

CRYPTREC Report 2005
(CRYPTREC-RP-2000-2005)

付録 5

暗号利用モード・MACに関する 技術調査報告

平成17年12月(平成18年5月改訂)

ハッシュ関数・暗号利用モード調査WG

ハッシュ関数・暗号利用モード調査WG委員構成

主査：古原 和邦 東京大学生産技術研究所

委員：廣瀬 勝一 福井大学工学部

委員：川村 信一 株式会社東芝

委員：古屋 聡一 株式会社日立製作所

委員：盛合 志帆 株式会社ソニー・コンピュータエンタテインメント

目次

1	はじめに	152
1.1	各用語の簡単な説明	152
2	記号や用語の厳密な定義	155
2.1	記法	157
2.2	ブロック暗号	157
2.3	暗号化方式	158
2.4	メッセージ認証コード	159
2.5	メッセージ認証つき暗号化方式	160
3	各々の利用モードを定義する文書	160
3.1	商務省連邦情報処理規格 (FIPS), 特殊文書 (SP)	160
3.1.1	ブロック暗号プリミティブ	161
3.1.2	ブロック暗号利用モード	161
3.2	ISO/IEC	162
3.3	JIS	163
3.4	金融に関するセキュリティ標準	163
3.5	ANSI	164
3.6	AES 利用モード候補方式	164
3.7	IEEE ディスクセクター暗号	164
3.8	NESSIE	166
3.9	その他工業製品や業界標準などで利用されたもの	166
3.10	その他学術論文などに提案されたもの	166
4	安全性の定義	167
4.1	ブロック暗号の安全性	167
4.1.1	擬似ランダム置換族	167
4.1.2	強擬似ランダム置換族	168
4.1.3	上記以外のブロック暗号の安全性	169
4.2	秘匿の安全性	169
4.3	従来の秘匿定義の関係	175
4.4	メッセージ認証コードの安全性	175
4.4.1	弱偽造不可能性	175
4.4.2	強偽造不可能性	176
4.4.3	MAC-G が決定的アルゴリズムである場合の安全性	177
4.4.4	上記以外の安全性	178

4.5	攻撃者の能力	178
4.6	証明可能安全性の仮定	179
4.7	利用モードに対する攻撃	179
5	秘匿に関する利用モード	180
5.1	ECB	180
5.2	CBC	182
5.3	k -CFB	185
5.4	OFB	188
5.5	CTR	190
5.6	2DEM	193
5.7	ABC	193
5.8	IGE	194
5.9	自己同期型利用モード	194
5.10	f_8 (3GPP)	196
6	認証暗号に関する利用モード	197
6.1	CCM	197
6.2	CWC	199
6.3	EAX	201
6.4	GCM	203
6.5	IACBC/XCBC	204
6.6	IAPM/OCB	207
6.7	k -PCFB	209
7	ディスクセクタ向け暗号利用モード	209
7.1	EMD	209
7.2	EME	210
7.3	CMC	210
7.4	LRW	211
7.5	NR	211
8	認証に関する利用モード	211
8.1	CBC MAC	212
8.2	EMAC	217
8.3	RMAC	220
8.4	XCBC	228
8.5	TMAC	231
8.6	OMAC/CMAC	234

8.7 XOR MAC	238
8.8 XECB MAC	244
8.9 PMAC	252
8.10 f_9 (3GPP)	255
8.11 NMAC, HMAC	258
8.11.1 NMAC	258
8.11.2 HMAC	260
9 まとめ	263
参考文献	265

1 はじめに

ブロック暗号は、暗号学的プリミティブの一つであり暗号化関数と復号関数により構成される。暗号化関数は、固定長の入力と鍵を受け入れ、入力を入力と同じ長さの出力に攪拌する。また、暗号化の際に利用した鍵と復号関数を用いることで出力を元の入力に戻すことができる。一般に、ブロック暗号は、入出力ペアから鍵情報を推定すること、および、鍵を知らずに出力から入力（あるいは入力から出力）を推測することが困難となるように設計されている。

しかしながら、ブロック暗号は固定長の入力しか変換できないため、長い平文を処理する際には、その利用方法を工夫する必要がある。その利用方法がブロック暗号利用モードであり、ブロック暗号利用モードには、秘匿性のみを提供するモードやメッセージ認証機能を提供するモード、更にはその両方を提供するモードなどがある。

本報告書では、各種標準化作業や学術出版物などで知られている暗号利用モードを調査し、その安全性、処理効率、その他工業的、学術的性質などについてまとめる。ただし、改ざん検出機能に関しては、ブロック暗号以外にハッシュ関数を用いるものも調査の対象としている。

1.1 各用語の簡単な説明

ブロック暗号は固定長、 n ビットの平文を暗号化する。多くのブロック暗号では $n = 64$ や、 $n = 128$ である。ブロック暗号は、ブロック暗号利用モードのプリミティブとして用いられる。ブロック暗号利用モードの主な機能はメッセージの秘匿と認証である。本報告書では、秘匿のためのブロック暗号利用モードを暗号化方式、認証のためのブロック暗号利用モードをメッセージ認証コード (Message Authentication Code, MAC) という。また、これらの機能を併せもつブロック暗号利用モードをメッセージ認証つき暗号化方式という。

- 暗号化方式はブロック暗号を用いて、 n ビットより長いメッセージを暗号化する。送信者と受信者が秘密鍵 K を共有しており、送信者は暗号化アルゴリズム \mathcal{E} を用いて、平文 M と鍵 K から暗号文 $C = \mathcal{E}_K(M)$ を計算し、 C を受信者に送る。 M の長さは n ビットよりも長くてよい。受信者は復号アルゴリズム \mathcal{D} を用いて、暗号文 C と鍵 K から平文 $M = \mathcal{D}_K(C)$ を計算する。

例として、ECB, CBC, OFB, CFB, CTR などがある。図 1 参照。

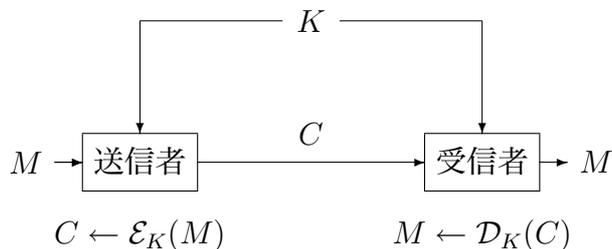


図 1: 暗号化方式のモデル

- メッセージ認証コードはブロック暗号を用いて、メッセージが偽造、改ざんされることを防ぐ技術である。送信者と受信者が秘密鍵 K を共有しており、送信者はタグ生成アルゴリズム \mathcal{G} を用いて、平文 M と鍵 K からタグ $T = \mathcal{G}_K(M)$ を計算し、メッセージ、タグのペア (M, T) を受信者に送る。 M の長さは、 n ビットよりも長くてもよい。 T は固定長であり、 32, 64, 96, 128 ビット程度の長さが一般的である。 (M, T) を受け取った受信者は確認アルゴリズム \mathcal{V} を用いて、受理信号、もしくは改ざん検出信号を出力する。 \mathcal{V} は、受け取ったメッセージに対し、 $T^* = \mathcal{G}_K(M)$ を計算し、 $T = T^*$ なら受理信号を、そうでなければ改ざん検出信号を出力する。

例として、CBC MAC, EMAC, OMAC, PMAC などがある。 図 2 参照。

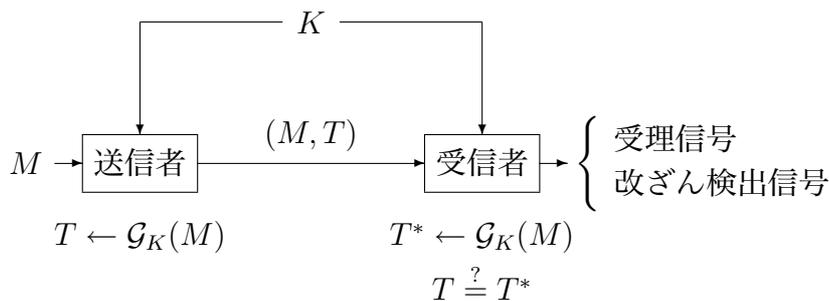


図 2: メッセージ認証コードのモデル

- メッセージ認証つき暗号化方式は、暗号化方式とメッセージ認証コードの機能を併せもつ。送信者と受信者が秘密鍵 K を共有しており、送信者は暗号化アルゴリズム \mathcal{E} を用いて、平文 M と鍵 K から暗号文 $C = \mathcal{E}_K(M)$ を計算し、 C を受信者に送る。受信者は復号ア

ルゴリズム D を用いて、暗号文 C と鍵 K から平文 $M = D_K(C)$ を、もしくは改ざん検出信号を出力する。

例として、CCM, IAPM, OCB などがある。また、任意の暗号化方式とメッセージ認証コードを組み合わせるメッセージ認証つき暗号化方式を構成する方法が知られている。図 3 参照。

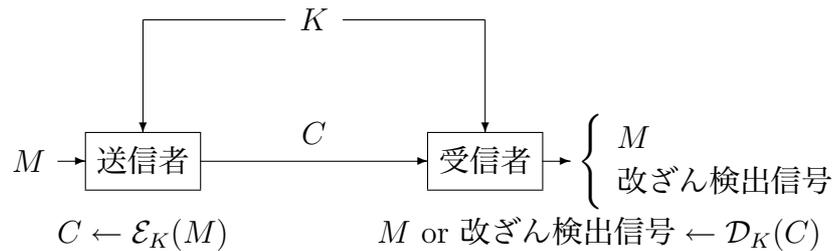


図 3: メッセージ認証つき暗号化方式のモデル

ブロック暗号利用モードの安全性 Bellare, Kilian, Rogaway により、CBC MAC の安全性が数学的に示された [BKR00]。ブロック暗号が安全な擬似ランダム置換族であれば、CBC MAC は偽造不可能性の意味で安全であることを示している。以降、多くのブロック暗号利用モードの安全性は、この種の証明可能安全性を根拠にしている。以下、暗号化方式、メッセージ認証コード、メッセージ認証つき暗号化方式、それぞれについて、安全性の定義を概説する。

- 暗号化方式に対しては、いくつかの安全性定義が存在する [BDJR97] が、ランダムビット列からの識別不能性が一般的である。暗号文 C か、もしくは C と同じ長さのランダムビット列 R が与えられ、有意な確率でこの 2 つを見分けることができないとき、暗号化方式は安全である、という。
- メッセージ認証コードに対しては、偽造不可能性が一般的である。(鍵 K を知らずに) $T = \mathcal{G}_K(M)$ となる (M, T) を出力できないとき、メッセージ認証コードは安全である、という。
- メッセージ認証つき暗号化方式では、暗号化方式の安全性定義とメッセージ認証コードの安全性定義の両方を考える。
 - 暗号文 C か、もしくは C と同じ長さのランダムビット列 R が与えられ、有意な確率でこの 2 つを見分けることができない。

- (鍵 K を知らずに) 改ざん検出信号 $\neq DEC_K(C)$ となる C を出力できない.

上記二つが成り立つとき、メッセージ認証つき暗号化方式は安全である、という。

ブロック暗号利用モードの効率 本報告書は、安全性を主眼にしているが、効率についても述べる。主に以下の点について述べる。

- 鍵長：安全性が変わらないのであれば、短いほうがよい。
- ブロック暗号鍵スケジューリングの呼び出し回数：一般的にブロック暗号鍵スケジューリングは計算時間がかかり、安全性が変わらないのであれば、少ないほうがよい。
- メッセージ M に対するタグを生成するのにかかるブロック暗号の呼び出し回数：安全性が変わらないのであれば、少ないほうがよい。
- 事前計算するべきブロック暗号の呼び出し回数：これらは、メッセージ M によらず実行できる。安全性が変わらないのであれば、少ないほうがよい。
- 並列処理性：ブロック暗号の並列処理が可能であれば、ハードウェア上で高速に実装できる。

2 記号や用語の厳密な定義

本章では本報告書で扱う記法をまとめるとともに、ブロック暗号、およびその利用モードの一般的な定義と、安全性の定義を述べる。

いくつかの標準化などでは、特定の利用モードを64ビットブロック暗号への適用のみに限定した記載などを行っている場合がある。しかし、本稿で扱う利用モードすべては、処理単位長に特化した利用モードであることはない。よって、その暗号学的な本質を議論することを目的として、汎用的なブロック暗号に対するモードとして議論を進める。具体的には、内部で用いるブロック暗号のブロック長を n ビットとする。

本稿では、排他的論理和演算を多用する。本稿ではそのサイズは文脈から明らかであり単に“ \oplus ”で示す。

あるシステムから固定長文字列を逐次的に生成する場合について、その文字列生成システム(または、生成する文字列)の性質として“nonce”(ナ

ンス、と読む)を説明する。これは、生成時点より前には生成されたことがないような値を出力するものである。その例としてカウンタや時刻情報などがあるが、これらは無限にこの性質をもつものではない。そういう意味では乱数発生系列も確率的ではあるが、多くの場合において、“nonce”の性質を持っているといえる。

主に安全性の議論で参照される技術用語に、ランダム関数、擬似ランダム関数、及び擬似ランダム置換がある。これらの正確な定義は専門書、及び技術論文に譲るとしてここでは簡単にその説明をする。

ランダム関数とは、与えた入力に対してその出力が決定されるものの、その出力は、どんな情報からも推測できないランダムな値であるような関数のモデルである。このような関数は現実に存在するかどうかは別にして、そのような関数の振舞いをブロック暗号の性質に見立てて、利用モードの安全性を議論することがある。

しかし、ランダム関数は入力に対する出力がどのような方法を用いても推測できない、という性質は、現在のブロック暗号にそれを求めるのは無理がある。ブロック暗号には鍵入力がありこの鍵が求まってしまえば、どの入力がどの出力を出すかがわかってしまう。そこで、ある程度の時間、計算量をかけた上で破れるかもしれないようなランダム関数のモデルを擬似ランダム関数モデルという。これを証明などで扱う場合、パラメータがつく。

また、ブロック暗号は入力、出力の間に単射という性質がある。これ自身もランダム関数にはない特殊な性質であるので、実際のブロック暗号は、単なる擬似ランダム関数ではなく、さらに弱い擬似ランダム置換というモデルまで落して考えることが多い。ここで扱うモデルを擬似ランダム置換という。

ディスクセクタ暗号の部分などで、universal hash(汎用ハッシュ)と呼ばれる関数を考えることがある。これは、ある種の関数であって、ざっくり説明すると任意長の入力とパラメータから固定長の出力を出す関数であって、任意に固定した二つの入力が衝突する場合というのが、パラメータすべてのうちごくわずかであることが、どのような二入力メッセージについても言えるような関数である。ただし、パラメータを知っている攻撃者のようなものがいれば、衝突を作ることは必ずしも難しくない。

その他の用語、記号については以下のとおりに定義する:

ρ	ランダム関数モデル
π	擬似ランダム関数
$a \lll k$	レジスタ値 a を左に k ビットシフトする演算
$\text{msb}_k(a)$	レジスタ値 a の上位 k ビットの値
$\text{Enc}_K(\cdot)$	あるブロック暗号プリミティブの暗号化処理 (鍵 K)
$\text{Dec}_K(\cdot)$	あるブロック暗号プリミティブの復号処理 (鍵 K)

2.1 記法

A が集合である場合, $a \stackrel{R}{\leftarrow} A$ は A から a を一様ランダムに選ぶことをあらわす. A がアルゴリズムである場合, $a \leftarrow A$ は A の実行結果を a とする, ということであらわす. A が確率的アルゴリズムであれば, $a \stackrel{R}{\leftarrow} A$ と表記する. 整数 l に対し, $\{0, 1\}^l$ はすべての l ビット列の集合をあらわす. また, $\{0, 1\}^{\leq l}$ は, l ビット以下のすべてのビット列の集合をあらわす. 長さ 0 のビット列 ε もこれに含める. 同様に, $(\{0, 1\}^l)^+$ は, 長さが l の整数倍のすべてのビット列からなる集合をあらわす. すなわち, $(\{0, 1\}^l)^+ = \bigcup_{l'=1, 2, \dots} \{0, 1\}^{ll'}$ である. 同様に, $\{0, 1\}^*$ は, すべてのビット列の集合を表す. すなわち, $\{0, 1\}^* = \bigcup_{l=0, 1, 2, \dots} \{0, 1\}^l$ である. a と b が同じ長さのビット列であれば, $a \oplus b$ はそれらのビットごとの排他的論理和をあらわす. a がビット列のとき, $|a|$ は a のビット長をあらわす.

