(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7700960号
(P7700960)

(45)発行日　令和7年7月1日(2025.7.1)

(24)登録日　令和7年6月23日(2025.6.23)

(51)国際特許分類
H04L 9/32 (2006.01)

FI
H04L 9/32 200A

請求項の数　8　（全18頁）

(21)出願番号　特願2024-515190(P2024-515190)
(86)(22)出願日　令和4年4月11日(2022.4.11)
(86)国際出願番号　PCT/JP2022/017506
(87)国際公開番号　WO2023/199379
(87)国際公開日　令和5年10月19日(2023.10.19)
審査請求日　令和6年7月26日(2024.7.26)

(73)特許権者　000004226
日本電信電話株式会社
東京都千代田区大手町一丁目５番１号
(74)代理人　110004381
弁理士法人ＩＴＯＨ
(72)発明者　細山田 光倫
東京都千代田区大手町一丁目５番１号
日本電信電話株式会社内
(72)発明者　藤堂 洋介
東京都千代田区大手町一丁目５番１号
日本電信電話株式会社内
(72)発明者　安田 幹
東京都千代田区大手町一丁目５番１号
日本電信電話株式会社内
審査官　平井 誠

最終頁に続く

(54)【発明の名称】　情報処理装置、方法及びプログラム

(57)【特許請求の範囲】

【請求項１】

ホワイトボックス攻撃に対して安全なメッセージ認証符号のタグ生成関数$MAC1_K$（ただし、Ｋは鍵）を用いて、データＸの真正性を検証するためのタグ$T=MAC1_K(X)$を生成するように構成されているタグ生成部、を有し、

前記タグ生成関数$MAC1_K$は、

ホワイトボックス攻撃に対して安全で入出力長がｎビットのブロック暗号の暗号化関数E_Kと、出力長が$m \geqq 2n$ビットの所定のハッシュ関数Ｈと、所定の関数$f1, f2: \{0,1\}^m \rightarrow \{0,1\}^n$及び$f3: \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^n$とを用いて構成される、情報処理装置。

【請求項２】

前記タグ生成関数$MAC1_K$は、

前記データＸを入力に取り、

$h \leftarrow H(X)$又は$h \leftarrow H(E_K(0^n) \| X)$（ただし、‖ はビット結合）、

$s_1 \leftarrow E_K(f1(h))$、$s_2 \leftarrow E_K(f2(h))$、

$T \leftarrow E_K(f3(s_1, s_2))$又は$T \leftarrow$（$s_1$と$s_2$の排他的論理和）、

により前記タグＴを出力する、請求項１に記載の情報処理装置。

【請求項３】

前記関数ｆ１は、ｍビットの入力のうちの上位ｎビットを出力し、

前記関数ｆ２は、ｍビットの入力のうちの上位２ｎビットを取り出した後、前記上位２

ｎビットの下位ｎビットを出力し、
　前記関数ｆ３は、（ｘ，ｙ）∈｛０，１｝ｎ×｛０，１｝ｎが入力されたとき、ｘとｙの排他的論理和を出力する、請求項２に記載の情報処理装置。
【請求項４】
　前記関数ｆ１は、ｍビットの入力のうちの上位ｎビットを取り出した後、前記上位ｎビットから下位（ｎ－２）を表すｄ$_1$を更に取り出し、０１‖‖ｄを出力し、
　前記関数ｆ２は、ｍビットの入力のうちの上位２ｎビットを取り出した後、前記上位２ｎビットから下位（ｎ－２）を表すｄ$_2$を更に取り出し、１０‖‖ｄを出力し、
　前記関数ｆ３は、（ｘ，ｙ）∈｛０，１｝ｎ×｛０，１｝ｎが入力されたとき、ｘとｙの排他的論理和の下位（ｎ－２）ビットを表すｓ$_3$を取り出し、１１‖‖ｓを出力する、請求項２に記載の情報処理装置。
【請求項５】
　ホワイトボックス攻撃に対して安全な認証付き暗号の暗号化関数Ｅ$_{(K,K')}$（ただし、Ｋ'は鍵）を用いて、データＭの暗号文Ｃを生成すると共に、前記タグＴとして前記暗号文Ｃの真正性を検証するためのタグを前記タグ生成部により生成するように構成されている暗号化部、を有し、
　前記暗号化関数Ｅ$_{(K,K')}$は、
　前記タグ生成関数ＭＡＣ１$_K$と、ホワイトボックス攻撃に対して安全な共通鍵暗号の暗号化関数Ｅｎｃ$_{K'}$とを用いて構成される、請求項１乃至３の何れか一項に記載の情報処理装置。
【請求項６】
　前記暗号化関数Ｅ$_{(K,K')}$は、
　初期ベクトルＩＶと、関連データＡと、前記データＭとを入力に取り、
　Ｘ＝ＩＶ‖Ａ‖Ｍ‖ｌｅｎ（Ｍ）（ただし、‖ はビット結合、ｌｅｎ（Ｍ）はＭの長さを表すビット列）として前記タグ生成部によりタグＴ←ＭＡＣ１$_K$（Ｘ）を計算し、
　Ｃ←Ｅｎｃ$_{K'}$（ａｄｊ（Ｔ），Ｍ）（ただし、ａｄｊ（Ｔ）は、Ｔの長さを、前記共通鍵暗号の初期ベクトルの長さに調整する所定の関数）を計算し、
　前記タグＴと前記暗号文Ｃとを出力する、請求項５に記載の情報処理装置。
【請求項７】
　ホワイトボックス攻撃に対して安全なメッセージ認証符号のタグ生成関数ＭＡＣ１$_K$（ただし、Ｋは鍵）を用いて、データＸの真正性を検証するためのタグＴ＝ＭＡＣ１$_K$（Ｘ）を生成するように構成されているタグ生成手順、をコンピュータが実行し、
　前記タグ生成関数ＭＡＣ１$_K$は、
　ホワイトボックス攻撃に対して安全で入出力長がｎビットのブロック暗号の暗号化関数Ｅ$_K$と、出力長がｍ≧２ｎビットの所定のハッシュ関数Ｈと、所定の関数ｆ１，ｆ２：｛０，１｝ｍ→｛０，１｝ｎ及びｆ３：｛０，１｝ｎ×｛０，１｝ｎ→｛０，１｝ｎとを用いて構成される、方法。
【請求項８】
　ホワイトボックス攻撃に対して安全なメッセージ認証符号のタグ生成関数ＭＡＣ１$_K$（ただし、Ｋは鍵）を用いて、データＸの真正性を検証するためのタグＴ＝ＭＡＣ１$_K$（Ｘ）を生成するように構成されているタグ生成手順、をコンピュータに実行させ、
　前記タグ生成関数ＭＡＣ１$_K$は、
　ホワイトボックス攻撃に対して安全で入出力長がｎビットのブロック暗号の暗号化関数Ｅ$_K$と、出力長がｍ≧２ｎビットの所定のハッシュ関数Ｈと、所定の関数ｆ１，ｆ２：｛０，１｝ｍ→｛０，１｝ｎ及びｆ３：｛０，１｝ｎ×｛０，１｝ｎ→｛０，１｝ｎとを用いて構成される、プログラム。
【発明の詳細な説明】
【技術分野】
【０００１】
　本開示は、情報処理装置、方法及びプログラムに関する。

