通常、すなわち、パケット紛失がなければ△R_A := 0で十分なはずである。すなわちR_Aはそれぞれのパケットによって１だけ増分される。

【０１２５】

トランザクション認証、すなわち、送信元の完全性、データの完全性及び新しさは、双方向通信プロトコル（チャレンジ-レスポンス）によって、又は全体的な独立したタイムスタンプ又はシーケンスカウンタによって達成することができることをここで明示的に述べておく。要求者／証明器、ここではセンサは、カウンタ又はクロックをリセットすることはできない。これは、KeeLoqの場合のようにサービス拒否攻撃につながる場合がある。カウンタは所定の有効ウインドウにおいてのみ承諾され、攻撃者はカウンタを常に無効となるように操作する。その結果、センサは承諾されないか、又は新たに学習させる必要がある。盗難防止の際には、認証される車両鍵自体が不正使用防止されており、このようにして不自然なカウンタを生成できないようになっている。車両鍵の内部のカウンタが最後に認証されたカウンタと数十万だけ異なっている場合、たいていは車両鍵に新たに学習させる必要がある。

【０１２６】

したがって、DE-102009002396によりセンサの認証とセンサデータの完全性保護との関連付けがコンセプト全体の安全性を高めるが、別の問題が未解決である。サイドチャンネ

ル攻撃（ＳＣＡ）、例えば差分電力解析(Differential Power Analysis: ＤＰＡ)による秘密鍵の紛失は除外できない。これまでに説明した構造から、次の問題提起を導くことができる：例えばサイドチャンネル攻撃により秘密の一部が知られた場合（ここでは鍵の１つKauth又はKmacの紛失）、プロトコル全体の望ましい特性はどのように挙動するのか。これまでに提案されたプロトコルでは、このような部分プロトコルの破損は、プロトコル全体の破損につながる。これまでに述べた安全性コンセプトでは、いわゆる「セッション分割攻撃(Session Splitting Attack)」、すなわち、いずれかの秘密鍵の破損を利用して安全性コンセプト全体を無効にする攻撃が可能であろう。この場合、攻撃者は、例えば完全性保護の破損後（Kmacが既知である）にトランザクション（すなわちセッション）を分割することができる：攻撃者は自ら選択したデータ自体について完全性保護のための値を計算し（Kmacは既知なので）、一次ソースを利用してチャレンジ‐レスポンスプロトコルをうまく完成させる。

【０１２７】

ＳＣＡは、例えば２つの部分プロトコルプロセスの間の最低限許容時間を延ばすことによって困難となった。これにより攻撃時に可能なサンプルレートの低下が生じ、したがって攻撃の成果を得るための消費時間が長くなる。部分プロトコルの完全性保護においてこの最低限の時間が延長された場合に、必要とされるデータ率による狭い限界が設定され、認証時には極めて高い値が使用され得る。

【０１２８】

以下に説明するコンセプトは、部分プロトコルに対してセッション分割攻撃（例えばSCA攻撃）が成功した場合にプロトコル全体の破損に対する効果的かつ効率的な措置である。この場合、これまでに説明したプロトコルの変更時に２つの部分プロトコルを相互に結びつけ、これにより、（例えば他の部分プロトコルの場合よりも防止することが困難となり得るSCAにより）例えばKmacが知られたことによりプロトコル全体の破損が引き起こされないようにすることが提案される。提案された変更はリソース及び安全性に関してもとのプロトコルのポジティブな特徴を含む。提案された変化態様の基盤は、チャレンジ及びK_{auth}からの秘密の値を確実に導くことである。この値は、それぞれのMAC計算で使用されるノンス（number only used once:一回だけ使用される数）の初期値R_{a0}としての役割を果たす。オリジナルプロトコル(DE-102009002396)では、この初期値は、攻撃者に既知の値であるチャレンジから直接に形成される。

【０１２９】

図６又は図６ａ及び図６ｂは、センサデータのトランザクション安全性のための完全性保護と認証とを結びつけるプロトコル全体を説明するための概略図を、新たに提案された変化態様で示す。この場合、センサＡと制御装置Ｂとの通信が示されている。センサＡから制御装置Ｂへのセンサデータはｘで示されている。センサＡ及び制御装置Ｂは共通の鍵$K_{AB,\ Auth}$、例えば128bitのAES鍵を認証のために使用することができ、共通の鍵$K_{AB,MAC}$、例えば128bitのAES鍵をメッセージ認証コードCMACのために使用することができる（代替的なメッセージ認証コードEMACでは２つの鍵$K_{AB,MAC1}$，$K_{AB,MAC2}$となる）。

【０１３０】

図６ａは、安全性コンセプト全体の第１ステップ、正確にはチャレンジ‐レスポンス方式の記述を示す。第１ステップでは、制御装置Ｂは暗号学的に安全な64bitの乱数R_Bを生成し、これをアイデンティティID_Bと共にセンサＡに送信する: B -> A: $R_B \| ID_B$。