2.2 ブロック暗号

ブロック暗号 (block cipher) E とは, $E : \mathcal{K}_E \times \mathcal{M}_E \rightarrow \mathcal{M}_E$ なる関数である. \mathcal{K}_E は, 鍵空間とよばれ, $\mathcal{K}_E = \{0, 1\}^k$ のとき, k を鍵長という. \mathcal{M}_E は, メッセージ空間, もしくは平文空間とよばれ, $\mathcal{M}_E = \{0, 1\}^n$ のとき, n をブロック長という. ただし, すべての $K \in \mathcal{K}_E$ に対し, $E(K, \cdot)$ は \mathcal{M}_E 上の置換でなくてはならない. $E(K, \cdot)$ は \mathcal{M}_E 上の置換なので, その逆関数 $E^{-1}(K, \cdot)$ が存在する. すべての鍵 $K \in \mathcal{K}_E$ とすべての平文 $X \in \mathcal{M}_E$ に対し, $E^{-1}(K; E(K, X)) = X$ であり, すべての鍵 $K \in \mathcal{K}_E$ とすべての暗号文 $Y \in \mathcal{M}_E$ に対し, $E(K; E^{-1}(K, Y)) = Y$ である. 関数 $E(K, \cdot)$ を暗号化関数, 関数 $E^{-1}(K, \cdot)$ を復号関数という. それぞれ $E_K(\cdot), E_K^{-1}(\cdot)$ と表記する.

2.3 暗号化方式

暗号化方式はメッセージ秘匿のためのブロック暗号利用モードである。暗号化方式 ENC は、三つのアルゴリズム $ENC = (ENC-K, ENC-E, ENC-D)$ から成る。 $ENC-K$ を鍵生成アルゴリズム、 $ENC-E$ を暗号化アルゴリズム、 $ENC-D$ を復号アルゴリズムという。また、メッセージ空間 \mathcal{M}_{ENC} をもつ。

ここで、鍵生成アルゴリズム $ENC-K$ は確率的アルゴリズムであり、入力はなく、ランダムな鍵 K を出力する。 $K \stackrel{R}{\leftarrow} ENC-K$ と表記する。暗号化アルゴリズム $ENC-E$ は、確率的、決定的、状態をもつ、もしくは nonce を利用するアルゴリズムである。鍵 K とメッセージ $M \in \mathcal{M}_{ENC}$ を入力とし、暗号文 C を出力する。 $C \stackrel{R}{\leftarrow} ENC-E(K; M)$ や $C \leftarrow ENC-E(K; M)$ と表記する。また、乱数や状態を明示的に入力に示すこともある。

暗号化アルゴリズムが確率的アルゴリズムである場合、入力 (K, M) が与えられるたびに乱数を選び、それを用いて暗号文 C を計算する。アルゴリズムが呼び出されるたびに乱数を選びなおす。同じ入力で 2 度アルゴリズムを呼び出したとしても、同じ出力になるとは限らない。

暗号化アルゴリズムが状態をもつアルゴリズムである場合、まずある定められた方法に従って状態を初期化する。入力 (K, M) が与えられると、 (K, M) と現在の状態に応じて暗号文 C を計算し、状態を更新し、新しい状態を保持する。多くの場合、状態は単なるカウンタである。

暗号化アルゴリズムが nonce を用いるアルゴリズムである場合、メッセージごとに異なる値である nonce を用いる。メッセージを数えるカウンタは、メッセージごとに異なる値であるので、nonce として用いることができる。ただし、nonce はカウンタのように値が増える（あるいは減る）必要はなく、単に異なるメッセージに対しては、異なる値であればよい。

復号アルゴリズム $ENC-D$ は、決定的アルゴリズムであり、鍵 K と暗号文 C を入力とし、メッセージ M を出力する。 $M \leftarrow ENC-D(K; C)$ と表記する。

全ての鍵 K と全てのメッセージ M に対し、

$$ENC-D(K; ENC-E(K; M)) = M$$

でなければならない。

$ENC-E(K; \cdot)$ と $ENC-D(K; \cdot)$ を、 $ENC-E_K(\cdot)$ や $ENC-D_K(\cdot)$ と表記する。

例として、一般的な ECB, CBC, OFB, CFB, CTR [SP800-38A], ディスク暗号化用の NR [NR99], CMC [HR03b], 3GPP の f_8 [3GPPa, 3GPPb] などがある。

2.4 メッセージ認証コード

メッセージ認証コード MAC は、メッセージ認証のためのブロック暗号利用モードである。メッセージ認証コード MAC は、三つのアルゴリズム $MAC = (MAC-K, MAC-G, MAC-V)$ から成る。 $MAC-K$ を鍵生成アルゴリズム、 $MAC-G$ をタグ生成アルゴリズム、 $MAC-V$ を確認アルゴリズムという。また、メッセージ空間 \mathcal{M}_{MAC} とタグ空間 \mathcal{T}_{MAC} をもつ。

鍵生成アルゴリズム $MAC-K$ は確率的アルゴリズムであり、入力はなく、鍵 K を出力する。 $K \stackrel{R}{\leftarrow} MAC-K$ と表記する。

タグ生成アルゴリズム $MAC-G$ は、確率的、決定的、もしくは状態をもつアルゴリズムである。鍵 K とメッセージ $M \in \mathcal{M}_{MAC}$ を入力とし、タグ $T \in \mathcal{T}_{MAC}$ を出力する。 $T \stackrel{R}{\leftarrow} MAC-G(K; M)$ や $T \leftarrow MAC-G(K; M)$ と表記する。また、乱数や状態を明示的に入力に示すこともある。

$MAC-G$ が確率的アルゴリズムである場合、入力 (K, M) が与えられるたびに乱数を選び、それを用いてタグ T を計算する。アルゴリズムが呼び出されるたびに乱数を選びなおす。同じ入力で2度アルゴリズムを呼び出したとしても、同じ出力になるとは限らない。

$MAC-G$ が状態をもつアルゴリズムである場合、まずある定められた方法に従って状態を初期化する。入力 (K, M) が与えられると、 (K, M) と現在の状態に応じてタグ T を計算し、状態を更新し、新しい状態を保持する。多くの場合、状態は単なるカウンタである。

確認アルゴリズム $MAC-V$ は、決定的アルゴリズムであり、鍵 K 、メッセージ $M \in \mathcal{M}_{MAC}$ 、タグ $T \in \mathcal{T}_{MAC}$ を入力とし、 `accept or reject` を出力する。 $MAC-V(K; M; T) = \text{accept}$ や、 $MAC-V(K; M; T) = \text{reject}$ と表記する。

全ての鍵 K と全てのメッセージ M に対し、

$$MAC-V(K; M; MAC-G(K; M)) = \text{accept}$$

でなければならない。

$MAC-G(K; \cdot)$ と $MAC-V(K; \cdot; \cdot)$ を、 $MAC-G_K(\cdot)$ や $MAC-V_K(\cdot, \cdot)$ と表記する。

例として、CBC MAC [BKR00], EMAC [BB+95, PR00], RMAC [JJ+02a, JJ+02b, JJ+02c], XCBC [BR00], TMAC [KI03], OMAC [IK03a], XOR MAC [BGR95], XECB MAC [GD01a], PMAC [BR02], f_9 [3GPPa, 3GPPb] などがある。

2.5 メッセージ認証つき暗号化方式

メッセージ認証つき暗号化方式はメッセージ秘匿とメッセージ認証の機能を併せ持つブロック暗号利用モードである。メッセージ認証つき暗号化方式 AE は、三つのアルゴリズム $AE = (AE-K, AE-E, AE-D)$ から成る。 $AE-K$ を鍵生成アルゴリズム、 $AE-E$ を暗号化アルゴリズム、 $AE-D$ を復号アルゴリズムという。また、メッセージ空間 \mathcal{M}_{AE} をもつ。

ここで、鍵生成アルゴリズム $AE-K$ は確率的アルゴリズムであり、入力はなく、ランダムな鍵 K を出力する。 $K \stackrel{R}{\leftarrow} AE-K$ と表記する。暗号化アルゴリズム $AE-E$ は、確率的、決定的、状態をもつ、もしくは nonce を利用するアルゴリズムである。鍵 K とメッセージ $M \in \mathcal{M}_{AE}$ を入力とし、暗号文 C を出力する。 $C \stackrel{R}{\leftarrow} AE-E(K; M)$ や $C \leftarrow AE-E(K; M)$ と表記する。また、乱数や状態を明示的に入力に示すこともある。

復号アルゴリズム $AE-D$ は、決定的アルゴリズムであり、鍵 K と暗号文 C を入力とし、メッセージ M 、もしくは改ざん検出信号を出力する。 $M \leftarrow AE-D(K; C)$ や、改ざん検出信号 $\leftarrow AE-D(K; C)$ と表記する。

全ての鍵 K と全てのメッセージ M に対し、

$$AE-D(K; AE-D(K; M)) = M$$

でなければならない。

$AE-E(K; \cdot)$ と $AE-D(K; \cdot)$ を、 $AE-E_K(\cdot)$ や $AE-D_K(\cdot)$ と表記する。

例として、IACBC, IAPM [J01], OCB [RBBK01a], XCBC [GD01a], CCM [WHF02, J02], CWC [KVW03], EAX [BRW03] などがある。

3 各々の利用モードを定義する文書

利用モードは各種標準化や、学術文書において定義されることが多い。ここでは、利用モードを定義する文書についての紹介を行う。

3.1 商務省連邦情報処理規格 (FIPS), 特殊文書 (SP)

米国では政府などで用いる暗号技術の方式を FIPS (Federal Information Processing Standard, 商務省連邦情報処理規格) で定めている [WWW4]。FIPS は NIST (National Institute of Standards and Technology, 商務省技術標準局) [WWW5] で編集が行なわれ、管理されている。

3.1.1 ブロック暗号プリミティブ

本報告の主要な対象技術はブロック暗号利用モードである。ここではブロック暗号プリミティブの代表例としてFIPSで記載のブロック暗号のうちDES, AESについて、仕様の概要を紹介する。本報告で扱う利用モードの適用対象はこれらに限定される訳ではない。

Data Encryption Standard (DES) ブロック暗号に関するNISTの標準としては、DES(Data Encryption Standard, データ暗号化規格)がFIPS46(1977年1月15日)で定義されており、現在その改訂などによりFIPS46-3(2003年11月時点)が公開されている[FIPS46-3]。FIPS46-3では、DESのブロック暗号としての強度を高める目的でTDEA(Triple Data Encryption Algorithm, 三連DES)が定義されており、三つ鍵版(K_1, K_2, K_3)の定義をもとに、鍵利用オプションとして二個鍵版($K_1 = K_3$)やDESコンパクト版($K_1 = K_2 = K_3$)が定義されている。これが通称トリプルDES(T-DES, 3DES)である。

DESは鍵長64ビットであるが、そのうちパリティビット8ビットは暗号学的強度に寄与しないため、実質56ビットである。ブロックサイズは64ビットである。TDEAはブロックサイズは変わらず、実質鍵長が、三つ鍵版168ビット、二つ鍵版112ビットである。

Advanced Encryption Standard (AES) —SやTDEAの安全性への懸念を受けて、NISTは1997年からの標準化活動の結果として、2001年11月26日、AES(Advanced Encryption Standard, 次世代暗号標準)をFIPS197として定義した[FIPS197]。

AESはブロック長128ビットで、鍵長は128ビット、192ビット、256ビットの三つの鍵長の処理(AES-128, AES-192, AES-256)が定義されている。

3.1.2 ブロック暗号利用モード

歴史的に標準を紹介すると、NISTはDESをFIPS掲載してから間もなく、DESの利用方法を定めるDES利用モードをFIPS81で定義した(1980年12月2日)[FIPS81]。また、仕様書の誤植の変更として1981年11月20日にChange Noticeが発行された。FIPS81では、ECB, CBC, k -CFB, k -OFBの4つのモードが定義されている。ただし、この文書に対するChange Notice 2(1996年5月31日)において、 k -OFBに関しては $k < 64$ では使うべきでなくこれを以降サポートしないことが記載された。

Change Notice 3は64-bit OFBのテストベクトルのみ記載されている。同様にFIPS113では、DESを使ったメッセージ認証符号の生成方法としてCBC-MACを定義している[FIPS113]。

NISTは次にAESのための利用モードを定義するが、ここではFIPSではなくSpecial Publicationとしての発行が準備されている[WWW6]。2003年11月時点では、5つの秘匿に関する利用モードがSP800-38A(2001年12月版)として定義されている(2001年12月版)[SP800-38A]。これには、FIPS81で定義した4つのモードに加えて、CTRモードが追加されている。またOFBモードは安全性の観点からパラメータ k はブロックサイズのみとし、末端処理の定義を付け加えている。これら方式は、FIPS認定の任意のブロック暗号アルゴリズムに適用できると記載されている。

またブロック暗号からメッセージ認証符号を生成するためのモードについては、2005年11月現在、SP800-38B(2005年5月版)としてCMACを定義している[SP800-38B]。そして、もうひとつ、認証暗号(すなわち、復号時、暗号文の改竄を検出できる暗号処理)の標準は2005年11月現在、SP800-38CとしてCCMを定義している[SP800-38C]。

SP800-38A 暗号化方式 (ECB, CBC, CFB, OFB, CTR)

SP800-38B メッセージ認証コード (CMAC)

SP800-38C メッセージ認証つき暗号化方式 (CCM)

また、DESを定義するFIPS46-3でもANSI X9.52で定義される7つの利用モードの利用を認めている。7つとは、すべてTDEA用であって4つはECB, CBC, CFB, OFBであり、残りはANSI X9.52版CBC, CFB, OFBモードである(これらはインターリービング、すなわちパイプライン処理系にも適用できるような仕様変更がなされている)。

3.2 ISO/IEC

ISO(International Organization for Standardization, 国際標準化機構)、及びIEC(International Electrotechnical Commission, 国際電気標準会議)は一部の国際規格を共同で策定している。特に暗号技術に関する分野では、ISO/IEC JTC 1/SC27などで標準化会議が開かれ、暗号や情報セキュリティに関するISO/IEC標準文書が作成されている。

利用モードに関する標準化文書としては、IS 8372(64ビットブロック暗号利用モード)[ISO8372]、ISO 10116[ISO10116] (n ビットブロック暗号利用モード, 2002年6月26日)がある。IS 8372の記述は、ISO 10116に統合されることから、近い将来ISが抹消されることになる。現在文書中

には、ECB, CBC, CFB, OFB の4つのモードが定義されているが、次の改訂作業でCTR モードが新たに加わる方向で議論が進んでいる。

また、ISO の金融取引に関する標準化 TC68 では、8731-1 で CBC-MAC を定義している。

ISO 8372 64 ビットブロック暗号利用モードである。4つの暗号化方式 ECB, CBC, CFB, OFB を定めている。DES に関する FIPS 81 と ANSI X3.106 を一般化し、任意の 64 ビットブロック暗号を対象としたものになっている。

ISO 9797 メッセージ認証コードである。CBC MAC を定めている。同様の標準として、ISO 8731-1, ISO 9807, ANSI X9.9, ANSI X9.19 がある。

ISO 10116 ISO 8372 を n ビットブロック暗号について定めたものである。

ISO 8631-1 ISO の TC68 では金融サービスのためのセキュリティ標準を定めている。以下の標準を定めている。

ISO 8731-1 メッセージ認証コード, CBC MAC
ISO 10126 メッセージ暗号化

3.3 JIS

JIS(日本工業規格)は、JISC(Japanese Industrial Standard Committee, 日本工業標準調査会)が制定・改正を行なう日本の工業標準となる国家規格である。具体的には、JISCでの審議のあと、主務大臣により制定され、JSA(Japanese Standards Association, 日本規格協会)から発行される。

JISでの利用モードに関する規格として、JIS X 5052, JIS X 5003がある。前者はISO 8372ならびにANSI X3.106 (American National Standard for Information Systems – Data Encryption Algorithm – Modes of Operation)と同一であり、JIS X 5053はISO/IEC 10116と同一である。

3.4 金融に関するセキュリティ標準

ANSI X3 ANSI (American National Standards Institute) では、以下の2つの標準を定めている。

表 1: 秘匿のための利用モード一覧

略号	名前	日本語
2DEM	2D-Encryption Mode	二次元暗号
ABC	Accumulated Block Chaining	累積ブロック連鎖
CTR	Counter Mode Encryption	カウンタ
IGE	Infinite Garble Extention	無限改竄拡張

ANSI X3.92 FIPS 46 で定められている DES

ANSI X3.106 FIPS 81 で定められている DES の利用モード

ANSI X9 ANSI X9 シリーズでは、以下の標準を定めている。

ANSI X9.9 メッセージ認証コード, CBC MAC

ANSI X9.19 メッセージ認証コード, CBC MAC

ANSI X9.23 メッセージ暗号化

ANSI X9.52 Triple DES と利用モード

3.5 ANSI

ANSI(American National Standards Institute, アメリカ規格協会)[WWW7]では、主に ANSI X3.106, X3.92 で共通鍵暗号技術を標準化している。具体的には、ANSI X3.92 は DES を定義し、X3.106 でその利用モードを定義する。

3.6 AES 利用モード候補方式

NIST が AES の利用モードを策定する活動でも、いくつかの利用モードが提案された。2003 年 11 月時点での公開されている提案利用モードは表 1, 2, 3, 4 のとおり。

3.7 IEEE ディスクセクター暗号

IEEE(the Institute of Electrical and Electronics Engineers, Inc., 電気電子学会) の Security in Storage WG[WWW1]では、セクターレベルの記憶装置における機密情報を守る構想を定義し、暗号アルゴリズムや利用モードを定義している。2003 年 11 月まで、5 回の会合(2002 年 6 月