【背景技術】
【０００２】
一般に、暗号技術は、ブラックボックス攻撃に対して安全性が担保されるように設計される。ブラックボックス攻撃とは、暗号化関数や復号関数に関して、入出力のペアを攻撃者が知ることができるという設定の攻撃のことである。一方で、入出力のペアだけでなく、暗号化関数や復号関数を実装したソフトウェアの中身までも攻撃者が見ることができる、というような攻撃をホワイトボックス攻撃という。ブラックボックス攻撃に対して安全なことを「ブラックボックス安全」ともいい、ホワイトボックス攻撃に対して（或る程度）安全なことを「ホワイトボックス安全」ともいう。
【０００３】
暗号技術は主に機密性を得るための技術であるが、近年では、機密性だけでなく真正性を得る技術も重要となっている。真正性を得る技術としてはメッセージ認証符号（ＭＡＣ：Message Authentication Code）が知られている。また、機密性だけでなく真正性も同時に得ることができる技術として認証付き暗号又はＡＥＡＤ（Authenticated Encryption with Associated Data）と呼ばれるものが知られている（例えば、非特許文献１等）。ＡＥＡＤは、例えば、ＣＴＲやＣＢＣ等といった共通鍵暗号とメッセージ認証符号とを組み合わせて構成されることが多い。
【先行技術文献】
【非特許文献】
【０００４】
【文献】Phillip Rogaway and Thomas Shrimpton. A provable-security treatment of the key-wrap problem. In Serge Vaudenay, editor, Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings, Vol. 4004 of Lecture Notes in Computer Science, pp. 373-390. Springer, 2006.
【発明の概要】
【発明が解決しようとする課題】
【０００５】
しかしながら、これまでにホワイトボックス安全なＡＥＡＤは存在しなかった。一方で、ＡＥＡＤの多くは共通鍵暗号とメッセージ認証符号とを組み合わせて構成できるため、ホワイトボックス安全なメッセージ認証符号を実現できれば、それとホワイトボックス安全な共通鍵暗号とを組み合わせることで、ホワイトボックス安全なＡＥＡＤを構成することできる。
【０００６】
本開示は、上記の点に鑑みてなされたもので、ホワイトボックス安全なメッセージ認証符号を実現する技術を提供する。
【課題を解決するための手段】
【０００７】
本開示の一態様による情報処理装置は、ホワイトボックス攻撃に対して安全なメッセージ認証符号のタグ生成関数$MAC1_K$（ただし、Ｋは鍵）を用いて、データＸの真正性を検証するためのタグ$T=MAC1_K(X)$を生成するように構成されているタグ生成部、を有し、前記タグ生成関数$MAC1_K$は、ホワイトボックス攻撃に対して安全で入出力長がｎビットのブロック暗号の暗号化関数E_Kと、出力長が$m \geqq 2n$ビットの所定のハッシュ関数Ｈと、所定の関数$f1, f2 : \{0,1\}^m \rightarrow \{0,1\}^n$及び$f3 : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^n$とを用いて構成される。
【発明の効果】
【０００８】
ホワイトボックス安全なメッセージ認証符号を実現する技術が提供される。
【図面の簡単な説明】

【０００９】
【図１】検証関数のアルゴリズムの一例を示す図である。
【図２】ＳＩＶの暗号化関数のアルゴリズムの一例を示す図である。
【図３】ＳＩＶの暗号化関数の構成例を示す図である。
【図４】ＳＩＶの復号関数のアルゴリズムの一例を示す図である。
【図５】ホワイトボックス安全なメッセージ認証符号のタグ生成関数のアルゴリズムの一例を示す図である。
【図６】ホワイトボックス安全なメッセージ認証符号の構成例を示す図である。
【図７】第一の実施形態に係る通信システムの構成例を示す図である。
【図８】第二の実施形態に係る通信システムの構成例を示す図である。
【図９】コンピュータのハードウェア構成例を示す図である。
【発明を実施するための形態】
【００１０】
以下、本発明の第一の実施形態及び第二の実施形態について説明する。第一の実施形態では、ホワイトボックス安全なメッセージ認証符号を構成した上で、このメッセージ認証符号により真正性を満たすデータ通信を可能にする通信システム１について説明する。また、第二の実施形態では、第一の実施形態で構成したメッセージ認証符号からＡＥＡＤを構成し、このＡＥＡＤにより機密性及び真正性を満たすデータ通信を可能にする通信システム１について説明する。
【００１１】
なお、第一の実施形態に係る通信システム１は真正性を担保する必要があるデータ通信が発生する様々なシステムに適用可能である。同様に、第二の実施形態に係る通信システム１は機密性及び真正性を担保する必要があるデータ通信が発生する様々なシステムに適用可能である。
【００１２】
［準備］
以下、第一の実施形態及び第二の実施形態で用いる用語や技術等について説明する。
【００１３】
＜機密性と真正性＞
情報セキュリティの重要な概念に機密性と真正性がある。機密性とは、保護したいデータに関する情報を、悪意を持った攻撃者が不正に得ることはできないという状態のことである。機密性を得るための技術としては、共通鍵暗号や公開鍵暗号等がある。また、真正性とは、保護したいデータが攻撃者により不正に改ざんされていないという状態、又は、攻撃者が不正にデータを捏造できないという状態のことである。真正性を得るための技術としては、メッセージ認証符号やデジタル署名等がある。
【００１４】
＜共通鍵暗号技術＞
共通鍵暗号技術とは、利用者が共通の秘密鍵を用いるような暗号技術のことである。例えば、或る人が別の人にデータを共通鍵暗号技術で暗号化して送る際には、暗号化と復号で同一の秘密鍵を使う必要がある。共通鍵暗号技術は、ＲＳＡ暗号等の公開鍵暗号技術と違って鍵（の一部）を公開することはできない。その反面、素因数分解問題等の代数的問題を用いる必要がなく、処理が高速だという長所がある。ＳＳＬ／ＴＬＳ等をはじめとした多くのデータ保護技術は、鍵（の一部）を公開できる公開鍵暗号技術と高速な共通鍵暗号技術の双方の長所を上手く組み合わせて実現されている。公開鍵暗号技術と共通鍵暗号技術は両者ともにセキュリティの基盤技術として欠かせない存在である。以下で説明する第一の実施形態及び第二の実施形態は共通鍵暗号技術に関するものである。
【００１５】
以下、単に「共通鍵暗号」といえば、様々な長さのデータを暗号化して機密性を守ることが主目的の、狭義の共通鍵暗号を指すものとする。狭義の共通鍵暗号には、ＣＴＲモードやＣＢＣモード等の秘匿モード、ＫＣｉｐｈｅｒ－２等のストリーム暗号等が挙げられ

る。なお、ＡＥＳ（Advanced Encryption Standard）やＣａｍｅｌｌｉａ、ＳＰＡＣＥ（参考文献１）、ＳＰＮＢｏｘ（参考文献２）等のブロック暗号も本来は狭義の共通鍵暗号に分類しても差し支えないものであるが、本明細書では説明の都合上、「（狭義の）共通鍵暗号」といったときは、ブロック暗号は含まないものとする。