【０１３１】

センサＡは対称的なAES鍵$K_{AB,Auth}$によって、受信したデータパケット($R_B \| ID_B$)を暗号化する。生じた長さ結果、128bitの$Enc_{KAB,Auth}(R_B \| ID_B)$から64msb(most significant bits: 最上位ビット)が取り出され、アイデンティティID_Aと共に制御装置Ｂに送信される: A -> B: $msb_{127\cdots64}[Enc_{KAB,Auth}(R_B \| ID_B)] \| ID_A$。

【０１３２】

データ$R_B \| ID_B$がセンサに送信された直後に、制御装置Ｂは$msb_{127\cdots64}[Enc_{K\ AB,Auth}(R_B \| ID_B)]$を計算し始める。次いで制御装置Ｂは受信したデータ、$msb_{127\cdots64}[Enc_{K\ AB,Aut}$

$_h(R_B \| ID_B)$]及びID_Aを比較する。それぞれ２つの値が一致した場合にのみプロトコルが継続され、センサは制御装置に対して認証される。１つの値のみが異なっている場合、「センサ認証失敗(Authentication of sensor failed)」というエラー報告が生成される。

【０１３３】

図６ｂは、チャレンジ及びK_{auth}からノンス（一回だけ使用される数）R_aの初期値R_{a0}を導き出して認証及び完全性保護の関連付けを示す。

【０１３４】

図６ａからわかるように、R_bは、認証応答を計算するためにID_bと共にセンサにおいて既に暗号化される。認証のチャレンジ-レスポンス方式のレスポンスとしてres := MSB_{64}(Enc Kauth) (Rb‖IDb)が伝送され、最後の8Bytesは切り捨てによって秘密に保持される。最初のR_a(R_{a0})は、$Enc_{K\ AB,\ Auth}$ ($R_B \| ID_B$)から、例えばLSB_{64}(Enc K_{auth} ($R_B \| ID_B$)として導き出され、したがって、例えばDE-102009002396におけるR_bの一部の代わりにEnc K_{auth} ($R_b \| ID_b$)におけるチャレンジ-レスポンスで切り捨てられた部分がR_aとして使用される。すなわち、センサは、例えば図６ｂに示すように、($Enc_{K\ AB,\ Auth}$ ($R_B \| ID_B$) ‖ ID_Aの64最下位ビット(least significant bits : Isb)をR_A又はノンスR_Aの初期値R_{a0}として保存する。

【０１３５】

すなわち、効率化のために、認証のチャレンジ-レスポンス方式からの既に計算された部分、例えば、送信されなかった（レスポンス切り捨て）部分が初期値R_{a0}として使用される。センサデータは上述のようにxによって示されている。以下では、上述のように例えばEMAC又はCMACなどのメッセージ認証コードを計算するための様々な可能性を用いることができる。時変パラメータと連結されたセンサデータxはメッセージ認証コードMAC_{KMac}のmsb_{64}と共に制御装置に送信される（図６ｂを参照のこと）。オリジナルプロトコルでは、時変パラメータ又は現在のノンスが暗号化されずに伝送される。しかしながら、ここで提案した変更されたプロトコルではこのようなことはもはや可能ではない。なぜなら、これは認証のレスポンス切捨てを弱めるからである。レスポンス切捨てを弱めないためには、好ましい構成では付加的にR_aの代わりに値cのみをオーバフロー、c:= R_a-R_{a0}を有する符号なしの整数として伝送し、さらにR_aをMAC計算で使用する（図６ｂを参照のこと）：

A -> B: x‖Isb32(c)‖msb64 (MAC_{Kmac} (x‖RA))。

次いでシーケンスカウンタR_Aが１だけ高められる。

【０１３６】

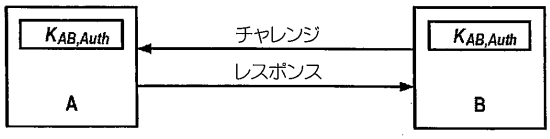
制御装置Ｂはセンサデータxを受信し、これを処理及び／又はさらに転送する。受信データによって、制御装置の側でMAC値が計算され、受信したMAC値と比較される。制御装置ではc又はR_Aのための出力値が既知なので、制御装置は、送信されたシーケンスカウンタcが所定の間隔内に位置するか否かをチェックすることができる。「はい」の場合、制御装置はこれを承諾し、上記計算を実施する。「いいえ」の場合、センサＡ及び制御装置Ｂは明らかにあまりに多数のパケットを紛失したか、又は同期性が失われてしまったことになる。この場合、とりわけc又はR_aの新たな処理のために再認証を行うべきである。MAC計算及び比較の後、c又はR_Aは１だけ増分される。