表 2: AES に提案された MAC 生成のための利用モード

略号	名前	日本語
OMAC	OMAC: One-Key CBC	一個鍵 CBC
PMAC	Parallelizable M-.A-.Code	並列 MAC
RMAC	Randomized MAC	攪拌 MAC
TMAC	Two-Key CBC-MAC	二個鍵 CBC-MAC
XCBC	Extended Cipher Block Chaining MAC	拡張 CBC-MAC
XECB	eXtended Electronic Code Book MAC	拡張 ECB-MAC

表 3: AES に提案された認証暗号のための利用モード

略号	名前
CCM	Counter with CBC-MAC
CWC	Carter Wegman with Counter
CS	Cipher-State
EAX	A Conventional Authenticated-Encryption Mode
GCM	Galois/Counter Mode
IACBC	Integrity Aware Cipher Block Chaining
IAPM	Integrity Aware Parallelizable Mode
OCB	Offset Codebook
PCFB	Propagating Cipher Feedback
XCBC	eXtended Cipher Block Chaining Encryption

表 4: AES に提案されたその他の利用モード

略号	名前	日本語
KFB	Key Feedback Mode	鍵フィードバック
AES-hash	AES-hash	AES ハッシュ

20日 New York/ 2002年10月10日 Ontario, Canada/ 2002年12月10日 Maryland/ 2003年4月10日 San Diego, CA/ 2003年8月21~22日 Goleta, CA) とワークショップ (SISW2003, 2003年10月31日 Washington D.C.) が開催された。

2005年11月時点では、AESを使ったLRWモード (Liskov-Rivest-Wagner) と、EMEの二つの操作モードの提案がドラフトとして公開されている。

3.8 NESSIE

NESSIE (New European Schemes for Signatures, Integrity, and Encryption) [WWW8] はヨーロッパで2000年1月に開始された3年間のプロジェクトで、ブロック暗号、メッセージ認証コード、公開鍵暗号、ハッシュ関数といった暗号プリミティブの評価を行うことを目的としている。2003年2月の最終報告書が公開され、メッセージ認証コードではEMACがportfolioに含まれた。

3.9 その他工業製品や業界標準などで利用されたもの

3GPP(3rd Generation Partnership Project) では、ブロック暗号KASUMI [3GPPb] 及びその利用モード [3GPPa] が作成されている。暗号化方式として f_8 が、メッセージ認証コードとして f_9 が策定されている。それぞれ従来よく知られた利用モードとは異なるものを用いている。

RFC2040ではブロック暗号RC5(TM)の利用方法として、CBCをベースにした末端処理つきCBCモードが記載されている。これはCTS (Cipher Text Stealing, 暗号文窃盗) と呼ばれている [RFC2040]。

また、Kerberos Version 4では、認証暗号の目的でPCBCが用いられていたが、安全性の観点で欠陥が見つかったため、Version 5では使われなくなった。

3.10 その他学術論文などに提案されたもの

主に学会でもブロック暗号の利用については議論されている。この中には、自己同期式利用モード各種 [M91, JR99, AGPS02] , iaPCBC[GD99], NCBC, RPCなどがある。

また、ブロック暗号の利用方法、という観点からもいくつかの提案があり、ブロック暗号を暗号学的一方向性ハッシュ関数に変換する利用モード

[BRS02, PGV94] や、ブロック暗号から、AONT(All-or-Nothing-Transform, 完全出鱈目変換) の手法を与える利用モード [R97], さらに、秘密でない乱数鍵が刺さったブロック暗号を鍵つきブロック暗号に変換する手法 [EM97] (さらにこれに対する安全性の検討 [D93]), 鍵長の短いブロック暗号の全数探索への強化方法 [KR96] (およびそれに関する検討 [M02]) などがある。

4 安全性の定義

利用モードの安全性の議論は 1990 年代から多く議論されるようになった。その大きな話題のひとつが、証明可能安全性に関する議論である。これは、内部で用いるブロック暗号を疑似ランダム置換 (PRP) としてモデル化しながら、利用モードが提供する機能を数学的に証明するものである。この利用モードにおける証明可能安全性についてより深く紹介する。

4.1 ブロック暗号の安全性

ブロック暗号の代表的な安全性の定義として、疑似ランダム置換としての安全性と強疑似ランダム置換としての安全性がある。

4.1.1 疑似ランダム置換族

ブロック暗号 $E : \mathcal{K}_E \times \mathcal{M}_E \rightarrow \mathcal{M}_E$ は、 \mathcal{M}_E 上の置換族 $\{E_K(\cdot) \in \text{Perm}(\mathcal{M}_E) \mid K \in \mathcal{K}_E\}$ と捉えることができる。ここで、 $\text{Perm}(\mathcal{M}_E)$ は \mathcal{M}_E 上のすべての置換の集合である。

直感的に、「あるブロック暗号が疑似ランダム置換族である」とは、適応的選択平文攻撃を行う任意の敵が、置換族 $\{E_K(\cdot) \in \text{Perm}(\mathcal{M}_E) \mid K \in \mathcal{K}_E\}$ と \mathcal{M}_E 上のすべての置換の集合 $\text{Perm}(\mathcal{M}_E)$ を区別できないことをいう。

より厳密には、敵 A として、オラクルにアクセスできるアルゴリズムを考える。何回かの質問の後、 A は 1 ビットを出力する。ブロック暗号 $E : \mathcal{K}_E \times \mathcal{M}_E \rightarrow \mathcal{M}_E$ の、敵 A に対する、疑似ランダム置換としての安全性は、アドバンテージ $\text{Adv}_E^{\text{PRP}}(A)$ によって評価される。ここで、

$$\text{Adv}_E^{\text{PRP}}(A) \stackrel{\text{def}}{=} \left| \Pr(K \stackrel{R}{\leftarrow} \mathcal{K}_E : A^{E_K(\cdot)} = 1) - \Pr(P \stackrel{R}{\leftarrow} \text{Perm}(\mathcal{M}_E) : A^{P(\cdot)} = 1) \right|$$

と定義され、 $A^{E_K(\cdot)}$ は質問 X に対し、 $Y = E_K(X)$ を返すオラクル $E_K(\cdot)$ を持つ敵 A を表し、 $A^{P(\cdot)}$ は質問 X に対し、 $Y = P(X)$ を返すオラクル $P(\cdot)$ を持つ敵 A を表す。特に断りがなければ、質問は適応的に行う。すなわち、ある質問に対する答えを得た後、次の質問を行う。

$\text{Perm}(\mathcal{M}_E)$ から一様ランダムに選ばれた P を \mathcal{M}_E 上のランダム置換、あるいは単に、ランダム置換という。

計算量理論的安全性 上記の定義はある一つの敵に対する評価である。一般的に、敵が利用できる資源をパラメータにし、そのパラメータを利用するすべての敵の最大のアドバンテージを考える。ブロック暗号の擬似ランダム置換族としての安全性を考える場合に扱う資源は、実行時間 t とオラクルへの質問回数 q である。ここで、実行時間に関しては、ある計算のモデルが固定されているとする。その単位時間によって、ランダムに K を選ぶ時間や、 $E_K(X)$ の計算にかかる時間があらかわせるものとする。また、実行時間 t には、 A の記述に要する長さ (A を記述するプログラムの長さ) が含まれているものとし、また、 A の実行に関するすべての時間が含まれる。これにはランダムに K を選ぶ時間や、(オラクルとの) 入出力にかかる時間、等も含まれる。以降のすべての実行時間 t は同様に定義される。

$$\text{Adv}_E^{\text{PRP}}(t, q) \stackrel{\text{def}}{=} \max_A \{ \text{Adv}_E^{\text{PRP}}(A) \}$$

と定義される。ただし、最大値は実行時間 t 、オラクルへの質問回数 q のすべての敵 A についてとる。

この定義においては、正確には「安全な擬似ランダム置換族」という概念は存在しない。すべてのブロック暗号 E は、ある大きさの $\text{Adv}_E^{\text{PRP}}(t, q)$ をもつ置換族である。「 E が安全な擬似ランダム置換族である」や、「 E が擬似ランダム置換族である」という表現や仮定は、「適当に大きい t と q に対し、 $\text{Adv}_E^{\text{PRP}}(t, q)$ が十分小さい」ということを意図している。厳密な安全性の定理を言う場合にはこれらの表現は用いない。

4.1.2 強擬似ランダム置換族

「あるブロック暗号が強擬似ランダム置換族である」とは、適応的選択平文暗号文攻撃を行う任意の敵が、置換族 $\{E_K(\cdot) \in \text{Perm}(\mathcal{M}_E) \mid K \in \mathcal{K}_E\}$ と \mathcal{M}_E 上のすべての置換の集合 $\text{Perm}(\mathcal{M}_E)$ を区別できないことをいう。

より厳密には、敵 A として、2つのオラクルにアクセスできるアルゴリズムを考える。何回かの質問の後、 A は1ビットを出力する。ブロッ

ク暗号 $E : \mathcal{K}_E \times \mathcal{M}_E \rightarrow \mathcal{M}_E$ の、敵 A に対する、強擬似ランダム置換としての安全性は、アドバンテージ $\text{Adv}_E^{\text{sprp}}(A)$ によって評価される。ここで、

$$\text{Adv}_E^{\text{sprp}}(A) \stackrel{\text{def}}{=} |\Pr(K \stackrel{R}{\leftarrow} \mathcal{K}_E : A^{E_K(\cdot), E_K^{-1}(\cdot)} = 1) - \Pr(P \stackrel{R}{\leftarrow} \text{Perm}(\mathcal{M}_E) : A^{P(\cdot), P^{-1}(\cdot)} = 1)|$$

と定義され、 $A^{E_K(\cdot), E_K^{-1}(\cdot)}$ は質問 X に対し、 $Y = E_K(X)$ を返す暗号化オラクル $E_K(\cdot)$ と、質問 Y に対し、 $X = E_K^{-1}(Y)$ を返す復号オラクル $E_K^{-1}(\cdot)$ を持つ敵 A を表し、 $A^{P(\cdot), P^{-1}(\cdot)}$ は質問 X に対し、 $Y = P(X)$ を返す暗号化オラクル $P(\cdot)$ と、質問 Y に対し、 $X = P^{-1}(Y)$ を返す復号オラクル $P^{-1}(\cdot)$ を持つ敵 A を表す。特に断りがなければ、質問は適応的に行う。すなわち、ある質問に対する答えを得た後、次の質問を行う。

計算量理論的安全性 ブロック暗号の強擬似ランダム置換族としての安全性を考える場合に扱う資源は、実行時間 t 、暗号化オラクルへの質問回数 q_e 、復号オラクルへの質問回数 q_d である。

$$\text{Adv}_E^{\text{sprp}}(t, q_e, q_d) \stackrel{\text{def}}{=} \max_A \{ \text{Adv}_E^{\text{sprp}}(A) \}$$

と定義される。ただし、最大値は実行時間 t 、暗号化オラクルへの質問回数 q_e 、復号オラクルへの質問回数 q_d のすべての敵 A についてとる。

一般に、「 E が安全な強擬似ランダム置換族である」や、「 E が強擬似ランダム置換族である」という表現は、「適当に大きい t, q_e, q_d に対し、 $\text{Adv}_E^{\text{sprp}}(t, q_e, q_d)$ が十分小さい」ということを意図している。

4.1.3 上記以外のブロック暗号の安全性

上記以外にもブロック暗号の安全性定義がいくつか存在する。鍵関連攻撃を考慮した安全性定義 [BK03] などがこれに含まれる。

また、理想的ブロック暗号モデルというブロック暗号のモデル化がある。ハッシュ関数のランダムオラクルに対応するものであり、RMAC [JJ+02a, JJ+02b] の安全性解析に用いられた。

4.2 秘匿の安全性

暗号学における利用モードに関する安全性とは、想定した攻撃者に対するメカニズムの性質を議論する。よって、攻撃者をきちんと定義する必要がある。

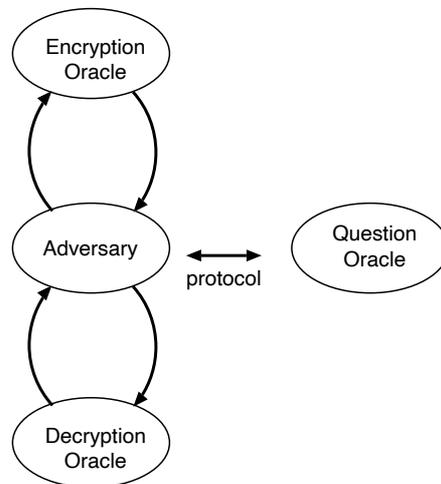


図 4: 証明可能安全性における攻撃者の例 (選択暗号文攻撃)

ここで考える攻撃者は限られた能力をもつものであつて、指定されたこと以外の動作や、動作から得られる以外の情報の獲得は考えられていない。研究として、なるべく現実に近い、すなわち能力が高く、さまざまな能力をもつ攻撃者を検討する方向はあるが、完全ではない。

ここで考える攻撃者は、まず最低質問オラクルとのゲームを1回だけ行なう。また、攻撃者の能力としてそれとは別にさまざまなオラクルへの通信が可能である。

多くのモデルで、攻撃者が暗号化オラクルに対するアクセスを許している。これは攻撃者が、任意の(あとあと都合のよい)平文を生成するとそれに対する暗号文を教えてもらえるものである。これを繰り返すことにより攻撃者が知識を獲得することが許される。

また、いくつかの暗号スキームに対する証明可能安全性では復号オラクル(暗号化と同様に、今度は暗号文に対して(必要であれば改竄検知をし、もし問題なければ)平文を返答教えてくれるもの)を考える場合もある。

もし、攻撃者の能力として、選択平文攻撃を考えるならば、その安全性の検討では、攻撃者の暗号化オラクルのアクセスを検討する。また、選択暗号文攻撃では(通常、選択平文攻撃の能力を含んだ定義を考えることが多いので)暗号化オラクルに加えて、復号オラクルを含めた評価を行なう。

以上の攻撃者の能力を特定した上で、スキームについての安全性を検討する。安全性は、秘匿、認証に分けて扱う。認証暗号は、これら秘匿、認証の両方の安全性を達成している。

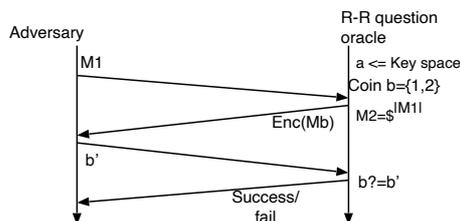


図 5: Real-or-Random notion を定義するゲームのプロトコル

まず、秘匿からはじめる。この分野における、秘匿の定義は厳密には複数存在する。しかし、その多くが計算量的に等価であることが知られているため、実質的にはひとつの安全性を達成すれば一般にいわれる秘匿の種類はある程度保証できる。

Real-or-Random (暗号文-乱数処理文識別) この秘匿に関する安全性を大雑把に理解するなら、攻撃者の目標は次の二つの暗号文を見分けることである、(1) 攻撃者自身が作成した平文に対応する暗号文、(2) その平文と同じ長さなだけで全然関係のない乱数を暗号化したもの。正式には、オラクルが行なう2種類のゲームで考える。オラクルはそれぞれのゲーム開始後には鍵を決定する。そして、攻撃者からメッセージ受信を待つ。ゲーム1では、メッセージを受信したら、さきほど決定した鍵で暗号化し、その結果を送信する。ゲーム2では、メッセージを受信しても、単にそれと同じ長さの乱数を発生し、それを送信する。

ある暗号化スキームが(ある条件下で、例えば選択平文攻撃などで) Real-or-Random で安全であるとは、(その条件が許される) どのような現実的な攻撃者も、ゲーム1とゲーム2を有意な確率で区別することが難しいことをいう。

定義 4.1 (Real-or-Random). 暗号化スキーム $\Pi = (\mathcal{E}, \mathcal{D}, \mathcal{K})$ が *Real-or-Random* の意味で $(t, q, \mu; \epsilon)$ -安全であるとは、次で指定される任意の攻撃者の利得 Adv_A^{rr} について下記が成り立つことである。攻撃者は、最大時間 t の間動作し、最大 q 回のオラクル質問(ここでは暗号化オラクルへの質問)を行ない、これらの質問の長さが最大 μ ビットであるような攻撃者である。

$$\text{Adv}_A^{\text{rr}} = \Pr[a \leftarrow \mathcal{K} : A^{\mathcal{E}_a(\cdot)} = 1] - \Pr[a \leftarrow \mathcal{K} : A^{\mathcal{E}_a(\mathcal{S}(\cdot))} = 1] \leq \epsilon.$$

Left-or-Right (左右平文暗号文識別) この秘匿に関する安全性でも二つのゲームを考える。質問オラクルへの攻撃者からの入力、長さが同じ

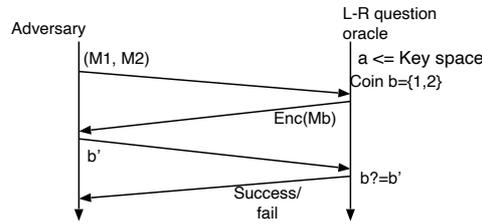


図 6: Left-or-Random notion を定義するゲームのプロトコル

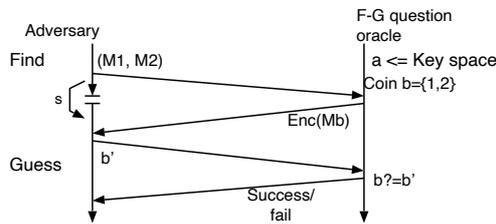


図 7: Find-then-Guess notion を定義するゲームのプロトコル

な平文のペアである (これらのペアが異なることが厳密には記載されていないが, 同じであれば攻撃者が不当に利得を得る可能性があるので一般的には異なるもののみを考える). オラクルは, ゲーム開始後には鍵を決定する. そして, 攻撃者から二つの同じ長さのメッセージペア (M_1, M_2) の受信を待つ. メッセージを受信したら, ゲーム 1 では M_1 を, ゲーム 2 では M_2 をそれぞれ, 先ほど生成した鍵で暗号化し, その結果を送信する.