【００１６】

近年では、データを暗号化する際に機密性のみでなく真正性も同時に提供しようという共通鍵暗号が広く使われている。そのような暗号は、認証付き暗号又はＡＥＡＤと呼ばれる。ＡＥＡＤの代表例としては、ＧＣＭ、ＣＣＭ、ＳＩＶ（Synthetic Initialization Vector）（非特許文献１）等が挙げられる。ＡＥＡＤは狭義の共通鍵暗号とメッセージ認証符号（ＭＡＣ）とを組み合わせて構成されることが多い。ここで、メッセージ認証符号とは、機密性は考慮せずに真正性のみを提供することが目的の共通鍵暗号技術である。以下で説明する第一の実施形態はメッセージ認証符号、第二の実施形態はＡＥＡＤに関するものである。

【００１７】

以下、ブロック暗号、狭義の共通鍵暗号、メッセージ認証符号、及びＡＥＡＤに関してより詳細に説明する。また、暗号学的ハッシュ関数についても説明する。なお、本明細書においては、すべてのデータや関数の入出力はビット列で表されているものとする。

【００１８】

・ブロック暗号

ブロック暗号とは、固定長（例：２５６ビット）の鍵Ｋと固定長（例：１２８ビット）の平文Ｍとを入力に取り、平文と同じ長さの暗号文Ｃを出力する暗号Ｅのことである。ＣがＭ及びＫに対応する暗号文であることを$C = E_K(M)$と書く。E_Kは暗号化関数とも呼ばれる。ブロック暗号は復号可能であることが要請される。すなわち、暗号文を元に戻す復号関数D_Kが存在し、$D_K(E_K(M)) = M$が任意の平文Ｍに対して成り立つ。また、$D_K(E_K(C)) = C$が任意の暗号文Ｃに対して成り立つ。ブロック暗号の代表例としては、ＡＥＳやＣａｍｅｌｌｉａ等が挙げられる。

【００１９】

ここで、たとえ短い固定長データを安全に暗号化するような良いブロック暗号があったとしても、任意長データ（特に長いデータ）をどうすれば安全に暗号化できるかは自明でないことに注意されたい。

【００２０】

・狭義の共通鍵暗号

本明細書で狭義の共通鍵暗号といえば、固定長の鍵Ｋと固定長の初期ベクトルＩＶと任意長の平文Ｍとを入力に取り、暗号文Ｃを出力する暗号Ｅのことであるものとする。ここで、ＩＶは、原則として、暗号化処理を一度行うごとに代わる値である。ＩＶは暗号化処理ごとにランダムに選ばれることもある。また、単一のＩＶが繰り返し使われることを許容する方式も存在する。ブロック暗号ではＭが固定長（例：１２８ビット）であったが、狭義の共通鍵暗号ではＭは任意長であり、例えば、Ｍは１ビットでもよいし、１ギガバイト等であってもよい。Ｃが（Ｋ，ＩＶ，Ｍ）に対応する暗号文であることを$C = Enc_K(IV, M)$と書く。Enc_Kは暗号化関数とも呼ばれる。ブロック暗号と同様に、狭義の共通鍵暗号も復号可能である。すなわち、暗号文を元に戻す復号関数Dec_Kが存在し、$Dec_K(IV, Enc_K(IV, M)) = M$が任意の平文Ｍに対して成り立つ。

【００２１】

任意長の平文を暗号化するための共通鍵暗号は、ブロック暗号を利用して作ることが多い。狭義の共通鍵暗号やメッセージ認証符号、ＡＥＡＤ等の構成であってブロック暗号を利用するものをブロック暗号利用モード又は単にモードという。モードでは、必要に応じてブロック暗号以外の共通鍵暗号技術を用いることもある。本明細書では、特に、狭義の共通鍵暗号を作るためのモードを秘匿モードと呼ぶことにする。秘匿モードの代表例としては、ＣＴＲモードやＣＢＣモード等が挙げられる。

【００２２】

・メッセージ認証符号（ＭＡＣ）
メッセージ認証符号は、固定長の鍵Ｋと任意長のメッセージＭとを入力に取り、タグと呼ばれる固定長の出力Ｔを出す関数（タグ生成関数）$\mathrm{MAC}_K(M)$と、（Ｋ，Ｍ，Ｔ）を入力に取り、記号
【００２３】
【数１】

$$\top \text{ 又は } \bot$$

を出力する検証関数$\mathrm{VER}_K(M,T)$とで構成される。ここで、
【００２４】
【数２】

$$\mathrm{VER}_K(M,T)=\top$$

は「ＴがＭとＫから生成された正当なタグである」ということを意味し、$\mathrm{VER}_K(M,T)=\bot$は「Ｔは不正に生成されたタグである」ということを意味する。安全なメッセージ認証符号は大雑把にいえば「秘密鍵Ｋを知っている者しか正当なＴの値を計算することはできない」、「異なるメッセージに対応するタグの値は異なる」という性質を満たし、不正な改ざんの検知（つまり、真正性の獲得）に用いることができる。
【００２５】
本明細書では、検証関数VER_Kが図１に示すＡｌｇｏｒｉｔｈｍ１のようにして計算されるようなメッセージ認証符号のみを考えるものとする。すなわち、検証関数VER_Kは、メッセージＭとタグＴとを入力に取り、$T' \leftarrow \mathrm{MAC}_K(M)$を計算（１行目）した後、Ｔ'＝Ｔであれば Ｔは正当なタグであることを表す記号を出力し（２～３行目）、そうでなければＴは不正なタグであることを表す記号を出力する（４～５行目）。
【００２６】
狭義の共通鍵暗号と同様に、メッセージ認証符号（のタグ生成関数）もブロック暗号を用いて（つまり、ブロック暗号利用モードとして）実現されることがある。タグ生成関数がブロック暗号利用モードとして実現されるメッセージ認証符号の代表例としては、ＣＭＡＣ等が挙げられる。
【００２７】
・認証付き暗号（ＡＥＡＤ）
ＡＥＡＤの暗号化関数Ｅは、狭義の共通鍵暗号が入力として取る（Ｋ，ＩＶ，Ｍ）に加えて、任意長の関連データＡも入力に取る。また、この暗号化関数Ｅは、暗号文Ｃに加えて、タグＴも出力する。（Ｔ，Ｃ）が（Ｋ，ＩＶ，Ａ，Ｍ）に対応するタグ及び暗号文であることを（Ｔ，Ｃ）＝E_K（ＩＶ，Ａ，Ｍ）と書く。狭義の共通鍵暗号と同様に、暗号文を元に戻す復号関数D_Kが存在し、D_K（ＩＶ，Ａ，E_K（ＩＶ，Ａ，Ｍ））＝Ｍが任意の平文Ｍに対して成り立つ。
【００２８】
ＡＥＡＤの復号関数D_Kはメッセージ認証符号の検証関数と同様に、改ざんを検知する機能を持つ。つまり、暗号文Ｃ及びタグＴが不正に生成された値であれば、D_Kは検証に失敗したことを示す記号⊥を出力する（つまり、D_K（ＩＶ，Ａ，（Ｔ，Ｃ））＝⊥となる。）。ＴとＣが暗号化関数により正当に計算された値である場合は、上述した通り、元

のメッセージMが出力される。

【００２９】

安全なＡＥＡＤは、メッセージMを暗号文Cに変換して機密性を守り、かつ、改ざんからも保護して真正性を保障する。攻撃者はMの値が何であるかCを見ただけではわからない。一方で、関連データAに関しては機密性が保障されない。すなわち、大雑把にいって、Aの値は暗号化されず攻撃者から見えることを前提とする。しかしAについても真正性は保障され、改ざんからは保護される。つまり、もしAが途中で別の値A'に改ざんされた場合、復号関数は検証に失敗し、D_K（ＩＶ，A'，（Ｔ，Ｃ））＝⊥となる。このため、機密性はなくてもよいが真正性は保障したいデータ（例えば、ＩＰパケットのヘッダ等）がAとなる。