【０１３７】

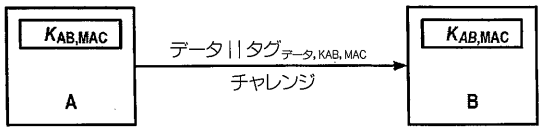
一般に、トランザクション認証又はセンサデータ-完全性保護の枠内で使用される時変パラメータを計算するために暗号学的な認証メッセージの少なくとも第２部分を使用し、暗号学的な完全性保護を計算するために暗号学的な認証メッセージの少なくとも第３部分を使用し、センサ認証の枠内では、暗号学的な認証メッセージ$MSB_{64}Enc_{KAB,Auth}$(RB‖ID_B)の第１部分を使用するように定式化することもできる。図６に示した実施例では、第２及び第３部分には$LSB_{64}Enc_{KAB,\ Auth}$(RB‖ID_B)が相当する。この場合、第２部分及び第３部分はもちろん同一である必要はない。第１部分が第２部分及び第３部分と重ならないか、又は少なくともこれらと同一ではない場合、有利であり得る。第２部分及び第３部分は同一であるか、重なるか、又は$Enc_{KAB,Auth}$(RB‖ID_B)から独立した部分であってもよい。

【０１３８】
現在のパラメータR_aは、それぞれのセンサデータ-トランザクション時に計算され、時変パラメータcは、それぞれのセンサデータ-トランザクション時において、$LSB_{64}Enc_{KAB,Auth}(RB \| ID_B)$の例における初期値$R_{a0}$と前回のセンサデータトランザクションの実際のパラメータとの差に相当する。
【０１３９】
上記の本発明による安全性コンセプト全般に対して代替的な実施例もある。
【０１４０】
トランザクション認証のそれぞれのステップで、上記実施例R_Aは１だけ増分される。規定された規則にしたがった他の増分又は他の変更もこのためには可能であり、有利であり得る。ここでは認証及び完全性保護又はトランザクション認証のための好ましい（暗号学的な）方法は、例示的なものにすぎず、明細書に挙げた代替的方法又は明細書に挙げていない比較可能な方法と置き換えることもできる。
【０１４１】
数R_Bは、上記コンセプトでは乱数として挙げられた。一般に、この数はノンス(Number only used once)とすることもでき、例えば乱数の他にタイムスタンプ又はシーケンスカウンタとすることもできる。

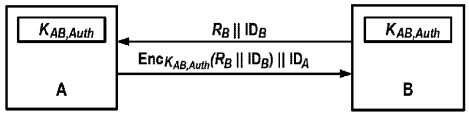
【図１】



【図２】

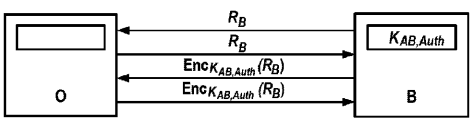


【図３】



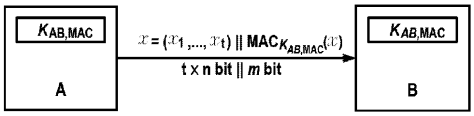
Figur 3

【図４】



Figur 4

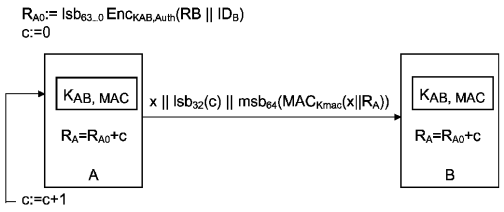
【図５】



Figur 5

【図６Ａ】

【図６Ｂ】



Figur 6b

フロントページの続き

(74)代理人 100147991
弁理士 鳥居 健一
(72)発明者 ニューサム，ジェームス
アメリカ合衆国ペンシルベニア州１５２１７，ピッツバーグ，ウィルキンス・アベニュー ６６１９
(72)発明者 スツェルヴィンスキー，ロベルト
ドイツ国 １２５５７ ベルリン，ジークフリート－ベルガー－シュトラーセ ３６
(72)発明者 ハイエク，ヤン
ドイツ国 ８１８２７ ミュンヘン，ゾラリンデンシュトラーセ ５７

審査官 青木 重徳

(56)参考文献 特開２００６－０８０５８７（ＪＰ，Ａ）
特表２００６－５２４３７７（ＪＰ，Ａ）
特開２００６－２７０３４８（ＪＰ，Ａ）
特開２００４－３２０２２９（ＪＰ，Ａ）
国際公開第２００７／０５２４７７（WO，Ａ１）
林誠一郎，斉藤泰一，村田祐一，“情報セキュリティ 暗号認証プログラムとそのメカニズム”，ＮＴＴ Ｒ＆Ｄ，日本，社団法人 電気通信協会，１９９５年１０月１０日，第４４巻，第１０号，ｐ．９１３－９２２
“本人認証技術の現状に関する調査報告書”，日本，情報処理振興事業協会セキュリティセンター［オンライ，２００３年 3月，ｐ．１３－１８，［平成２４年 8月１７日検索］，インターネット，ＵＲＬ，<http://www.ipa.go.jp/security/fy14/reports/authentication/authentication2002.pdf>

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｈ０４Ｌ ９／３２
Ｇ０９Ｃ １／００
Ｈ０４Ｌ ９／０８