ある暗号化スキームが (ある条件下で, 例えば選択平文攻撃などで) Left-or-Right で安全であるとは, (その条件が許される) どのような現実的な攻撃者も, ゲーム 1 とゲーム 2 を有意な確率で区別することが難しいことをいう.

定義 4.2 (Left-or-Right). 暗号化スキーム $\Pi = (\mathcal{E}, \mathcal{D}, \mathcal{K})$ が *Left-or-Right* の意味で $(t, q, \mu; \epsilon)$ -安全であるとは, 次で指定される任意の攻撃者の利得について下記が成り立つことである. 攻撃者は, 最大時間 t の間動作し, 最大 q 回のオラクル質問 (ここでは暗号化オラクルへの質問) を行ない, これらの質問の長さが最大 μ ビットであるような攻撃者である (ただし, 質問オラクルへのメッセージペア, (M_1, M_2) は同じ長さとする).

$$\text{Adv}_A^{\text{lr}} = \Pr[a \leftarrow \mathcal{K} : A^{\mathcal{E}_a(\text{left}(\cdot, \cdot))} = 1] - \Pr[a \leftarrow \mathcal{K} : A^{\mathcal{E}_a(\text{right}(\cdot, \cdot))} = 1] \leq \epsilon.$$

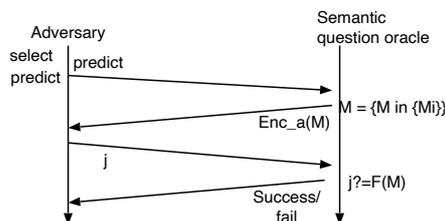


図 8: Semantic-security notion を定義するゲームのプロトコル

Find-then-Guess (発見-推測識別) Find-then-Guess は [GM84, MRS88] で扱っている多項式計算量的安全性の言い替えである。ここでは攻撃者は二つのステージを考える。第一の find ステージでは、攻撃者は最終的に同じ長さのメッセージペア (M_1, M_2) を生成するが、その目的は次のステージでこれらの暗号文を区別することである。また、この間に攻撃者は知識を蓄えることができ、最終的にあとで使う知識 s を生成してこのステージを終了する。

もうひとつの guess ステージでは、質問オラクルから暗号文 C を受信する。 C はさきほどの (M_1, M_2) どちらかの暗号文である。攻撃者は知識 s を知っている。ここで、その暗号文 C がどちらの平文のものであるかを定めることができれば、「攻撃者の勝ち」とする。

ある暗号化スキームが(ある条件下で、例えば選択平文攻撃などで) Find-then-Guess で安全であるとは、(その条件が許される) どのような現実的な攻撃者も、それらを有意な確率で区別することが難しいことをいう。

定義 4.3 (Find-then-Guess). 暗号化スキーム $\Pi = (\mathcal{E}, \mathcal{D}, \mathcal{K})$ が *Find-then-Guess* の意味で $(t, q, \mu; \epsilon)$ -安全であるとは、次で指定される任意の攻撃者の利得について下記が成り立つことである。攻撃者は、最大時間 t の間動作し、最大 q 回のオラクル質問(ここでは暗号化オラクルへの質問)を行ない、これらの質問の長さが最大 μ ビットであるような攻撃者である。

$$\text{Adv}_A^{\text{fg}} = 2 \cdot \Pr[a \leftarrow \mathcal{K} : (M_1, M_2, s) \leftarrow A^{\mathcal{E}_a(\cdot)}(\text{find}); b \leftarrow \{1, 2\}; C \leftarrow \mathcal{E}_a(M_b) : A^{\mathcal{E}_a(\cdot)}(\text{guess}, C, s) = b] - 1 \leq \epsilon.$$

Semantic (意味抽出) Goldwasser と Micali[GM84] では、semantic security を「暗号文が与えられてから平文に関してわかる情報というのは、暗号文がなくともわかるものだけだ」と説明している。ここでの semantic は公開鍵暗号における semantic security をそのまま適応する。 f を平文を引数にとることができる関数とする。この関数は、攻撃者が(暗号文から)

知ろうとしている情報の種類を表していると考えることができる。平文空間は確率的な分布を取るものとして考える。任意の整数 m に対して、「平文空間における m 分布」とは、 m ビット以下の文字列で代表される平文空間上の確率分布、の集合 $\mathcal{M} = \{\mathcal{M}_\gamma\}_{\gamma \in \{0,1\}^{\leq m}}$ とし、すべての \mathcal{M}_γ が有効 (valid) とする。ここで、有効とはすべての確率分布 \mathcal{M}_γ について、確率が非 0 の文字列すべてが同じ長さであり、その長さは最大で m である、ということの意味する。 $p_{f, \mathcal{M}_\gamma}^* = \max_{C^*} \{\Pr[M \leftarrow \mathcal{M}_\gamma : f(M) = C^*]\}$ と定義する。これは平文の確率分布でもっともありうる $f(\cdot)$ 値である。

攻撃者は二つのステージを考える。第一の select ステージでは、攻撃者は都合の良い平文分布 \mathcal{M}_γ を生成する。もうひとつの predict ステージでは、質問オラクルが、指定された平文分布に従って無作為にメッセージ M を生成し、暗号文 C を送信する。攻撃者はこれを受信し、 $f(M)$ 値を予想しようとする。

ある暗号化スキームが (ある条件下で、例えば選択平文攻撃などで) Semantic で安全であるとは、(その条件が許される) 関数 f と分布 \mathcal{M} に対して、どのような現実的な攻撃者も、 $p_{f, \mathcal{M}_\gamma}^*$ を越える確率で $f(M)$ を予想することができない、ことをいう。

従来 (つまり公開鍵暗号で議論されていたところ) の定義では、この条件はすべての関数 f について成り立つ必要があった。共通鍵暗号においては、関数 f と確率分布 \mathcal{M} がパラメータとする。このことで、ある特殊な平文の性質が、ある特別な分布において、ちゃんと情報が隠れているか/いないかを議論できる。

定義 4.4 (Semantic). 関数 f を、平文空間を入力としてなにかしらのバイナリ文字列を出力する関数とする。 $\mathcal{M} = \{\mathcal{M}_\gamma\}_{\gamma \in \{0,1\}^{\leq m}}$ を平文空間における m 分布とする。

暗号化スキーム $\Pi = (\mathcal{E}, \mathcal{D}, \mathcal{K})$ が Semantic の意味で f と \mathcal{M} に対して $(t, q, \mu; \epsilon)$ -安全であるとは、次で指定される任意の攻撃者の利得について下記が成り立つことである。攻撃者は、最大時間 t の間動作し、最大 q 回のオラクル質問 (ここでは暗号化オラクルへの質問) を行ない、これらの質問の長さが最大 μ ビットであるような攻撃者である。

$$\text{Adv}_A^{\text{sm}}(f, \mathcal{M}) = \mathbf{E}[a \leftarrow \mathcal{K} : (\gamma, s) \leftarrow A^{\mathcal{E}_a(\cdot)}(\text{select}) : \alpha(a, \gamma, s)] \leq \epsilon.$$

ここで

$$\alpha(a, \gamma, s) = \Pr[M \leftarrow \mathcal{M}_\gamma; C \leftarrow \mathcal{E}_a(M) : A^{\mathcal{E}_a(\cdot)}(\text{predict}, C, s) = f(M)] - p_{f, \mathcal{M}_\gamma}^*.$$

Ciphertext-Random (暗号文-乱数) 近年の認証暗号など新しい利用モードの提案では、Real-or-Random とは異なる、暗号文-乱数不可識別性

で秘匿の安全性を証明する利用モードがある。Real-or-Random に酷似するため詳細は省略する。

Real-or-Random では、Game 2 において、乱数の暗号文を返答していた。この定義では、乱数そのものを返信するプロトコルでゲームをする。

4.3 従来の秘匿定義の関係

上記、秘匿の定義のうち前4つの定義については、[BDJR97] で詳細に扱われており、同じ文献でこれら4つの定義間の関係が明らかにされている。

4つの定義で最強とされるものは、Left-or-Right と Real-or-Random である。これらは安全性パラメータも損なわない多項式還元が実現されており、他の2つの定義へも効率的に還元できる。すなわち秘匿の定義としては最強の定義である。

これらに対して Find-then-Guess と Semantic については、安全性パラメータが多少欠損するものの、これらが言えれば上記2方式の安全性も保証することができる。

以上により、上記4つの定義のどれかを達成していれば共通鍵暗号における秘匿は十分なレベルが達成できていると言える。

4.4 メッセージ認証コードの安全性

メッセージ認証コード $MAC = (MAC-K, MAC-G, MAC-V)$ の安全性には、弱偽造不可能性と強偽造不可能性がある。

どちらの場合も、敵 A として、タグ生成オラクルと確認オラクルにアクセスできるアルゴリズムを考える。 $A^{MAC-G_K(\cdot), MAC-V_K(\cdot, \cdot)}$ は、メッセージ M に対し、タグ $T = MAC-G_K(M)$ を返すタグ生成オラクル $MAC-G_K(\cdot)$ と、メッセージ、タグのペア (M, T) に対し、accept or reject = $MAC-V_K(M, T)$ を返す確認オラクル $MAC-V_K(\cdot, \cdot)$ をもつ敵をあらわす。質問は適応的に行う。すなわち、ある質問に対する答えを得た後、次の質問を行う。

4.4.1 弱偽造不可能性

弱偽造不可能性の意味でメッセージ認証コード $MAC = (MAC-K, MAC-G, MAC-V)$ を破ろうとする敵 A が、タグ生成オラクルに q 個のメッセージ M_1, \dots, M_q を質問し、その答え T_1, \dots, T_q を得たとする。また、確認オ

ラクルに q' 個のメッセージ, タグのペア $(M'_1, T'_1), \dots, (M'_{q'}, T'_{q'})$ を質問したとする.

ある i に対し, $MAC\text{-}\mathcal{V}_K(M'_i, T'_i) = \text{accept}$ であり, $M'_i \notin \{M_1, \dots, M_j\}$ であれば, A は弱偽造不可能性の意味で偽造に成功した, という. ここで, $\{M_1, \dots, M_j\}$ は, (M'_i, T'_i) を確認オラクルに質問する以前に, タグ生成オラクルに送った質問である.

直感的には, 見たことのないメッセージに対するタグを出力できたらなば, 偽造に成功したことになる.

メッセージ認証コード $MAC = (MAC\text{-}\mathcal{K}, MAC\text{-}\mathcal{G}, MAC\text{-}\mathcal{V})$ の, 敵 A に対する, 弱偽造不可能性の意味での安全性は, アドバンテージ $\text{Adv}_E^{\text{w-uf}}(A)$ によって評価される. ここで,

$$\text{Adv}_{MAC}^{\text{w-uf}}(A) \stackrel{\text{def}}{=} \Pr(K \xleftarrow{R} MAC\text{-}\mathcal{K} : A^{MAC\text{-}\mathcal{G}_K(\cdot), MAC\text{-}\mathcal{V}_K(\cdot, \cdot)} \text{が弱偽造不可能性の意味で偽造に成功})$$

と定義される.

計算量理論的安全性 メッセージ認証コード $MAC = (MAC\text{-}\mathcal{K}, MAC\text{-}\mathcal{G}, MAC\text{-}\mathcal{V})$ の, 弱偽造不可能性の意味での安全性を考える場合に扱う資源は, 実行時間 t , タグ生成オラクルへの質問回数 q , それら質問の長さ σ (ビット単位, もしくはブロック単位), 確認オラクルへの質問回数 q' , それら質問の長さ σ' (ビット単位, もしくはブロック単位) である. 実行時間 t はブロック暗号と同様に定義される.

$$\text{Adv}_{MAC}^{\text{w-uf}}(t, q, \sigma, q', \sigma') \stackrel{\text{def}}{=} \max_A \{ \text{Adv}_{MAC}^{\text{w-uf}}(A) \}$$

と定義される. ただし, 最大値は実行時間 t , タグ生成オラクルへの質問回数 q , それら質問の長さ σ , 確認オラクルへの質問回数 q' , それら質問の長さ σ' のすべての敵 A についてとる.

4.4.2 強偽造不可能性

強偽造不可能性の意味でメッセージ認証コード $MAC = (MAC\text{-}\mathcal{K}, MAC\text{-}\mathcal{G}, MAC\text{-}\mathcal{V})$ を破ろうとする敵 A が, タグ生成オラクルに q 個のメッセージ M_1, \dots, M_q を質問し, その答え T_1, \dots, T_q を得たとする. また, 確認オラクルに q' 個のメッセージ, タグのペア $(M'_1, T'_1), \dots, (M'_{q'}, T'_{q'})$ を質問したとする.

ある i に対し, $MAC\text{-}\mathcal{V}_K(M'_i, T'_i) = \text{accept}$ であり, $(M'_i, T'_i) \notin \{(M_1, T_1), \dots, (M_j, T_j)\}$ であれば, A は強偽造不可能性の意味で偽造に成功した,

という。 $\{(M_1, T_1), \dots, (M_j, T_j)\}$ は、 (M'_i, T'_i) を確認オラクルに質問する以前に、タグ生成オラクルに送った質問とその答えである。

直感的には、見たことのないメッセージ、タグのペアを出力できたらなば、偽造に成功したことになる。タグが異なっていれば、メッセージ自体は見たことがあってもよい。

メッセージ認証コード $MAC = (MAC-K, MAC-G, MAC-V)$ の、敵 A に対する、強偽造不可能性の意味での安全性は、アドバンテージ $\text{Adv}_E^{\text{s-uf}}(A)$ によって評価される。ここで、

$$\text{Adv}_{MAC}^{\text{s-uf}}(A) \stackrel{\text{def}}{=} \Pr(K \xleftarrow{R} MAC-K : A^{MAC-G_K(\cdot), MAC-V_K(\cdot, \cdot)} \text{ が強偽造不可能性の意味で偽造に成功})$$

と定義される。

計算量理論的安全性　メッセージ認証コード $MAC = (MAC-K, MAC-G, MAC-V)$ の、強偽造不可能性の意味での安全性を考える場合に扱う資源は、弱偽造不可能性の場合と同様である。

$$\text{Adv}_{MAC}^{\text{s-uf}}(t, q, \sigma, q', \sigma') \stackrel{\text{def}}{=} \max_A \{ \text{Adv}_{MAC}^{\text{s-uf}}(A) \}$$

と定義される。ただし、最大値は実行時間 t 、タグ生成オラクルへの質問回数 q 、それら質問の長さ σ 、確認オラクルへの質問回数 q' 、それら質問の長さ σ' のすべての敵 A についてとる。

4.4.3 $MAC-G$ が決定的アルゴリズムである場合の安全性

$MAC-G$ が決定的アルゴリズムの場合、弱偽造不可能性の意味での安全性と強偽造不可能性の意味での安全性は同一の定義となる。また、この場合、タグ生成オラクルが確認オラクルのかわりになり得る。すなわち、タグ生成オラクルに M_i を質問し、 T_i を得たなら、確認オラクルは質問 (M_i, T_i) に対しては `accept` を返し、質問 (M_i, T'_i) (ただし $T'_i \neq T_i$) に対しては `reject` を返す。したがって、 q' と σ' のパラメータを用いなくて、 q と σ にこれらを含めるのが一般的である。 $MAC-G$ が決定的アルゴリズムの場合、弱偽造不可能性と強偽造不可能性とを区別せず、単に偽造不可能性という。

敵 A として、タグ生成オラクルにアクセスできるアルゴリズムを考える。 $A^{MAC-G_K(\cdot)}$ は、メッセージ M に対し、タグ $T = MAC-G_K(M)$ を返すタグ生成オラクル $MAC-G_K(\cdot)$ をもつ敵をあらわす。質問は適応的に行う。すなわち、ある質問に対する答えを得た後、次の質問を行う。

偽造不可能性の意味でメッセージ認証コード $MAC = (MAC-K, MAC-G, MAC-V)$ を破ろうとする敵 A がタグ生成オラクルにメッセージ M_1, \dots, M_j を質問し、その答え T_1, \dots, T_j を得たとする。タグ生成オラクルへの質問の途中、 A は偽造文 (M_{j+1}, T_{j+1}) を出力する。

$MAC-V_K(M_{j+1}, T_{j+1}) = \text{accept}$ であり、 $M_{j+1} \notin \{M_1, \dots, M_j\}$ であれば、 A は偽造不可能性の意味で偽造に成功した、という。 $\{M_1, \dots, M_j\}$ は、 (M_{j+1}, T_{j+1}) を出力する以前に、タグ生成オラクルに送った質問である。ある偽造文が reject された場合でも、 A はさらにタグ生成オラクルに質問を続け、あたらな偽造文を出力してよい。ただし、 (M_{j+1}, T_{j+1}) はタグ生成オラクルに対する質問として数える。

直感的には、見たことのないメッセージに対するタグを出力できたらなば、偽造に成功したことになる。

メッセージ認証コード $MAC = (MAC-K, MAC-G, MAC-V)$ の、敵 A に対する、偽造不可能性の意味での安全性は、アドバンテージ $\text{Adv}_{MAC}^{\text{mac}}(A)$ によって評価される。ここで、

$$\text{Adv}_{MAC}^{\text{mac}}(A) \stackrel{\text{def}}{=} \Pr(K \stackrel{R}{\leftarrow} MAC-K : A^{MAC-G_K(\cdot)} \text{ が偽造不可能性の意味で偽造に成功})$$

と定義される。

計算量理論的安全性 メッセージ認証コード $MAC = (MAC-K, MAC-G, MAC-V)$ の、偽造不可能性の意味での安全性を考える場合に扱う資源は、実行時間 t 、タグ生成オラクルへの質問回数 q (M' を含む)、それら質問の長さ σ (ビット単位、もしくはブロック単位、 M' の長さも含む) である。実行時間 t はブロック暗号と同様に定義される。

$$\text{Adv}_{MAC}^{\text{mac}}(t, q, \sigma) \stackrel{\text{def}}{=} \max_A \{ \text{Adv}_{MAC}^{\text{mac}}(A) \}$$

と定義される。ただし、最大値は実行時間 t 、タグ生成オラクルへの質問回数 q 、それら質問の長さ σ のすべての敵 A についてとる。

4.4.4 上記以外の安全性

上記以外にもいくつかの安全性定義が存在する。それらについては、そのつと説明をする。

4.5 攻撃者の能力

安全性の証明を考える上で攻撃者の能力を正確に決める必要がある。これについては、暗号化(秘匿の利用モード, ならびに認証暗号の利用モード)と認証(MAC生成のモード)で独立に考える。

暗号における証明可能安全性では、攻撃者の能力として

- A 攻撃者自身で都合良く選んだ平文に対して、それに対応する暗号文を知ることができる,
- B 攻撃者自身で都合良く選んだ暗号文に対して、それに対応する平文を知ることができる.