【００３０】

ＡＥＡＤは狭義の共通鍵暗号とメッセージ認証符号とを組み合わせて構成されることが多い。そのような構成の代表的なものとしては、ＧＣＭ、ＣＣＭ、ＳＩＶ（非特許文献１）等が挙げられる。例えば、狭義の共通鍵暗号（$Enc_{K'}$，$Dec_{K'}$）とメッセージ認証符号のタグ生成関数MAC_Kとが与えられたものとする。このとき、ＳＩＶの暗号化関数$E_{(K,K')}$は、例えば、図２に示すＡｌｇｏｒｉｔｈｍ２で定義される。すなわち、ＳＩＶの暗号化関数$E_{(K,K')}$は、（ＩＶ，Ａ，Ｍ）を入力に取り、Ｔ←MAC_K（ｐａｄ（ＩＶ，Ａ，Ｍ））及びＣ←$Enc_{K'}$（ａｄｊ（Ｔ），Ｍ）を計算（１～２行目）した後、（Ｔ，Ｃ）を出力する（３行目）。このように構成される暗号化関数$E_{(K,K')}$を模式的に表すと図３のようになる。ここで、ｐａｄは、ビット列の三つ組（ＩＶ，Ａ，Ｍ）を入力に取り、単一のビット列ＩＶ∥Ａ∥Ｍ∥ｌｅｎ（Ｍ）を出力する関数である。なお、ｘ∥ｙはｘとｙのビット結合を表し、ｌｅｎ（Ｍ）はＭのビット長を何等かのビット列で表現したものである。また、ａｄｊは、MAC_Kの出力であるＴを入力に取り、Ｔの長さ（以下、この長さをτビットとする。）が、$Enc_{K'}$や$Dec_{K'}$で使われるＩＶの長さ（以下、この長さをνビットとする。このＩＶはＳＩＶが入力に取るＩＶとは別であることに注意されたい。）と違う場合に、Ｔの一部のビットを切り捨てたり、Ｔに対して余分なビットを付け加えたりして長さをνビットに調節する関数である。具体的には、τ＝νの場合、ａｄｊ（Ｔ）は入力Ｔをそのまま出力する。一方で、τ＞νの場合、ａｄｊ（Ｔ）は入力Ｔの下位νビットを出力する。また、τ＜νの場合、ａｄｊ（Ｔ）は$0^{\nu-\tau}$∥Ｔを出力する。ここで、$0^{\nu-\tau}$は、すべてのビットが０であるような（ν－τ）ビットのビット列である。

【００３１】

また、ＳＩＶの復号関数$D_{(K,K')}$は、例えば、図４に示すＡｌｇｏｒｉｔｈｍ３で定義される。すなわち、ＳＩＶの復号関数$D_{(K,K')}$は、（ＩＶ，Ａ，Ｔ，Ｃ）を入力に取り、Ｍ←$Dec_{K'}$（ａｄｊ（Ｔ），Ｃ）及びＴ'←MAC_K（ｐａｄ（ＩＶ，Ａ，Ｍ））を計算（１～２行目）した後、Ｔ＝Ｔ'であればＭを出力し（３～４行目）、そうでなければ⊥を出力する（５～６行目）。

【００３２】

なお、図２に示すＡｌｇｏｒｉｔｈｍ２と図４に示すＡｌｇｏｒｉｔｈｍ３では、ＭＡＣとＥｎｃ及びＤｅｃとで独立した異なる鍵（ＫとＫ'）を用いているが、ＫとＫ'とが独立していない（つまり、Ｋ＝Ｋ'）としてもよい。

【００３３】

図２に示すＡｌｇｏｒｉｔｈｍ２と図４に示すＡｌｇｏｒｉｔｈｍ３では、説明の都合上、ＳＩＶの構成が提案された原論文（非特許文献１）から変更している箇所が何点かある。まず、原論文ではＩＶを入力に取ることはない（Ａの一部として見做されている）が、図２に示すＡｌｇｏｒｉｔｈｍ２と図４に示すＡｌｇｏｒｉｔｈｍ３ではＩＶも入力に取るものとしている。また、原論文では、MAC_Kとｐａｄを組み合わせる代わりに、ビット列の組を入力として取れる疑似ランダム関数が用いられている。ａｄｊも、説明の都合上、図２に示すＡｌｇｏｒｉｔｈｍ２と図４に示すＡｌｇｏｒｉｔｈｍ３に導入したものである。ＳＩＶの本質は「鍵付き関数に（ＩＶと）ＡとＭを入力し、出力Ｔを$Enc_{K'}$

のＩＶの代わりに使う」というものであるため、上記の変更はその本質を変えるものではない。

【００３４】

・暗号学的ハッシュ関数

暗号学的ハッシュ関数（以下、単にハッシュ関数と呼ぶ）Ｈは、任意長のデータＭを入力に取り、固定長（例：２５６ビット）のビット列Ｈ（Ｍ）を出力する。ハッシュ関数は基本的にその仕様が公開されており、誰でも計算ができる固定された関数である。特に、ハッシュ関数は鍵を入力に取らない。安全なハッシュ関数は、原像攻撃や衝突攻撃に対して耐性を持つことが求められる。安全かつ広く使われているハッシュ関数の例としては、ＳＨＡ－２やＳＨＡ－３等が挙げられる。ハッシュ関数は鍵を入力に取らないが、慣例的に共通鍵暗号技術に分類される。

【００３５】

＜ホワイトボックス暗号技術＞

一般に、暗号技術は、ブラックボックス攻撃に対して安全性が担保されるように設計される。ブラックボックス攻撃とは、暗号化関数や復号関数に関して、入出力のペアを攻撃者が知ることができるという設定の攻撃のことである。例えば、攻撃対象がブロック暗号の場合、攻撃者は、任意の平文Ｍを選び、それに対する暗号文Ｃ：＝E_K（Ｍ）を知ることが許される（あるいは、攻撃者は、任意の暗号文Ｃに対する平文Ｍ：＝D_K（Ｃ）を知ることが許される。）。攻撃者は、複数の平文に対応する暗号文を学習した後、秘密鍵Ｋを暴いたり、本来完全にランダムに見えるべきであるはずの暗号文に何等かの偏りがあることを発見したりすることを目指す。

【００３６】

ここで重要なのは、ブラックボックス攻撃の攻撃者は暗号アルゴリズムがどのように実装されているかまでは知ることができないということである。攻撃者に許されるのはあくまで、暗号化関数E_K又は復号関数D_Kの入出力ペアを学習することである。

【００３７】

一方で、攻撃者がE_KやD_Kの入出力ペアだけでなく、E_KやD_Kを実装したソフトウェア（秘密鍵に関する情報も含む）の中身までも見ることができる、というような攻撃をホワイトボックス攻撃という。ホワイトボックス攻撃の例としては、暗号技術が実装されたソフトウェアを利用するＰＣにマルウェアが忍び込み、そのソフトウェアのデータを外部に送信するというものが考えられる。ホワイトボックス攻撃に対しても安全性を或る程度担保できるような暗号技術をホワイトボックス暗号技術という。

【００３８】

当然、数十ビット～数千ビット程度の短い秘密鍵がソフトウェアに愚直に埋め込まれたような実装では、ホワイトボックス攻撃に対して安全性を一切担保できない。攻撃者が短い秘密鍵を見てコピーしてしまえば、すべての暗号文が自由に復号できるようになるためである。また、暗号技術が実装されたソフトウェアを攻撃者が自由に実行できる状況では何ら安全性を保障することはできない。例えば、復号関数が実装されたソフトウェアを攻撃者がいつでも自由に実行できる場合、任意の暗号文がその攻撃者によって自由に復号され、元の平文が見られてしまう。

【００３９】

そこで、ホワイトボックス攻撃に対する安全性を議論する際は、以下の１や２のような設定を前提とする。

【００４０】

１．秘密鍵が何等かの方法で非常に大きな（例：数十ギガバイト）データに拡大されている。

【００４１】

２．攻撃者の能力に一定の制限がある。例えば、ソフトウェア実装のデータのうち、外部に送信可能なデータ量の上限が決まっていたり、ソフトウェアの中身を見られる時間に上限があったりする等。

【００４２】
上記のような設定においてホワイトボックス安全な既存技術として、ＳＰＡＣＥ（参考文献１）やＳＰＮＢｏｘ（参考文献２）といったブロック暗号が存在する。これらブロック暗号の主張する安全性を破るようなホワイトボックス攻撃は現在のところ知られていない。
【００４３】
ホワイトボックス攻撃は明らかにブラックボックス攻撃よりも強力で攻撃者の自由度が大きい攻撃である。ゆえに、ホワイトボックス安全な暗号技術はブラックボックス安全であることに注意されたい。
【００４４】
［第一の実施形態］
以下、第一の実施形態について説明する。
【００４５】
真正性は機密性に劣らず重要な性質であるため、共通鍵暗号技術で暗号化を行う際にはＡＥＡＤを用いて機密性と真正性の両方を同時に保障しよう、というのが最近の共通鍵暗号設計の主な流れである。しかしながら、これまでにホワイトボックス安全なＡＥＡＤは存在していなかった。ＧＣＭやＣＣＭといった既存のＡＥＡＤはホワイトボックス攻撃を考慮に入れていない。特に、ＧＣＭやＣＣＭはブロック暗号利用モードであるが、たとえ利用するブロック暗号がホワイトボックス安全であっても、ＡＥＡＤとしてのＧＣＭやＣＣＭはホワイトボックス安全とは限らない。
【００４６】
一方で、上述したように、ＡＥＡＤは狭義の共通鍵暗号とメッセージ認証符号とを組み合わせて構成されることが多い。このため、ホワイトボックス安全なメッセージ認証符号を実現できれば、このメッセージ認証符号とホワイトボックス安全な狭義の共通鍵暗号とを組み合わせることで、ホワイトボックス安全なＡＥＡＤを構成することができる。
【００４７】
そこで、第一の実施形態では、ホワイトボックス安全なメッセージ認証符号を構成した上で、このメッセージ認証符号により真正性を満たすデータ通信を可能にする通信システム１について説明する。
【００４８】
＜ホワイトボックス安全なメッセージ認証符号を構成するためのアイデアの概要＞
メッセージ認証符号やＡＥＡＤは任意長の入力から短い固定長のタグＴを生成する必要がある。特に、入力が非常に長いときは、その長いデータを何等かの方法で短いデータへ圧縮せねばならない。ＧＣＭやＣＣＭといった（ホワイトボックス攻撃を考慮に入れていない）既存方式では、秘密鍵Ｋに依存した方法で圧縮を行っている。ブラックボックス攻撃者には圧縮の過程が見えないためＧＣＭやＣＣＭはブラックボックス攻撃に対して安全であるが、ホワイトボックス攻撃者はソフトウェアの中身が見えるため、圧縮の過程も見えてしまう。この「圧縮の過程が見える」ということが、メッセージ認証符号やＡＥＡＤの安全性を破ることに繋がりかねない。
【００４９】
そこで、メッセージ認証符号（及びＡＥＡＤのタグ生成）における圧縮の大半を、秘密鍵Ｋに依存しない暗号学的ハッシュ関数Ｈで行うことを考える。ハッシュ関数は元から「長いビット列を短いビット列に圧縮する」、「圧縮の過程は攻撃者にも見える」ということを前提にして設計されているため、暗号学的ハッシュ関数Ｈを用いて圧縮を行うようなタグ生成関数はホワイトボックス攻撃に対して安全であることが期待できる。
【００５０】
一方で、真正性を担保するためには、タグＴの値が秘密鍵に依存するようにしなければならない。ハッシュ関数は秘密鍵を入力に取らないため、Ｈだけでは安全性が保障できない。そこで、ハッシュ関数で圧縮した後に（ホワイトボックス安全な）ブロック暗号を複数回呼ぶことにより真正性を担保することを目指す。

【００５１】
＜ホワイトボックス安全なメッセージ認証符号の構成例＞
・記法
E_Kを入出力長がｎビットのブロック暗号の暗号化関数、Ｈを出力長が$m \geqq 2n$ビットの暗号学的ハッシュ関数とする。また、（$Enc_{K'}$，$Dec_{K'}$）を狭義の共通鍵暗号とする。加えて、ｆ１，ｆ２：｛０，１｝^{m}→｛０，１｝^{n}とｆ３：｛０，１｝^{n}×｛０，１｝^{n}→｛０，１｝^{n}を以下の設定例１又は設定例２のいずれかで定まる関数とする。
【００５２】
≪ｆ１～ｆ３の設定例１≫
ｆ１：ｍビットの入力ｘのうちの上位ｎビットを出力する。
【００５３】
ｆ２：ｍビットの入力ｘのうち、まず上位２ｎビットを取り出し、これをｄとおく。このｄの下位ｎビットを出力する。なお、ｍ＝２ｎならばｆ２は単に入力の下位ｎビットを出力することになり、ｘ＝ｆ１（ｘ）‖ｆ２（ｘ）が成り立つことに注意されたい。
【００５４】
ｆ３：入力（ｘ，ｙ）∈｛０，１｝^{n}×｛０，１｝^{n}が与えられたとき、ｘとｙの排他的論理和、つまり
【００５５】
【数３】

$$x \oplus y$$

を出力する。
【００５６】
≪ｆ１～ｆ３の設定例２≫
ｆ１：ｍビットの入力ｘのうち、まずその上位ｎビットをd_1とする。このd_1のうち更に下位（ｎ－２）ビットをd_1'として、０１‖d_1'を出力する。
【００５７】
ｆ２：ｍビットの入力ｘのうち、まずその上位２ｎビットを取り出し、これをｄとおく。このｄのうち更に下位（ｎ－２）ビットをd_2'として、１０‖d_2'を出力する。
【００５８】
ｆ３：入力（ｘ，ｙ）∈｛０，１｝^{n}×｛０，１｝^{n}が与えられたとき、ｘとｙの排他的論理和の下位（ｎ－２）ビットをs_3として、１１‖s_3を出力する。
【００５９】
・ホワイトボックス安全なメッセージ認証符号のタグ生成関数
以上のような記法及び設定の下、上述したアイデアに基づいて、任意長のデータＸを入力に取り、図５に示すＡｌｇｏｒｉｔｈｍ４のようにしてタグＴ＝ＭＡＣ１$_K$（Ｘ）を計算及び出力するタグ生成関数を考える。すなわち、任意長のデータＸを入力に取り、ｈ←Ｈ（Ｘ）を計算（１行目）した後、$s_1 \leftarrow E_K$（ｆ１（ｈ））及び$s_2 \leftarrow E_K$（ｆ２（ｈ））を計算（２行目）し、その後、$T \leftarrow E_K$（ｆ３（s_1，s_2））を計算及び出力（３～４行目）するタグ生成関数ＭＡＣ１$_K$を考える。このとき、ハッシュ関数Ｈが安全で、かつ、E_Kがホワイトボックス安全であれば、ＭＡＣ１$_K$もホワイトボックス安全であることが期待できる。このように構成されるタグ生成関数ＭＡＣ１$_K$を模式的に表すと図６のようになる。
【００６０】
なお、ｆ１～ｆ３の設定例２を用いた場合、３回呼ばれるブロック暗号の暗号化関数E_Kの入力が重複しない。このため、ｆ１～ｆ３の設定例１よりも安全性の向上が期待でき