の二つの能力を考える。そして暗号が扱われる現実世界や、これまで提案されてきた利用モードの性質から、現状(B)のみが許されるような攻撃者は考えない。よって、暗号の安全性の前提となる攻撃者の種類は(A)のみを対象とした場合(選択平文攻撃)か、もしくは(A)(B)両方が可能な攻撃者を対象とした場合(選択暗号文攻撃)のふたとおりどちらかである。

4.6 証明可能安全性の仮定

証明可能安全性と現実での暗号利用には大きな差がある。その差に関する研究結果もいくつか知られてきているが、それがすべてではない。

まず、初期値に関する議論がある。これらすべての証明可能安全性において、初期値を正しく生成する必要がある。しかし、それに必要とされる乱数性や信頼性(カウンタのリセットを防止するメカニズムなど)を現実的に暗号に利用するのは多くの場合困難である。

次に攻撃者の能力である。多くの秘匿に関する証明可能安全性は、自分で生成した暗号文に対応する平文の情報を知ることができないことになっている。しかし、暗号文の改竄や、あるいは通信ノイズがあるような通信路の暗号処理では、攻撃者にいかなる復号結果を渡してはならない。現に、安易なチェックサムを用意してしまった暗号方式から、チェックサムの合否を用いて平文を読みとる攻撃手法が発見された事例がある[V02]。これについてはあとで述べる。

最後に、攻撃者の暗号文を取得できる能力は、メッセージ単位に限定されている、という点である。場合によっては、メッセージという単位よりもより細かい単位(例えばブロック単位など)で攻撃を組み立てる攻撃者が存在するかもしれない。また、このような、攻撃が可能である場合、証明可能だった安全性が崩れる例が知られている[JMV02]。

4.7 利用モードに対する攻撃

ここまで議論したような証明可能安全性は、現実世界が完全にモデル化されたとおりに動く場合に限って現実的に信頼できる。しかしながら、実際にはそうでない場合がある。安全性に関する議論の最後に、これまでに知られている攻撃関連の話題をいくつか紹介する。

[V02]では、CBCモードに対する攻撃を示している。不適切な実装として、改竄検知を目的としたパディングとそれによる改竄検知がいくつかの標準化で実装された。この改竄検知機能を利用することで、本来秘匿されるはずの情報を読みとることができる。CBCモードは秘匿にのみ使われるべきであり、不用意に、利用モードの範囲外のことを行なうと、もとの安全性も崩れる典型的な例である。

[JMV02]では、CBC, IACBC, (そして公開鍵とのハイブリッド暗号GEM)に対する秘匿に関する攻撃の可能性を示している。ここでは現実的には考えにくいほど強力な攻撃者を想定するが、攻撃は攻撃である。従来安全性評価は、攻撃者の判定したいメッセージ対はメッセージストリームを最後まで消化した上で暗号文の最初のブロックが生成されていた。ところが、オンライン処理を用いるときなどは、かならずしも暗号文出力のために、メッセージを受信終了をまたずに出力することは多い。このような攻撃者の場合、暗号文で見分けがつくようなメッセージ対を生成することができる、という攻撃である。

またDESに対する辞書攻撃、ならびに鍵の全数探索に対する強度向上を目的とした、DESの三重利用モードに対する解析がある。Bihamは[B96]で、多くの多重利用モードが有効な強化策となっていないことを示している。こののち、Wagnerはさらに初期値の制御を使うことにより、Bihamが安全であろうとしたいくつかのモードについても別の懸念があることを指摘した[W98]。

5 秘匿に関する利用モード

この章では、これまで知られている秘匿に関する利用モードのうち、主に工業的に用いられているものや、機能面などで重要視する必要のあるものを詳細に説明する。

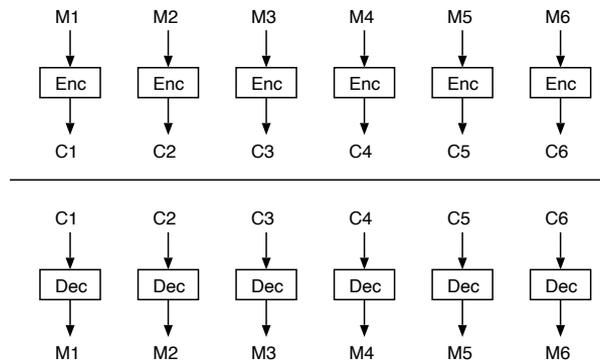


図 9: ECB モードの暗号化と復号のブロック図

5.1 ECB

仕様の概要 ECB(Electronic CodeBook, 電子辞書) モードは, 平文長が n の倍数であるような平文に対して暗号化を行なう利用モードである [FIPS81]. 手法は, 平文を n ビット毎のブロックに分割し (それぞれを M_i とする), それぞれ独立にブロック暗号の暗号化関数の入力とする. その結果得られた出力が暗号文ブロック (C_i) となり, 暗号文はそれらを接続したものである.

$$C_i = Enc_K(M_i).$$

この利用モードには初期値がない. 平文と鍵のみから暗号文が生成される. 復号はその逆関数である.

$$M_i = Dec_K(C_i).$$

安全性 ECB モードには以下のような欠点があるため, その特性が必要でない限り利用すべきではない. 具体的には, 平文がオールゼロなど, ある文字列を繰り返すものを想定すると, 暗号文もあるパターンを繰り返すことになる. 一般化して, 同じ平文パターンは同じ暗号文パターンとして再現されるため, 暗号文からそのような情報が漏洩する.

この欠点を補う方法としては, 平文ブロックが衝突しない (同じ値にならない) ように圧縮を掛けたり, 平文としてエントロピの高いデータを用いることなどが挙げられる. しかしながら, これらの対策も万全ではないため, できる限り他の利用モードを使うべきである.

効率 平文長 $t \times n$ ビットに対して, ブロック暗号を t 回呼び出すのみであり, 処理効率はよい.

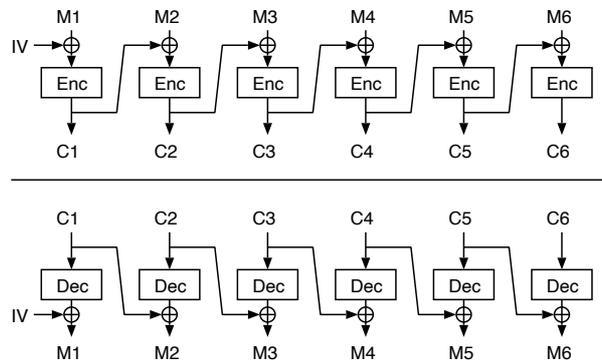


図 10: CBC モードの暗号化と復号のブロック図

エラー伝播 暗号文を伝送するなどした時に発生した 1 ビットのエラーは該当ブロック n ビットに影響を及ぼす可能性がある。

同期ずれについては (同期のずれる幅がブロック単位である特殊な場合を除いて), 別途再同期のメカニズムが必要である。

並列処理性など 暗号化, 復号ともに並列処理性と, 処理順序不問性 (Out-of-order) 性がある。すなわち, ブロック単位でデータが入れ替わったとしても, その順序入れ替えをすることなく, 到着した順序に復号処理に渡すことができる。もちろん, 復号結果は, 適切な順序に並べかえねば正しい平文には戻らない。

復号 復号時にも, 暗号化処理と同様に並列処理性や Out-of-order の利点がある。復号処理には, ブロック暗号の復号関数を使う。

5.2 CBC

仕様の概要 CBC(Cipher Block Chaining, 暗号文ブロック連鎖) モードは, 平文長が n の倍数であるような平文に対して暗号化を行なう利用モードである [FIPS81]。手法は, 平文を n ビット毎のブロックに分割し (それぞれを M_i とする), 中間値 $H_i = M_i \oplus C_{i-1}, C_0 = IV$ を生成したあと, それをブロック暗号の暗号化関数の入力とする。その結果得られた出力が暗号文ブロック (C_i) となり, 暗号文はそれらを接続したものである。

$$C_i = Enc_K(M_i \oplus C_{i-1}).$$

復号はその逆関数である.

$$M_i = Dec_K(C_i) \oplus C_{i-1}.$$

安全性 CBCモードの安全性は [BDJR97] で議論されている. ここでは, 以下の条件がすべて満たされる場合において秘匿の意味で証明可能安全性を持つ.

1. 攻撃者は適応的選択平文攻撃のみである.
2. 初期値生成が以下に限定されるものである.
 - (a) 攻撃者が事前に知ることができない乱数
 - (b) 信頼できる nonce を一度ブロック暗号 (鍵は暗号化鍵でよい) で攪拌したもの
3. 内部で用いるブロック暗号が, 擬似ランダム置換モデル以上の安全性をもつ.
4. 攻撃者の選択平文に対する暗号文の獲得は, メッセージ単位である.

より, 具体的には, Left-or-Right 不可識別性 (秘匿に関する定義のひとつ) の観点からいくつかの安全性が [BDJR97] で示されており:

1. 内部のブロック暗号をランダム関数モデルに置き換えた場合の, CBCモードの安全性が与えられている. Left-or-Right におけるアドバンテージの定義は参考文献を参照頂くとして, そのアドバンテージが以下の式で評価できる.

$$\text{Adv}_{CBC-\rho}^r \leq (\mu^2/n^2 - \mu/n) \cdot 2^{-n}.$$

ここで, 攻撃者の能力として最大 q 回の選択平文質問を行ない, その平文長が合計 μ ビットとする.

2. 内部のブロック暗号を擬似ランダム関数モデルに置き換えた場合の, CBCモードの安全性が与えられている. 具体的には, 擬似ランダム関数のパラメータを $(t', q'; \epsilon')$ とすると, 任意の q に対して, これを使った CBCモードについての安全性が $(t, q, \mu; \epsilon)$ -安全であることをいうための定数 c が存在する. ここで

$$(t, \mu, \epsilon) = (t' - c\mu, q'n, 2\epsilon' + (\mu^2/n^2 - \mu/n) \cdot 2^{-n}).$$

効率 平文長 $t \times n$ ビットに対して、ブロック暗号を t 回呼び出すのみであり、処理効率はよい。

エラー伝播 暗号文を伝送するなどした時に発生した1ビットのエラーは該当ブロック n ビットに影響を及ぼす可能性があり、次のブロックの該当部分1ビットが確実に反転する。

同期ずれについては、ECBと同様、特殊な場合を除いてそれ自身で回復しないため、別途再同期のためのメカニズムが必要である。

並列処理性など 暗号化には、まったくの並列処理性がない。一方、復号では、ブロック暗号処理に関する並列処理は可能である。しかしながら、平文データを復元するためには前ブロックの暗号文ブロックが必要であることを注意しなければならない。

また、ECBモードほど小さな単位では実現できないが、ある程度ブロックがまとまれば、Out-of-order的な復号処理も可能な場合がある。すなわち、 t ブロック単位でデータが入れ替わったとしても、その順序いれかえをすることなく、到着した順序に復号処理に渡すことができ、その場合、最初のブロックを除いた $t-1$ ブロックは正常に復号可能である。

ただし、例外的に並列処理が可能な運用もある。ANSI X3.106 や ISO 10116 では、CBCモードをインターリーブすることにより、ある程度の並列度を持たせることができる暗号方式を記載している。具体的には、独立なCBCモードを並列度数だけ飛ばしながらメッセージストリームを処理する仕様である。この場合、初期値も並列度数だけ用意せねばならず、それぞれ独立かつランダムに選択する必要がある。

復号 復号時に関する特別な注意事項はない。復号処理には、ブロック暗号の復号関数を使う。

CTS CTS(CipherText Stealing, 暗号文窃盗)モードは、RFC2040[RFC2040]で提案された、CBCモード向けの端数処理モードである。RFCではバイト単位の端数処理のみが定義されているが、単純に一般化することで n ビット以上の任意のビット数のメッセージに対して処理可能となる。

このモードはほとんどの処理がCBCモードであるので、安全性以外の主な特徴はCBCモードに準じる。

安全性については特別に議論された技術文書は見当たらないが、次のように考えることで秘匿に関する安全性は保持できていると考える。

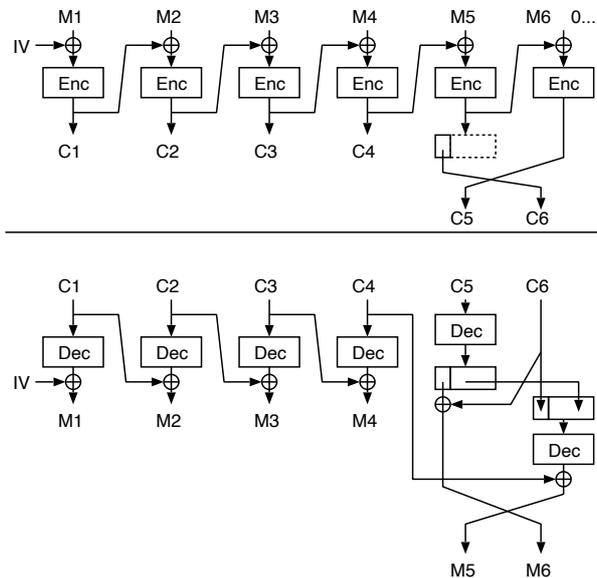


図 11: CTS モードの暗号化と復号のブロック図

CBC⁺という新しい利用モードを考える。従来の CBC モードを暗号化する場合に、 n ビットの 0 パディングを行なってから CBC モードを処理する。CBC⁺と同様に安全であると考えられる。

ここで $m \times n$ ビット長の CBC⁺ モードでの暗号化結果と $(m - 1) \times n + t$ ($1 \leq t < n$) ビット長の CTS モードでの暗号化の強度は、後者、すなわち CTS のほうが強力である。なぜならば、CTS における任意の攻撃者の振舞いは、すべて前者に対する攻撃者として再現できるからである。よって CTS は CBC と同程度に強力であると考えられる。

5.3 k -CFB

仕様の概要 CFB(Cipher FeedBack, 暗号文フィードバック) モードは、パラメータ k を持つブロック暗号利用モードである [FIPS81]。平文長が k の倍数であるような平文に対して暗号化を行なう利用モードであることから、バイト単位のデータなど、データ単位長がブロック長の倍数でないような場合に用いられていた。便宜的に内部レジスタ R を考えながら処理を説明する。

k ビットの倍数長のメッセージ M は、 k ビット毎のブロックに分割しする (それぞれを M_i とする)。初期値 IV は R の初期値 R_0 である。各ブ