る。
【００６１】
・ホワイトボックス安全なメッセージ認証符号の検証関数
図１に示すＡｌｇｏｒｉｔｈｍ１において、入力の「メッセージＭおよびタグＴ」を「データＸおよびタグＴ」、１行目の「Ｔ'←ＭＡＣ$_K$（Ｍ）」を「Ｔ'←ＭＡＣ１$_K$（Ｘ）」にそれぞれ変更すれば、ホワイトボックス安全なメッセージ認証符号の検証関数（これをＶＥＲ１$_K$（Ｘ，Ｔ）と表す。）が構成できる。
【００６２】
＜通信システム１の構成例＞
第一の実施形態に係る通信システム１の構成例を図７に示す。図７に示すように、第一の実施形態に係る通信システム１では、送信側の端末である通信端末１０と、受信側の端末である通信端末２０とがインターネット等の通信ネットワーク３０を介して通信可能に接続されている。なお、通信端末１０及び通信端末２０としては、例えば、ＰＣ、スマートフォン、タブレット端末、ウェアラブルデバイス、車載器、電化製品等といった通信可能な各種情報処理装置が挙げられる。
【００６３】
通信端末１０は、認証部１０１と送信部１０２とを有する。認証部１０１及び送信部１０２は、例えば、通信端末１０にインストールされた１以上のプログラムが、ＣＰＵ（Central Processing Unit）等のプロセッサに実行させる処理により実現される。
【００６４】
認証部１０１は、送信対象のメッセージＸ＝Ｍに対するタグＴをタグ生成関数ＭＡＣ１$_K$（Ｘ）により生成する。送信部１０２は、（Ｘ，Ｔ）を通信端末２０に送信する。これにより、通信端末１０は、送信対象のメッセージＸに対するタグＴ＝ＭＡＣ１$_K$（Ｘ）を認証部１０１により生成した上で、（Ｘ，Ｔ）を送信部１０２により通信端末２０に送信することができる。
【００６５】
通信端末２０は、受信部２０１と検証部２０２とを有する。受信部２０１及び検証部２０２は、例えば、通信端末２０にインストールされた１以上のプログラムが、ＣＰＵ等のプロセッサに実行させる処理により実現される。
【００６６】
受信部２０１は、（Ｘ，Ｔ）を通信端末１０から受信する。検証部２０２は、受信した（Ｘ，Ｔ）をＶＥＲ１$_K$（Ｘ，Ｔ）により検証する。これにより、通信端末２０は、通信端末１０から（Ｘ，Ｔ）を受信部２０１により受信し、その（Ｘ，Ｔ）の真正性を検証部２０２により検証することができる。
【００６７】
＜変形例＞
以下では、タグ生成関数ＭＡＣ１$_K$のアルゴリズム（図５に示すＡｌｇｏｒｉｔｈｍ４）の変形例について説明する。
【００６８】
・変形例１－１
図５に示すＡｌｇｏｒｉｔｈｍ４の１行目の「ｈ←Ｈ（Ｘ）」を「ｈ←Ｈ（Ｅ$_K$（０^n）‖Ｘ）」に変更する。秘密鍵に依存する値である$_K$（０^n）を追加で結合させることにより、安全性の向上が期待できる。
【００６９】
・変形例１－２
図５に示すＡｌｇｏｒｉｔｈｍ４の３行目の「Ｔ←Ｅ$_K$（ｆ３（ｓ$_1$，ｓ$_2$））」を
【００７０】
【数４】

$$T \leftarrow s_1 \oplus s_2$$

に変更する。ブロック暗号の暗号化関数E_Kを呼び出す回数が削減されるため、処理効率の向上が期待できる。

【００７１】

・変形例１－３

変形例１－１と変形例１－２とを組み合わせる。すなわち、図５に示すＡｌｇｏｒｉｔｈｍ４の１行目の「ｈ←Ｈ（Ｘ）」を「ｈ←Ｈ（E_K（0^n）┆┆Ｘ）」に変更すると共に、３行目の「Ｔ←E_K（ｆ３（s_1，s_2））」を「Ｔ←（s_1とs_2の排他的論理和）」と変更する。安全性をあまり損なわずに処理効率の向上が期待できる。

【００７２】

［第二の実施形態］

以下、第二の実施形態について説明する。第二の実施形態では、第一の実施形態で構成したメッセージ認証符号からＡＥＡＤを構成し、このＡＥＡＤにより機密性及び真正性を満たすデータ通信を可能にする通信システム１について説明する。

【００７３】

＜ホワイトボックス安全なＡＥＡＤの構成例＞

安全なハッシュ関数Ｈとホワイトボックス安全なブロック暗号の暗号化関数E_Kとに加えて、狭義の共通鍵暗号（$Enc_{K'}$，$Dec_{K'}$）であってホワイトボックス安全なものがあったとすれば、第一の実施形態で説明した$MAC1_K$と（$Enc_{K'}$，$Dec_{K'}$）からＳＩＶを構成することで、ホワイトボックス安全なＡＥＡＤを構成することができる。

【００７４】

すなわち、（$Enc_{K'}$，$Dec_{K'}$）をホワイトボックス安全な狭義の共通鍵暗号とする。このとき、図２に示すＡｌｇｏｒｉｔｈｍ２の１行目の「Ｔ←MAC_K（ｐａｄ（ＩＶ，Ａ，Ｍ））」を「Ｔ←$MAC1_K$（ｐａｄ（ＩＶ，Ａ，Ｍ））」としてＳＩＶの暗号化関数$E_{(K,K')}$を構成する。また、図４に示すＡｌｇｏｒｉｔｈｍ３の２行目の「Ｔ'←MAC_K（ｐａｄ（ＩＶ，Ａ，Ｍ））」を「Ｔ'←$MAC1_K$（ｐａｄ（ＩＶ，Ａ，Ｍ））」としてＳＩＶの復号関数$D_{(K,K')}$を構成する。これにより、ホワイトボックス攻撃という非常に強力な攻撃に対しても真正性と機密性の両方を担保したデータ通信を実現することができる。

【００７５】

なお、メッセージ認証符号と狭義の共通鍵暗号からＡＥＡＤを得るための構成はＳＩＶの他にも存在するが、ＳＩＶはＤＡＥ（deterministic authenticated-encryption）という非常に強い安全性定義を満足するため、強力なホワイトボックス攻撃に対して安全性を担保するにはＳＩＶを用いるのが好ましい。