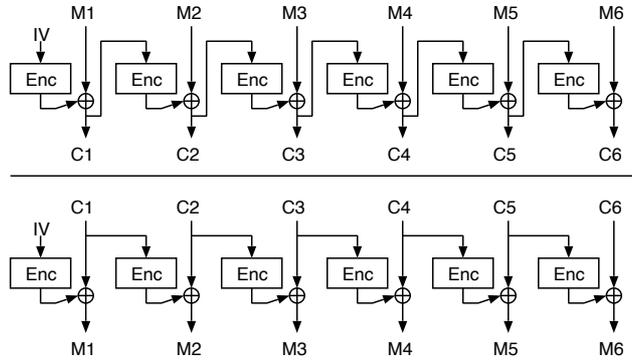


図 12: n -CFB モードの暗号化と復号のブロック図

ブロックでのブロック処理は、まず中間値 $H_i = Enc_K(R_i)$ を生成することから始まり、このうち上位 k ビットの値 \hat{H}_i を、平文ブロック M_i と排他的論理和をとることで暗号文ブロック $C_i = M_i \oplus \hat{H}_i$ を得る。最後に R を更新する。 R を上位へ k ビットシフトし、シフトの際、0 が埋められた下位 k ビットに C_i を埋め込む。よって、 $k = n$ の場合は、 $R_i = C_i$ となる。

$$C_i = M_i \oplus \text{msb}_k(Enc_K(R_{i-1})),$$

$$R_i = ((R_{i-1}) \ll k) \oplus C_i.$$

復号はその逆関数である。

$$M_i = C_i \oplus \text{msb}_k(Enc_K(R_{i-1})),$$

$$R_i = ((R_{i-1}) \ll k) \oplus C_i.$$

安全性 CFB モードの安全性については、[AGPS02] で評価している。ここではレジスタ値の衝突確率をもとに Left-or-Right における攻撃者のアドバンテージの上限を求めている。 $k < n$ の場合には、異なる時間のあいだでレジスタの値同士が独立でない。これも考慮した上での評価である。ランダム関数を使った場合には、攻撃者のアドバンテージは

$$\epsilon_{\text{CFB-}\rho}^{\text{lr}} \leq q(q-1)2^{-l-1},$$

となる。ここで t は攻撃者の計算時間、 q は攻撃者の質問回数、 l はランダム関数の入力長、 L はランダム関数の出力長である。

さらに、擬似ランダム関数を使った k -CFB モードについての安全性の評価結果も調べられており、 l ビット入力- L ビット出力の $(t', q'; \epsilon')$ -安全

な擬似ランダム置換を使った場合、CFBモードは $(t, q, \mu; \epsilon)$ -安全である。ここで

$$(t, q, \mu, \epsilon) = (t' - q \times t_{\text{CFB}} - t_{\text{const}}, q', q'L, 2\epsilon' + q(q-1)2^{-l-1}),$$

であり、 t_{CFB} はランダム関数の呼び出しを除いたに CFB モード 1 ブロック処理に必要な処理時間である。

ただし、特に k が小さい場合には、初期値に注意する必要がある。例えば、0ばかり続く平文（もしくは1ばかり続く平文）を初期値 $IV = 0^n$ や $IV = 1^n$ の 1-CFB で暗号化した場合、約半分の鍵に対しては内部レジスタの更新がまったくおこなわれなため安全性に問題が生じることとなる [W02b]。

効率 CFBモードは、パラメータの値に応じて処理効率に変化し、場合によっては、他のモードよりも極端に非効率的となる。

具体的には mk ビットのメッセージを暗号化するためには m 回のブロック暗号の呼び出しを必要とする。 $k = n$ の場合、ECB や CBC と同じ程度の効率であるが、それ以外の場合、約 n/k 倍の処理量となる。

エラー伝播 1 ビットの暗号文におけるエラーにより、まず該当の平文ビットの反転が起こる。さらに該当エラーがレジスタに残る限り、平文回復ができないので、その間はエラーがおき続ける可能性がある。これは最悪、 $[n/k]$ ブロック分、エラーが起こる可能性がある。

並列処理性など CBCモードと同様、暗号化には並列処理性がない。復号では、該当ブロックのブロック暗号処理結果は次のブロック暗号処理に直接影響しない。よって構成上はパイプラインングなど並列処理性はある。しかし、該当ブロックを処理するためには、該当ブロック以前の暗号文ブロックが必要であるので、各々のブロック暗号エンジンでこれらをバッファリングするメカニズムが必要である。これらバッファは左右にずれているだけであるので、(並列度に応じた長い) バッファを共有することでも実現可能である。

また CBC モードと同様に、例外的に並列処理が可能な運用もある。ANSI X3.106 や ISO 10116 では、CFB をインターリーブすることにより、ある程度の並列度を持たせることができる暗号方式を記載している。具体的には、独立な CFB モードを並列度数だけ飛ばしながらメッセージストリームを処理する仕様である。この場合、初期値も並列度数だけ用

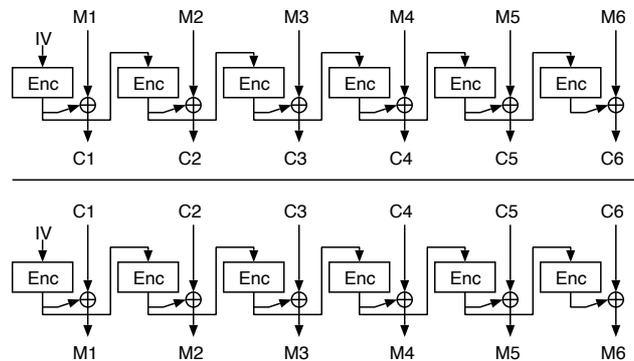


図 13: OFB モードの暗号化と復号のブロック図

意せねばならず、それぞれが安全であるためには、ランダムかあるいは攻撃者に選択されないような nonce にする必要がある。

復号 CFB モードでは、ブロック暗号の復号関数を利用しない。よって、CFB 暗号化、CFB 復号の両方の機能を実装する場合には、その実装コストは、CBC や ECB に比較して軽いことが期待できる。

自己同期性 CFB の大きな特徴として、自己同期性がある。これはブロック単位でのデータの欠損や挿入については、ある程度のエラーブロックをひきずりながらも、その後には、復号処理が回復するものである。

この機能は CBC にも一応あてはめることはできる (ただし同期パケットの境界がブロック暗号のブロック長) が、比較的大きいためこの機能が現実的に便利であることはあまりない。

CFB の場合、ブロック長を任意に設定することができるため、例えばバイト単位や、極端な例ではビット単位の同期ずれやデータ欠損挿入などにも回復する強みがある。ただし、注意を要するのは、ビット単位やバイト単位など短いデータ境界での自己同期を期待すればするほど、その処理負荷が大きくなる。

これを解決したのが、OCFB モードである。詳細は OCFB モード参照。

5.4 OFB

仕様の概要 OFB(Output FeedBack, 出力フィードバック) モードは、初期値のみに依存し逐次的に擬似乱数を生成しながら暗号化を行なう方法

であり、任意のビット長の平文を処理できる [FIPS81]. まず、平文を n ビット毎のブロックに分割 (それぞれを M_i とする) し、最後の端数の部分は端数ブロックとして扱う. 初期値 IV を内部レジスタの初期値 H_0 とする. H_{i-1} をブロック暗号入力とし、暗号化処理の結果を H_i とする (すなわち次のブロックの内部レジスタの値にもなる). これより暗号文ブロック $C_i = M_i \oplus H_i$ を生成する.

$$\begin{aligned} H_i &= Enc_K(H_{i-1}), \\ C_i &= M_i \oplus H_i. \end{aligned}$$

この利用モードには初期値がない. 平文と鍵のみから暗号文が生成される. 復号はその逆関数である.

$$\begin{aligned} H_i &= Enc_K(H_{i-1}), \\ M_i &= C_i \oplus H_i. \end{aligned}$$

安全性 OFB モードに関するきちんとした安全性の証明は知られていない. しかし、ブロック暗号出力全体をそのまま入力に戻すことで内部のブロック暗号が理想的である場合、周期が約 2^{n-1} になることが知られている. この周期の中では乱数性の高い鍵ストリームとして利用できるため、高い安全性が期待できる.

効率 OFB は ECB や CBC と同等の処理効率で暗号化、復号処理を行なうことができる.

エラー伝播 暗号文における 1 ビットのエラーは、対応する平文ビットの反転を起こす. しかし、それ以降のエラー伝播などの影響はない.

ただし、同期ずれについては (ECB と同様、ブロック長単位の同期ずれでない限り) 耐性がなく、同期ずれが起こるような場合には、別途再同期のメカニズムが必要である.

並列処理性など 暗号化、復号処理ともに、並列処理性はまったくない.

しかし、インターリービングによる並列処理が可能な運用方法が知られ、ANSI X3.106 や ISO 10116 など記載されている. 具体的には、独立な OFB モードを並列度数だけ飛ばしながらメッセージストリームを処理するものである. この場合、初期値も並列度数だけ用意せねばならず、それぞれが安全であるためには、ランダムかあるいは攻撃者に選択されないような nonce にする必要がある.

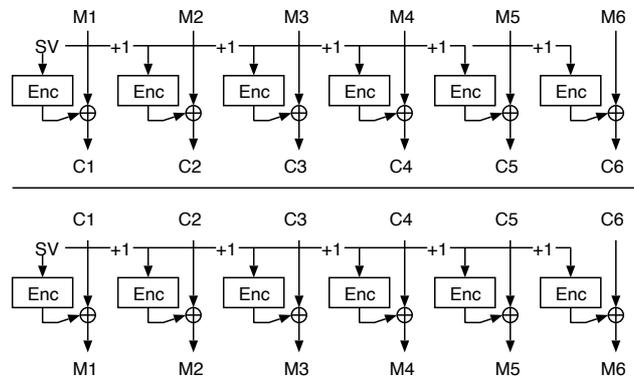


図 14: CTR モードの暗号化と復号のブロック図

復号 OFB モードでは、ブロック暗号の復号関数を利用しない。よって、OFB 暗号化、OFB 復号の両方の機能を実装する場合には、その実装コストは、CBC や ECB に比較して軽いことが期待できる。

***k*-OFB** FIPS-81 など古い利用モードの標準化では、OFB モードを *k* ブロック単位で行なうことも指摘されていた。これは、*k*-CFB と同様、*n* ビット単位でないデータを扱う場合への適用を考えたものであった。しかし、*k* < *n* の場合、安全性の観点から大きな問題があることを理由に、OFB は *k* = *n* として用いるべきとなった。よってこの古い仕様は今後使われるべきでない。

安全性の懸念には二通りある。第一には、脆弱な初期値の存在である。初期値に *n* ビットの 0 を与えて 1-OFB を実行すると約半数の鍵において 0 ばかり続く鍵系列が出力され、初期値に *n* ビット値 1 を与えて 1-OFB を実行すると約半数の鍵において 1 ばかり続く鍵系列が出力される [W02b]。また、第二の安全性の懸念として、*k* < *n* の場合には、内部レジスタの更新関数が単射性を保証できなくなり、これが理由で周期の平均が $2^{n/2}$ ブロック程度となることが挙げられる。

5.5 CTR

仕様の概要 CTR(カウンタ)モードは、初期値のみに依存し逐次的に擬似乱数を生成しながら暗号化を行なう方法であり、任意のビット長の平文を処理できる [DH79, LRW00]。まず、平文を *n* ビット毎のブロックに分割(それぞれを M_i とする)し、最後の端数の部分は端数ブロックとし

て扱う。開始値 SV を内部レジスタの初期値 R_1 とする。 R_i をブロック暗号入力とし、暗号化処理の結果を H_i とする。これより暗号文ブロック $C_i = M_i \oplus H_i$ を生成する。次のブロックでは、内部レジスタ R を整数カウンタとして1数えあげる。

$$\begin{aligned} C_i &= M_i \oplus Enc_K(R_i), \\ R_{i+1} &= R_i + 1. \end{aligned}$$

復号はその逆関数である。

$$\begin{aligned} M_i &= C_i \oplus Enc_K(R_i), \\ R_{i+1} &= R_i + 1. \end{aligned}$$

ここで開始値とは、特殊な運用が必要な初期値である。CTR が安全な処理モードであるために、同一の鍵が用いられている間は常に異なるブロック暗号入力を与える必要がある。

CTR モードでは、内部状態の更新がカウンタであるため、システム要件から、カウンタの更新回数の限度などを知ることができる場合がある。このような情報を使いながら、うまく開始値を定義して、(同じ鍵のもとで) 複数の平文を安全に暗号化できるようにする。

具体的には、ひとつのメッセージ長が32ブロック未満で定義されるシステムでは、下位5ビットをカウンタ動作部分としてリザーブしておき、残り上位 $n-5$ ビットをメッセージIDとして固有な数字を埋め込む。こうすることにより、最大 2^{n-5} 個のメッセージを安全に処理できる¹。

安全性 CTR モードの安全性については [BDJR97] で議論されている。該当の文献では (モードの名称はCTRでなくXORであるが)、開始値が乱数の場合と、カウンタの場合との二種類について検討している。

前者、開始値が乱数の場合、ランダム関数を使ったスキームの安全性について、攻撃者のアドバンテージは

$$Adv_E^t \leq \mu(q-1)/(L \cdot 2^l),$$

となる。ここで t は攻撃者の計算時間、 q は攻撃者の質問回数、 l はランダム関数の入力長、 L はランダム関数の出力長である。

さらに、擬似ランダム関数を使った、開始値が乱数のCTRモードについての安全性の評価結果も調べられており、 l ビット入力- L ビット出力の

¹厳密には 2^{n-5} 個も暗号化してしまうと、別の情報が漏洩するため安全とはいえない。

$(t', q'; \epsilon')$ -安全な擬似ランダム関数を使った場合、乱数開始値のCTRモードは $(t, q, \mu; \epsilon)$ -安全である。ここで

$$(t, \mu, \epsilon) = (t' - c \cdot \frac{\mu}{L}(l + L), q'L, 2\epsilon' + \mu(q - 1)/(L \cdot 2^l)),$$

である。

また、カウンタを初期値にしたCTRモードをランダム関数モデルと一っしょに用いた場合、 $\text{Adv}_E^{\text{ctr}} = 0$ となる。ここで攻撃者のパラメータとして、計算時間が最大 t 、質問回数が最大 q 、質問長が最大 $\mu < L2^l$ の場合を考える。

そして、擬似ランダム関数を使った、開始値がカウンタのCTRモードについては、 l ビット入力- L ビット出力の $(t', q'; \epsilon')$ -安全な擬似ランダム関数を使った場合、 $(t, q, \mu; \epsilon)$ -安全である。ここで

$$(t, \mu, \epsilon) = (t' - c \cdot \frac{\mu}{L}(l + L), \min(q'L, L2^l), 2\epsilon'),$$

である。

効率 CTRはECBやCBCモードとほぼ同程度に効率的である。

エラー伝播 暗号文における1ビットのエラーは、対応する平文ビットの反転を起こす。しかし、それ以降のエラー伝播などの影響はない。

ただし、同期ずれについては(ECBと同様、ブロック長単位の同期ずれでない限り)耐性がなく、同期ずれが起こるような場合には、別途再同期のメカニズムが必要である。

並列処理性など 暗号化復号ともに並列処理性が実現可能である。しかし、このためには、処理しているブロックが平文(もしくは暗号文)の何ブロック目であるかという情報を処理系が知っている必要がある。従って、パイプラインングなどのようなメカニズムで、メッセージ(もしくは暗号文)を最初のブロックから処理する場合には問題とはならない。

同様に、何ブロック目のデータであるかがわかれば、処理順序不問性(Out-of-order)性も達成できる。すなわち、ブロック単位でデータが入れ替わったとしても、その順序いれかえをすることなく、到着した順序に復号処理に渡すことができる。もちろん、復号結果は、到着順序に応じて並べかえねば正しい平文には戻らない。

復号 CTRモードでは、ブロック暗号の復号関数を利用しない。よって、CTR暗号化、CTR復号の両方の機能を実装する場合には、その実装コストは、CBCやECBに比較して軽いことが期待できる。

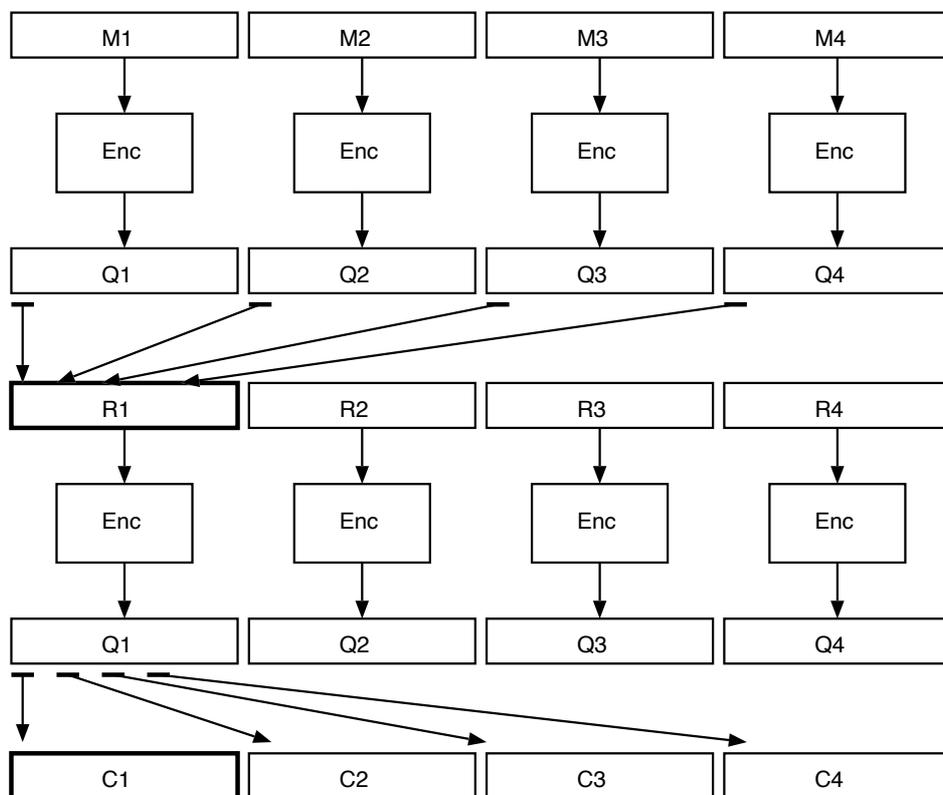


図 15: 2DEM モードの暗号化と復号のブロック図

5.6 2DEM

仕様の概要 この2DEM (2D Encryption Mode, 二次元暗号化モード) の目的は、ECBの安全性の懸念、CBCの並列処理性の低さを克服することを目的に、主にバイトデータを二次元配列で解釈し、暗号化処理を行なうことを記述したものである [BA01].

具体的には、メッセージをまずECBで処理したものを、バイト単位でインターリーブする。そうしてできたブロック列を再度ECBで処理し、その結果を再度インターリーブして暗号文ブロック列とするものである。

5.7 ABC

仕様の概要 ABC (Accumulated Block Chaining, 累積ブロック連鎖) は、エラー伝播が最後まで続くような暗号利用モードとしてAES利用モードに提案された [K00]. しかし、提案は秘匿の目的のみであり、上記性質が

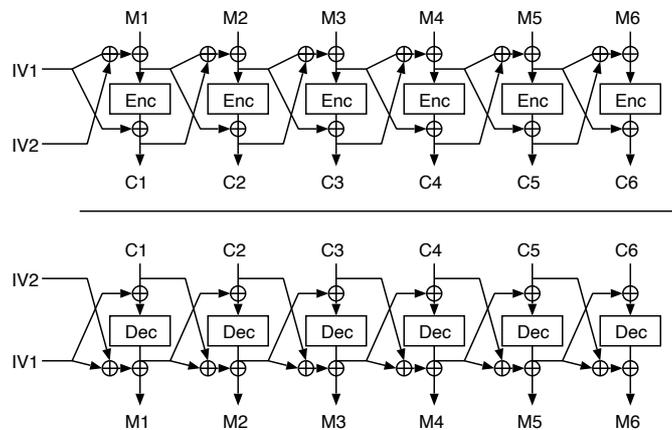


図 16: ABC モードの暗号化と復号のブロック図

暗号学的な意味のある安全性には特に関連していない。処理の流れを図 16 に示しておくが詳細な説明は省略する。

5.8 IGE

IGE (Infinite Garble Extension, 無限改竄拡張) は、もともと CBC と同じくらい古くに提案された利用モードである [C78]。AES の利用モードで、このモードに対する解析結果が発表され、メッセージ認証に対して安全でないことが示されている [GD00]。

暗号化と復号の処理フローが同じである (つまり両方がそれぞれ上下対称) であるのは、なんらかの実装の利点があるかもしれないが、それ以上に、復号処理でブロック暗号の復号関数が必要となり、そうでない CFB, OFB, CTR などがより効率的である可能性が高い。ここでは詳細な記述は行なわない。

5.9 自己同期型利用モード

CFB モードは FIPS81 に掲載され、長い間使われ続けてきた。しかし、CFB モードが特徴とする自己同期性には一つの懸念があった。なるべく小さいデータ単位、例えば 1 ビットや 1 バイト単位などでの自己同期を行なうためには、それだけの処理負荷の増大を伴うことである。例えば、AES で 1-CFB を実行してしまうと、CBC モードの 128 倍もの処理が必

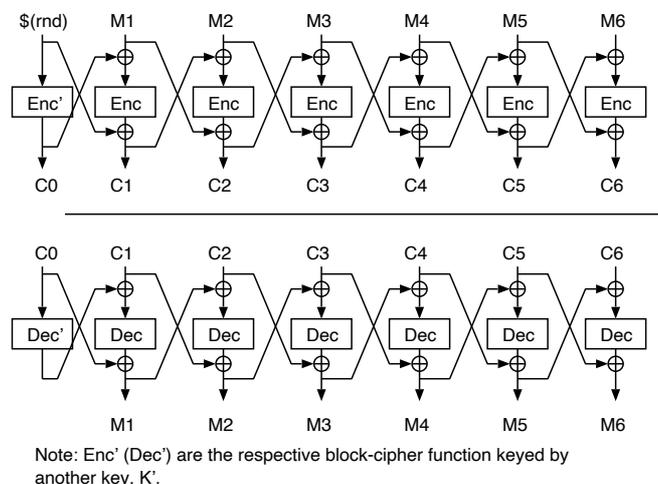


図 17: IGE モードの暗号化と復号のブロック図

要となる。しかしビット単位の自己同期が可能な唯一の標準利用モードであった。

しかし、効率の良い自己同期に関する一連の研究成果があり、標準化されるに至っていないものの、技術的に重要なものであるのでここに紹介する。

Maurer は自己同期に関する新しいアプローチとしてその設計手法と解析結果を発表した [M91]。この発表から遅れて、Jung, Ruland は [JR99] にて類似の手法を提案している。さらに、Alkassar, Gerald, Pfitzmann, Sadeghi も同様な手法を提案している [AGPS02]。提案手法の主に通ずる部分では、目的として任意の自己同期機能を実現しながらもその処理速度、厳密にはブロック暗号の呼び出し回数はなるべく ECB に近付けるものである。

具体的には、 k -CFB を改良する方向で理解するとわかりやすい。ブロック暗号出力をこれまで捨てていたところをバッファとして動かせることにより効率化を行なっている。バッファが空になれば再度ブロック暗号処理を行ない新しい乱数列を充填する。

さらに同期回復のためのアイデアとして、暗号文パターンを監視し、特定のパターンが出現したところで、先述のバッファの残量を無視して、ブロック暗号を処理させ、バッファをフラッシュする。

これら二点のアイデアを使うことにより、同期が暗号文パターンで行なわれるため自己同期が実現でき、かつ、パターンサイズを適切に選ぶことでバッファから捨てる乱数長を減らすことができる。

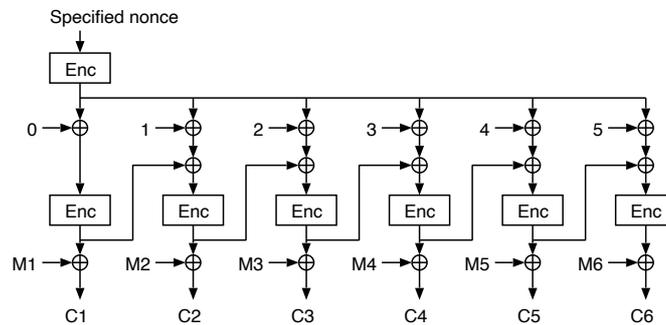


図 18: $f8$ モードの暗号化処理のブロック図

安全性に関する考察は [M91] でも行なわれているが、現在議論される証明可能安全性の観点からは、[AGPS02] に記載されている安全性の証明が参考になる。安全性に関する欠陥は今のところ見つかっていない。

また、処理速度の観点からの解析はさまざまな論文でなされており [H01a, H01b, H03a, JKRW01, AGPS02]、結果として、場合により CFB よりも非常に効率的な処理となっている。

5.10 $f8$ (3GPP)

3GPP では、ブロック暗号 KASUMI の利用モードとして二つの利用モード $f8$ と $f9$ を定義している [3GPPa]。それぞれ、秘匿、メッセージ認証に関する利用モードである。KASUMI の設計も含め、この標準化は、3GPP での利用を目的としており、モバイル端末と基地局間の無線区間の暗号化に特化している。従って、汎用目的にはあるべき性質などが棄却され、必要な目的に特化した方式であることを注意しておく。

$f8$ は、仕様で定義された “nonce” 入力と鍵から鍵ストリームを生成する方法である。暗号化はこの鍵ストリームと入力ストリームとの排他的論理和をとることで行われる。鍵ストリームの生成は、カウンタモードに CBC モードを組み合わせたようなものである。具体的には、鍵ストリームブロックを生成するために、ブロック暗号入力に “nonce” 値、カウンタ値、そして前ブロックの鍵ストリーム値全部の排他的論理和をとったものである。

これについては安全性の問題点はないように見える。また、処理効率も ECB 程度であり、復号には CFB などと同様、KASUMI の暗号化関数だけで処理が可能である。ただし、並列処理性能がない仕様となってい

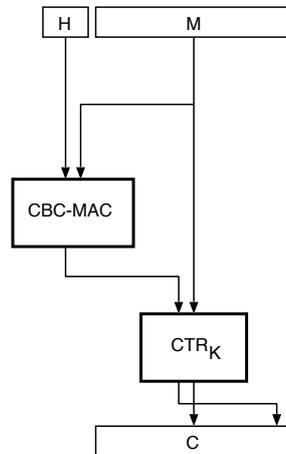


図 19: CCM モードの暗号化と復号のブロック図

る (が, これは必要ない実装だけで用いられるという理由で問題にはならない).

6 認証暗号に関する利用モード

この章では, これまで提案されてきた認証暗号に関する利用モードのうち, 特に安全性の欠陥の見つかっていないものについて簡単に紹介する.

ここで, 「本章にて扱わなかった認証暗号の利用モードはすべて安全性に欠陥がある」という意味ではないことに注意する.

6.1 CCM

仕様の概要 CCM (Counter with CBC-MAC)[WHF02] は CTR モードによる暗号化と CBC-MAC を組み合わせたメッセージ認証つき暗号化方式である. 具体的には, メッセージ, 及び認証データに対して, CBC-MAC による MAC を生成し, MAC 処理で生成されたタグとメッセージを, CTR モードで暗号化する. CTR, CBC-MAC の両方に同じ秘密鍵を用いているので鍵のセットアップは 1 回である.

暗号処理や MAC 生成には, いくつかの詳細なパディングが記載されており, その一部に長さ情報が含まれる.

本方式は 2004 年 5 月に発行された NIST SP800-38C[SP800-38C] に規定されている.

安全性 CTRは秘匿の観点から安全性が証明されてはいるが、攻撃者の能力が、CTRで暗号化してくれる暗号化オラクルとの通信に限定されていることが重要である。CCMでは、同じ暗号化オラクルでも暗号化処理が異なるため、攻撃者に得られる情報が異なる。よって従来のCTRモードの安全性証明はCCMの安全性とは独立と考えるべきである。

CCMについての安全性評価はJonssonにより[J02]で与えられている。ここでは、メッセージ認証、秘匿の二つについての安全性の議論が与えられている。

メッセージ認証に関する安全性評価では、一般的な改竄攻撃の定義を用いており、アドバンテージは攻撃者の改竄成功確率としている。攻撃者の能力として、1. 暗号化オラクルへの暗号化要求(質問長上限 μ_E)、2. 改竄試行(復号オラクルへ暗号文を質問し、その暗号文が有効か無効かの判定、質問長上限 μ_F 、質問回数上限 q_F)の二つが許可されている。

ブロック暗号のブロック長を n 、タグ長を t とすると、ブロック暗号を擬似ランダム関数で置き換えた場合の攻撃者のアドバンテージは

$$\text{Adv}_{\text{CCM}}^{\text{auth}} = \epsilon' + q_F \cdot 2^{-t} + (\mu_E + \mu_F)^2 \cdot 2^{-n-1},$$

となる。ここで、 ϵ' は、擬似ランダム関数に関する安全性のパラメータであるが、質問回数などのその他のパラメータとの相関が示されておらず、不完全な記述である。

また、秘匿に関しては別の定義を用いている。この定義はReal-or-Randomに類似するが、乱数の暗号文、ではなく、乱数そのものをオラクルは攻撃者に返す。あとは、アドバンテージの定義なども含めて、Real-or-Randomと同じである。攻撃者は暗号化オラクルのみが利用可能であり、この場合擬似ランダム関数を使ったスキームに対する攻撃者のアドバンテージは

$$\text{Adv}_{\text{CCM}}^{\text{rr}'} = \epsilon' + (\mu_E)^2 \cdot 2^{-n-1},$$

となる。

効率 CCMはECBやCBCモードの2倍のブロック暗号呼び出しを行なうため、処理量もECBのそれに比べて約2倍である。

ただし、実質的にCTRモードとCBC-MACの組合せであり、データサイズが(処理系が扱えるメモリサイズに対して)大きい場合には注意が必要である。例えば、ストリーミングデータなどへの処理には、CCMとして、内部で呼び出すCTRの処理とCBC-MACの処理、両方を交互に行なうような実装を行なわないと、処理が不可能となる。この場合、中間データの保持のためにいくらかの必要レジスタサイズの増加が考えられる。

その他の懸念とされる事項が技術文書として公開されている。後の議論の章を参照頂きたい。

並列処理性など まず、CCM 処理中の CTR 処理と、CBC-MAC 処理は並列処理が可能である。従って適切な実装により 2 並列度までは簡単に達成できる。しかし、CBC-MAC には並列処理機能がないため、それ以上の並列処理は CTR のみに適用可能となる。

これは復号処理についても同じことがいえる。

復号 CCM モードでは、ブロック暗号の復号関数を利用しない。よって、CCM 暗号化、CCM 復号の両方の機能を実装する場合には、その実装コストは、CBC や ECB に比較して軽いことが期待できる。

議論 CCM は IEEE 802.11 の標準ドラフトなど、いくつかの業界標準方式として採用されている実績がある [WHR02] このモードの利用に関する注意を記した文書が Rogaway, Wagner らにより公開されている [RW03]。主に効率に関するコメントと安全性に関するコメントであるが、安全性は上記 [J02] の結果を否定するものではなく、CCM の NIST への提案文書 [WHF02] における安全性の主張に根拠がなく、かつ誤りと思われる宣言がいくつかある、という指摘に留まっている。

[RW03] で指摘する効率に関する注意点は次の 3 点である。

1. オンラインアルゴリズムでない。
2. ワード境界がずれる可能性がある。
3. 固定ヘッダ情報に対しての事前計算ができない。

その他、仕様が複雑であることや、タグ長 (改竄検知に関する安全性レベル) の柔軟性から考えうる安全性への懸念などが示されている。

これらを指摘した [RW03] では、CCM の代替として、EAX の利用を提案している。

6.2 CWC

仕様の概要 CWC (Carter Wegman with Counter) モードは、CTR モードの暗号化と、Universal hash (汎用ハッシュ) による MAC 生成とを利用したメッセージ認証つき暗号化方式である [KVV03]。

具体的には、メッセージに対してCTRモードで一度暗号文を生成し、その暗号文に対して(暗号化されない付加データの入力を許して)MACをつけるというものである。

MAC生成は、universal hashという性質をもつ特殊な(パラメータ付きの)ハッシュ関数を使って暗号文のハッシュ値を生成し、さらにこれを使い捨て的な乱数(ただし、真の乱数ではなく、仕様で定義された計算方法で求められる、攻撃者には計算できない値)でマスクして暗号文に添付するものである。

本方式はNISTの策定しているAES利用モードへ提案された利用モードである[KVW03].