【００７６】

＜通信システム１の構成例＞

第二の実施形態に係る通信システム１の構成例を図８に示す。図８に示すように、第二の実施形態に係る通信システム１では、送信側の端末である通信端末１０と、受信側の端末である通信端末２０とがインターネット等の通信ネットワーク３０を介して通信可能に接続されている。なお、通信端末１０及び通信端末２０としては、例えば、ＰＣ、スマートフォン、タブレット端末、ウェアラブルデバイス、車載器、電化製品等といった通信可能な各種情報処理装置が挙げられる。

【００７７】

通信端末１０は、暗号化部１０３と送信部１０２とを有する。暗号化部１０３及び送信部１０２は、例えば、通信端末１０にインストールされた１以上のプログラムが、ＣＰＵ

等の演算装置に実行させる処理により実現される。

【００７８】

暗号化部１０３は、送信対象のメッセージＸ＝Ｍの暗号文ＣとそのメッセージＸに対するタグＴとを、上記のホワイトボックス安全なＡＥＡＤの暗号化関数$E_{(K,K')}$（ＩＶ，Ａ，Ｍ）により生成する。暗号化部１０３には認証部１０１が含まれており、メッセージＸに対するタグＴを生成する際には、第一の実施形態と同様に、この認証部１０１が$MAC1_K$によりタグＴを生成（例えば、図２に示すＡｌｇｏｒｉｔｈｍ２の１行目）する。送信部１０２は、（Ｔ，Ｃ）を通信端末２０に送信する。これにより、通信端末１０は、送信対象のメッセージＸの暗号文ＣとそのメッセージＸに対するタグＴを暗号化部１０３により生成した上で、（Ｔ，Ｃ）を送信部１０２により通信端末２０に送信することができる。

【００７９】

通信端末２０は、受信部２０１と復号部２０３とを有する。受信部２０１及び復号部２０３は、例えば、通信端末２０にインストールされた１以上のプログラムが、ＣＰＵ等の演算装置に実行させる処理により実現される。

【００８０】

受信部２０１は、（Ｔ，Ｃ）を通信端末１０から受信する。復号部２０３は、上記のホワイトボックス安全なＡＥＡＤの復号関数$D_{(K,K')}$（ＩＶ，Ａ，Ｔ，Ｃ）によりタグＴを検証すると共に、その検証に成功した場合は暗号文Ｃを復号する。復号部２０３には検証部２０２が含まれており、タグＴを検証する際には、この検証部２０２が$MAC1_K$によりタグＴ'を生成し、このタグＴ'がタグＴと一致する否かを検証する（例えば、図４に示すＡｌｇｏｒｉｔｈｍ４の２～３行目）。これにより、通信端末２０は、通信端末１０から（Ｔ，Ｃ）を受信部２０１により受信し、その（Ｔ，Ｃ）の真正性の検証と復号とを復号部２０３により行うことができる。

【００８１】

＜変形例＞

以下では、ホワイトボックス安全なＳＩＶの暗号化関数$E_{(K,K')}$及び復号関数$D_{(K,K')}$のアルゴリズムの変形例について説明する。なお、ＳＩＶの暗号化関数$E_{(K,K')}$のアルゴリズムは、図２に示すＡｌｇｏｒｉｔｈｍ２の１行目の「Ｔ←MAC_K（ｐａｄ（ＩＶ，Ａ，Ｍ））」を「Ｔ←$MAC1_K$（ｐａｄ（ＩＶ，Ａ，Ｍ））」としたものである。また、ＳＩＶの復号関数$D_{(K,K')}$のアルゴリズムは、図４に示すＡｌｇｏｒｉｔｈｍ３の２行目の「Ｔ'←MAC_K（ｐａｄ（ＩＶ，Ａ，Ｍ））」を「Ｔ'←$MAC1_K$（ｐａｄ（ＩＶ，Ａ，Ｍ））」としたものである。

【００８２】

・変形例２－１

第一の実施形態の変形例１－１～変形例１－３のいずれかのアルゴリズムで実現される$MAC1_K$を用いる。

【００８３】

・変形例２－２

Ｋ'＝Ｋとする。これにより、保持しておくべき秘密鍵の長さが削減され、処理効率の向上が期待できる。

【００８４】

・変形例２－３

変形例２－１と変形例２－２とを組み合わせる。すなわち、第一の実施形態の変形例１－１～変形例１－３のいずれかのアルゴリズムで実現される$MAC1_K$を用いると共に、Ｋ'＝Ｋとする。

【００８５】

［ハードウェア構成例］

第一の実施形態及び第二の実施形態に係る通信端末１０及び通信端末２０は、例えば、図９に示すコンピュータ５００のハードウェア構成で実現可能である。図９に示すコンピ

ュータ５００は、入力装置５０１と、表示装置５０２と、外部Ｉ／Ｆ５０３と、通信Ｉ／Ｆ５０４と、ＲＡＭ（Random Access Memory）５０５と、ＲＯＭ（Read Only Memory）５０６と、補助記憶装置５０７と、プロセッサ５０８とを有する。これらの各ハードウェアは、それぞれがバス５０９を介して通信可能に接続されている。

【００８６】

入力装置５０１は、例えば、キーボード、マウス、タッチパネル、物理ボタン等である。表示装置５０２は、例えば、ディスプレイ、表示パネル等である。なお、コンピュータ５００は、例えば、入力装置５０１及び表示装置５０２の少なくとも一方を有していなくてもよい。

【００８７】

外部Ｉ／Ｆ５０３は、記録媒体５０３ａ等の外部装置とのインタフェースである。コンピュータ５００は、外部Ｉ／Ｆ５０３を介して、記録媒体５０３ａの読み取りや書き込み等を行うことができる。記録媒体５０３ａとしては、例えば、フレキシブルディスク、ＣＤ（Compact Disc）、ＤＶＤ（Digital Versatile Disk）、ＳＤメモリカード（Secure Digital memory card）、ＵＳＢ（Universal Serial Bus）メモリカード等が挙げられる。

【００８８】

通信Ｉ／Ｆ５０４は、通信ネットワークに接続するためのインタフェースである。ＲＡＭ５０５は、プログラムやデータを一時保持する揮発性の半導体メモリ（記憶装置）である。ＲＯＭ５０６は、電源を切ってもプログラムやデータを保持することができる不揮発性の半導体メモリ（記憶装置）である。補助記憶装置５０７は、例えば、ＨＤＤ（Hard Disk Drive）やＳＳＤ（Solid State Drive）、フラッシュメモリ等のストレージ装置（記憶装置）である。プロセッサ５０８は、例えば、ＣＰＵ等の演算装置である。

【００８９】

第一の実施形態及び第二の実施形態に係る通信端末１０及び通信端末２０は、図９に示すコンピュータ５００のハードウェア構成を有することにより、上述した各種処理を実現することができる。なお、図９に示すコンピュータ５００のハードウェア構成は一例であって、これに限られるものではない。例えば、図９に示すコンピュータ５００は、複数の補助記憶装置５０７や複数のプロセッサ５０８を有していてもよいし、図示したハードウェア以外の様々なハードウェアを有していてもよい。

【００９０】

本発明は、具体的に開示された上記の実施形態に限定されるものではなく、請求の範囲の記載から逸脱することなく、種々の変形や変更、既知の技術との組み合わせ等が可能である。

【００９１】

［参考文献］

参考文献１：Andrey Bogdanov and Takanori Isobe. White-box cryptography revisited: Space-hard ciphers. In Indrajit Ray, Ninghui Li, and Christopher Kruegel, editors, Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-16, 2015, pp. 1058-1069. ACM, 2015.

参考文献２：Andrey Bogdanov, Takanori Isobe, and Elmar Tischhauser. Towards practical whitebox cryptography: Optimizing efficiency and space hardness. In Jung Hee Cheon and Tsuyoshi Takagi, editors, Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I, Vol. 10031 of Lecture Notes in Computer Science, pp. 126-158, 2016.

【符号の説明】

【００９２】

１　　通信システム
１０　　通信端末
２０　　通信端末
３０　　通信ネットワーク
１０１　　認証部
１０２　　送信部
１０３　　暗号化部
２０１　　受信部
２０２　　検証部
２０３　　復号部
５００　　コンピュータ
５０１　　入力装置
５０２　　表示装置
５０３　　外部Ｉ／Ｆ
５０３ａ　　記録媒体
５０４　　通信Ｉ／Ｆ
５０５　　ＲＡＭ
５０６　　ＲＯＭ
５０７　　補助記憶装置
５０８　　プロセッサ
５０９　　バス

【図面】

【図１】

Algorithm 1 検証関数 $\mathrm{VER}_K(M, T)$

入力: メッセージ M およびタグ T
出力: $\top$ または $\bot$

$T' \leftarrow \mathrm{MAC}_K(M)$
if $T' = T$ **then**
　return $\top$
else
　return $\bot$
end if

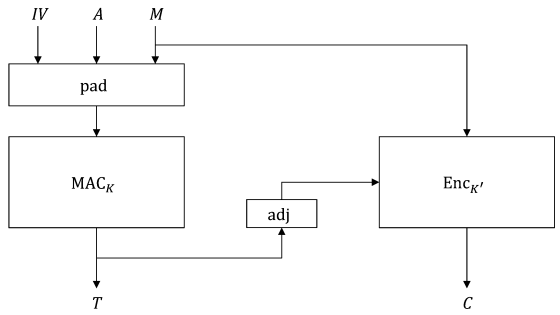
【図２】

Algorithm 2 SIV の暗号化関数 $\mathcal{E}_{(K,K')}(IV, A, M)$

入力: IV, A, M
出力: T, C

$T \leftarrow \mathrm{MAC}_K(\mathsf{pad}(IV, A, M))$
$C \leftarrow \mathrm{Enc}_{K'}(\mathsf{adj}(T), M)$
return (T, C)

【図３】



【図４】

Algorithm 3 SIV の復号関数 $\mathcal{D}_{(K,K')}(IV, A, T, C)$

入力: IV, A, T, C
出力: M あるいは $\perp$（$\perp$ は暗号文が不正に作られたものであることを示す記号）

$M \leftarrow \mathrm{Dec}_{K'}(\mathsf{adj}(T), C)$
$T' \leftarrow \mathrm{MAC}_K(\mathsf{pad}(IV, A, M))$
if $T = T'$ **then**
  return M
else
  return $\perp$
end if

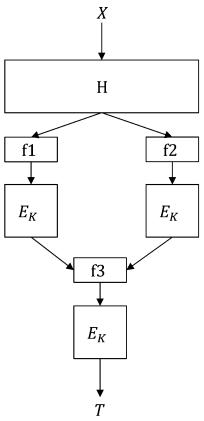
【図５】

Algorithm 4 ホワイトボックス安全なタグ生成関数 $\mathrm{MAC1}_K(X)$

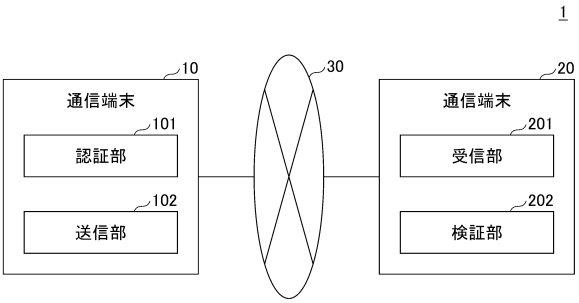
入力: X
出力: T

$h \leftarrow H(X)$
$s_1 \leftarrow E_K(\mathsf{f1}(h))$, $s_2 \leftarrow E_K(\mathsf{f2}(h))$
$T \leftarrow E_K(\mathsf{f3}(s_1, s_2))$
return T

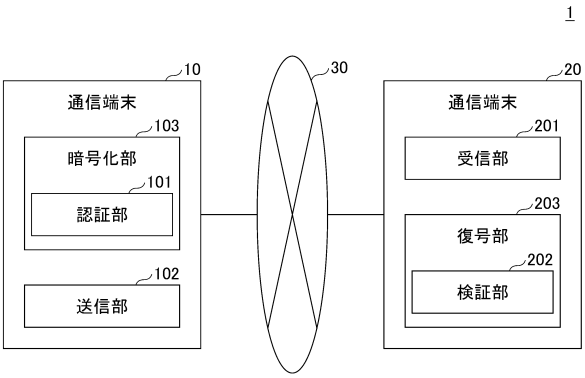
【図６】



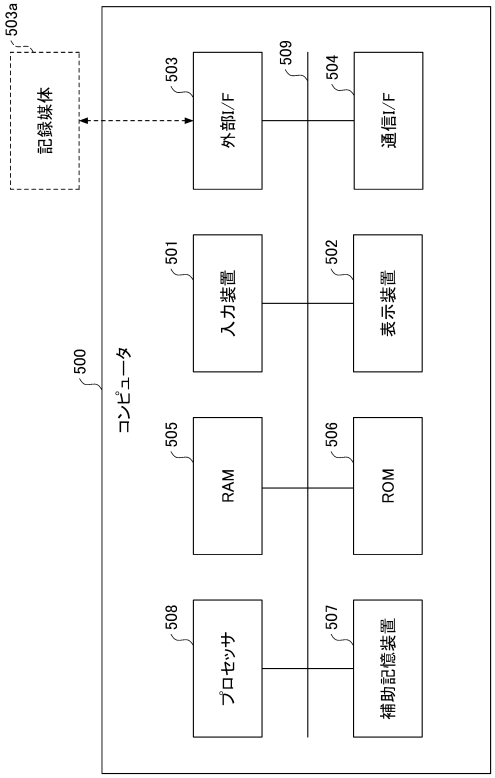
【図７】


1
10
通信端末
101
認証部
102
送信部
30
20
通信端末
201
受信部
202
検証部

【図８】


1
10
通信端末
103
暗号化部
101
認証部
102
送信部
30
20
通信端末
201
受信部
203
復号部
202
検証部

【図９】


503a
記録媒体
503
外部I/F
509
504
通信I/F
501
入力装置
502
表示装置
500
コンピュータ
505
RAM
506
ROM
508
プロセッサ
507
補助記憶装置

フロントページの続き

(56)参考文献　　米国特許出願公開第２０２０／００９９５２５（ＵＳ，Ａ１）
米国特許出願公開第２０１８／０３５９０８１（ＵＳ，Ａ１）

(58)調査した分野　(Int.Cl.，ＤＢ名)
Ｈ０４Ｌ　９／３２