安全性 提案の文書では、128ビットブロック暗号に限定した安全性評価を行なっている。

メッセージ認証については、内部で用いるブロック暗号を擬似ランダム関数に置き換えた時(それに対する攻撃者のアドバンテージを ϵ' と定義した時)、MAC偽造を目的とした攻撃者のアドバンテージは以下のようになる。

$$\text{Adv}_{\text{CWC}}^{\text{auth}} \leq \epsilon' + (\mu_M + \mu_A)/2^{133} + 2^{-125} + 2^{-t}.$$

ここで、 μ_M, μ_A はそれぞれ、メッセージ、付加情報の長さの上限であり、 t はタグ長に相当するアルゴリズムの安全性のパラメータの一つである。

また、内部で用いるブロック暗号を擬似ランダム置換とした場合には、質問回数が最大 $q-1$ 、オラクルへの質問長が最大 μ である時、改竄を行なう攻撃者のアドバンテージは以下のようになる。

$$\text{Adv}_{\text{CWC}}^{\text{auth}} \leq \epsilon' + (\mu/128 + 3q + 1)^2/2^{129} + (\mu_M + \mu_A)/2^{133} + 2^{-125} + 2^{-t}.$$

秘匿については、暗号文が乱数との識別できる/できないという定義で議論している。具体的な評価では、内部で用いるブロック暗号を擬似ランダム関数に置き換えた時(それに対する攻撃者のアドバンテージを ϵ' と定義した時)、暗号文を乱数と区別する攻撃者のアドバンテージは以下のようになる。

$$\text{Adv}_{\text{CWC}}^{\text{priv}} \leq \epsilon'.$$

また、内部で用いるブロック暗号を擬似ランダム置換とした場合には、質問回数が最大 $q-1$ 、オラクルへの質問長が最大 μ である時、改竄を行なう攻撃者のアドバンテージは以下のようになる。

$$\text{Adv}_{\text{CWC}}^{\text{priv}} \leq \epsilon' + (\mu/128 + 3q + 1)^2/2^{129}.$$

効率 この利用モードは効率の評価がやや困難である。処理はCTRモードの処理と universal hash の計算の部分が大部であるが、後者がブロック暗号による処理でないもののそれ相応の処理となるため、処理するプラットフォームや開発に用いる記述言語などにより universal hash の計算の効率が大きく変化すると考えられる。

少なくともこれまでの利用モードには珍しい(秘密情報に依存した)算術乗算演算があるため、実装には注意が必要な場合がある。

並列処理性など universal hash の処理はCTRモードの結果を用いるため、安直に実装してしまうとこれらの並列処理性がないような実装に陥る可能性がある。しかしながら、CTRモードの処理の最後のデータがMACの最初の処理に用いられるものではないので、仕様書から技術を十分読みとれば、CTRと universal hash との両方の処理を交互に処理するような実装が可能である。

CTR自身は並列処理可能である一方、universal hash の並列処理には、冪乗計算を並列に行なうための工夫が必要である。そのための概要は示してあるが、一般のエンジニアがこれらの文面から並列処理を実現するには別の技術解説文書が必要である。

復号 CWCモードでは、ブロック暗号の復号関数を利用しない。よって、CWC暗号化、CWC復号の両方の機能を実装する場合には(もちろん、冪乗演算のコストが新たに必要だが)、CBCやECBに比較して軽いことが期待できる。

6.3 EAX

仕様の概要 EAX (A Conventional Authenticated-Encryption Mode) は、CTRモードと OMAC [IK03a] を組み合わせた利用モードである [BRW03]。機能としては、入力としてメッセージ、nonce、ヘッダ情報があり、暗号化することにより、メッセージの情報が秘匿されることが保証され、かつメッセージとヘッダ情報の認証が復号時に行なわれる。

具体的な処理としては、以下のような処理となる。メッセージは、nonceから生成された攪拌 nonce N を開始値としてCTRモードにより暗号化する。この結果を暗号文とする。そして N 、暗号文のMAC、ヘッダ情報のMACとの排他的論理和をとり、その結果をタグとするものである。

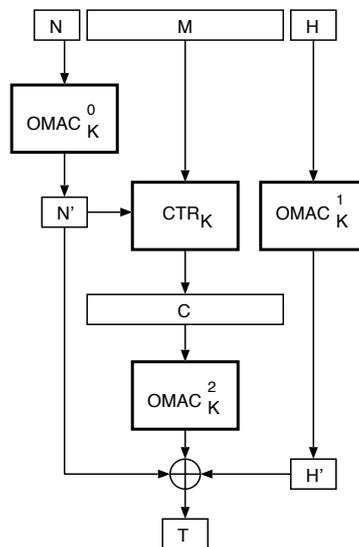


図 20: EAX モードの暗号化の処理を示すブロック図

安全性 証明可能安全性をもつとされているが、その証明についてはまだ公開されていない。近い将来公開される予定である。

効率 処理効率はヘッダ部分の処理と、秘匿するメッセージ部分の処理との重さが異なる。秘匿するメッセージ部分に対しては ECB の二倍必要であるが、ヘッダについては ECB と同等の処理速度である。

並列処理性など 暗号化や MAC 生成など、処理の本質となる部分が3つあるため、並列処理できる/できないという表現では説明次第ではあいまいになる。ここを整理しながら説明する。

ヘッダ部分への処理は他とはほぼ独立であり、ここは切り離して並列度に数えることができる。

メッセージについては、CTR と OMAC が直列に並んでいるため、それ自身では並列処理はできないように見える。しかしながら、CTR が処理した結果である暗号文が生成されれば OMAC 処理は開始できるので多少の遅れをもって並列処理可能である。

また、メッセージ長が長い場合には、CTR と OMAC を同時に動かす必要があるため、そのための実装には注意と工夫が必要である。

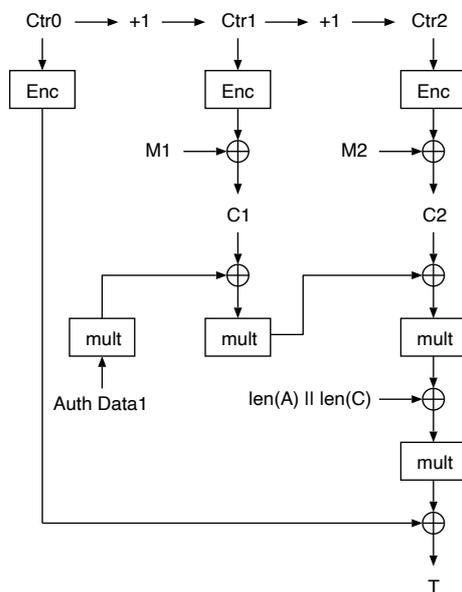


図 21: GCM モードの暗号化処理を示すブロック図

復号 EAX モードでは、ブロック暗号の復号関数を利用しない。よって、EAX 暗号化、EAX 復号の両方の機能を実装する場合には、その実装コストは、CBC や ECB に比較して軽いことが期待できる。

6.4 GCM

仕様の概要 GCM (Galois/Counter Mode) モードは、CTR モードの暗号化と universal hash 関数による MAC 生成を組み合わせたメッセージ認証つき暗号化方式である [MV05]。

具体的には、メッセージに対して CTR モードで暗号文を生成し、その暗号文と、暗号化はされない認証対象のメッセージに関して、 $GF(2^{128})$ 上で定義された universal hash 関数により認証タグを計算し、暗号文に付加する。

安全性 文献 [MV04] にて安全性評価が行なわれている。

メッセージ認証については、内部で用いるブロック暗号を擬似ランダム関数に置き換えた時(それに対する攻撃者のアドバンテージを ϵ' と定義した時)、ブロック長を w 、質問回数が最大 q 、オラクルへの質問長が最大

l_P , それぞれの質問について $\text{len}(C) + \text{len}(A) \leq l$, $\text{len}(IV) \leq l_{IV}$ である時, MAC 偽造を目的とした攻撃者のアドバンテージは以下ようになる.

$$\text{Adv}_{\text{GCM}}^{\text{auth}} \leq \epsilon' + (l_P/w + 2q)2^{-w-1} + q((l_P/w + 2q + 1)[l_{IV}/w + 1]2^{1-w} + [l/w + 1]2^{-t})$$

秘匿については, 暗号文が乱数との識別できる/できないという定義で議論している. 具体的な評価では, 内部で用いるブロック暗号を擬似ランダム関数に置き換えた時, 攻撃者がブロック暗号と擬似ランダム関数を識別できるアドバンテージを ϵ' と定義した時, 暗号文を乱数と区別する攻撃者のアドバンテージは以下ようになる.

$$\text{Adv}_{\text{GCM}}^{\text{priv}} \leq \epsilon' + (l_P/w + 2q)2^{-w-1} + q((l_P/w + 2q)[l_{IV}/w + 1]2^{1-w} + [l/w + 1]2^{-t})$$

効率 文献 [MV04] における GCM のソフトウェア実装では, $\text{GF}(2^{128})$ 上の乗算を 256B, 4KB, 64KB のテーブル参照を用いて実装しており, テーブルサイズが大きいくほど高速である. また, ハードウェア実装においても $\text{GF}(2^{128})$ 上の乗算は CWC で利用されている整数乗算に比べて極めて小型かつ高速に実装することができ, GCM はソフトウェア実装, ハードウェア実装ともに CWC に対し優位性をもっている.

並列処理性など universal hash 関数の積和演算は基本的に逐次処理であるが, $\text{GF}(2^{128})$ 上の乗算は AES の暗号化処理に対して非常に軽いため, CTR モードの高速性を犠牲にすることはほとんどない.

復号 GCM モードでは, ブロック暗号の復号関数を利用しない. よって, GCM 暗号化, GCM 復号の両方の機能を実装する場合には (もちろん, $\text{GF}(2^{128})$ 上乗算のコストが新たに必要だが AES の暗号化処理に対して非常に軽いため), CBC や ECB に比較して軽いことが期待できる.

6.5 IACBC/XCBC

仕様の概要 IACBC[J01, J00], XCBC[GD01a, GD01b] とともに, CBC モードにおいてブロック暗号の出力をマスクすることで本質的なメッセージ認証の安全性を与えた利用モードである.

IACBC について説明する. 処理するメッセージ長を m ブロックとすると, 第一の鍵と初期値 (乱数) から, $\lceil \log_2 m \rceil$ ブロックのマスクの種 W_i

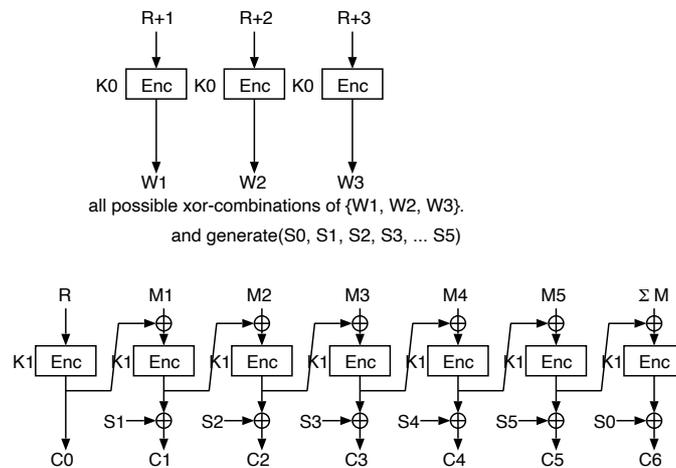


図 22: IACBC モードの暗号化処理を示すブロック図

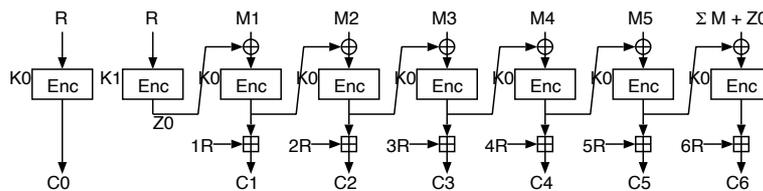


図 23: XCBC モードの暗号化処理を示すブロック図

を生成する。この W_i から約 m ブロック分の pairwise independent (ペア単位では独立) なブロック列 S_i を生成する ($\log_2 m$ 個の要素を含む集合から、可能な要素の組み合わせの数は $2^{\log_2 m} = m$ 通り)。

これら W_i ならびに S_i はメッセージ到着と同時に必要な時逐次的に生成が可能であるため、これらの処理はオンライン処理性を崩すことはない。

これら S_i 列をそれぞれブロック暗号の出力にマスクしながら第二の鍵で CBC モードのようなブロック連鎖を伴いながら暗号文を出力することが、スキームの主要部分となる。最後のブロックはメッセージのチェックサムとその暗号化のための末端処理がある。

一方、XCBC は、初期値である秘密乱数 R と二つの鍵から C_0, Z_0 を生成する。そして IACBC モードでいうところの S_i 列は、整数倍の R となり、ブロック毎に整数乗算 (おそらく 128 ビット幅) を行なう。暗号文の生成は、出力と S_i 列との算術加算の結果である。

仕様の定義を厳密に記すと、XCBC は暗号化を行なうものであり、秘

置のみを保持する。このモードを使って、平文に特殊な秘密パディングを施したもの(仕様書では、その一つを XCBC-XOR と呼んでいる)が認証暗号の機能を達成することができる。

安全性 IACBC, XCBC とともに、近年の共通鍵暗号の安全性に関する議論を踏まえた安全性の証明を示している。

XCBC/XCBC-XOR についても秘匿とメッセージ認証両方の観点からの攻撃者のアドバンテージの上限を与えている。XCBC が $(q', t'; \epsilon')$ -安全な擬似ランダム関数を用いているとすると、初期値が乱数である XCBC は Left-or-Right に関する秘匿の意味で $(q, t, \mu; \epsilon)$ -安全である。ただし

$$(t, \mu, \epsilon) = (t' - c\mu, q \cdot n, 2\epsilon' + (\mu^2/n^2 - \mu/n)2^{-n}).$$

XCBC-XOR に対するメッセージ認証に関する安全性として攻撃者の改竄成功確率の上限を与えている。 (q', t', ϵ') を秘匿の場合の定義と同じとして、

$$\begin{aligned} \text{Adv}_{\text{XCBC}}^{\text{auth}} \leq & \epsilon + \frac{\mu_v(\mu_v - n)}{n^2 2^{n+1}} + \frac{q_e(q_e - 1)}{2^{n+1}} + \frac{(q_e + 1)\mu_v}{n 2^n} \\ & + \frac{\mu_v}{n 2^{n+1}} (\log_2 \frac{\mu_v}{l} + 3) + \frac{q_v \mu_e}{n 2^n} (\log_2 \frac{\mu_e}{n} + 3). \end{aligned}$$

効率 メッセージ認証つき暗号化方式のスキームでは効率的な両方式である。IACBC は、メッセージのブロック数 m に対して $m + \log m$ 程度のブロック暗号呼び出しに加えて排他的論理和を基本とした補助演算が含まれる。メッセージ長が大きくなると、CBC や ECB に対する負荷処理は割合としてさほど小さくなる。

XCBC-XOR ではブロック暗号の呼び出し回数は ECB, CBC とほぼ同じである $m + 3$ 回程度の処理を行なうが、マスクの生成、ならびにマスク処理のために、算術加算(また実装によっては算術乗算)の処理が含まれる。これらは 128 ビットのレジスタで処理される演算である。

並列処理性など 並列処理性については CBC モードと同様である。暗号化においては、主要な演算部分の並列処理性は達成できない。ただし、XCBC では、CBC で適用されたようにインターリーブする手法が記述されている。

復号 復号では、ブロック暗号プリミティブの暗号化演算、復号演算両方を利用するため、復号処理では両方を同時に実装する必要がある。

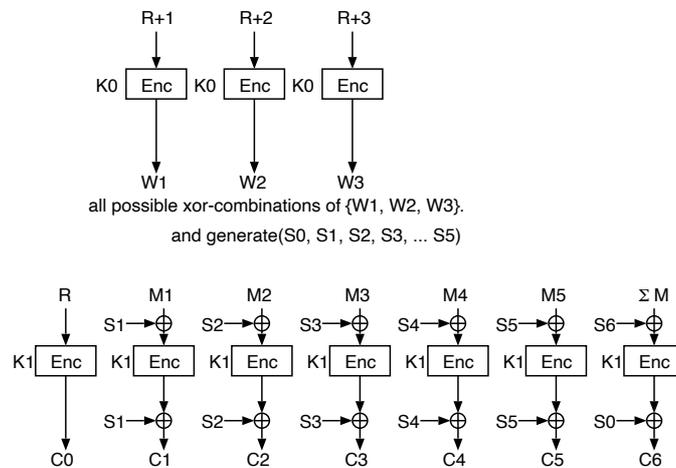


図 24: IAPM モードの暗号化処理を示すブロック図

6.6 IAPM/OCB

仕様の概要 IAPM[J00, J01], OCB[RBBK01a, RBBK01b] とともにブロック連鎖のない暗号処理である。主要な攪拌部分は、ECB モードにおいて、ブロック暗号の入出力部で、ブロック位置に応じた秘密マスクを行なうことである。

IAPM では、これら秘密マスクを $\log m$ ブロックの W_i 列の (擬似) 乱数ブロックから m ブロックの pairwise independent ブロックを生成している。一方、OCB では、秘密鍵と nonce から生成する (擬似) 乱数 2 ブロック、 L, R から S_i を生成している。具体的には、ブロック位置 i 番目には $\gamma_i L \oplus R$ を生成するような、線形式によるブロック列の生成である。

OCB では、これら線形列の生成が隣のブロックに対して排他的論理和を一度行なうだけでよいように gray code (自然数の並べかえであって、隣り合う整数どうしのバイナリ表現によるハミング距離が常に 1 であるような順列) の技術を使って生成している。

IAPM に比べて OCB は、後で提案されたこともあり様々な観点から改良と呼ぶことができる特徴がいくつもある。OCB では秘密鍵を 1 個利用する (IAPM は 2 個)。OCB では初期値として nonce であればよい (IAPM は乱数)。OCB では、ブロック暗号の呼び出しは m 回程度である (IAPM は約 $m + \log m$ 回)。OCB では端数処理の定義があり、パディングが本質的理由となる暗号文の増加が最小限に押えられている。

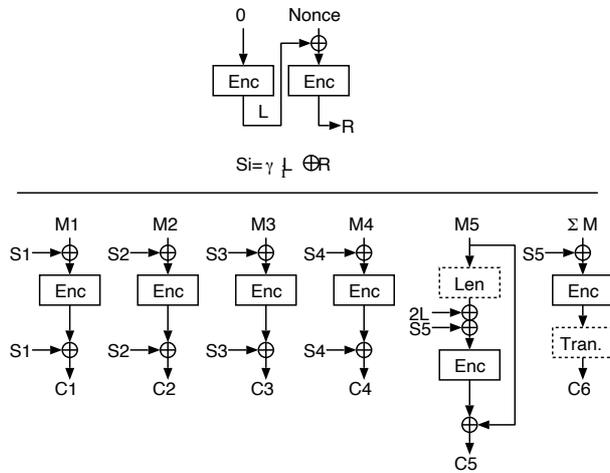


図 25: OCB モードの暗号化処理を示すブロック図

安全性 ランダム関数を内部で用いる OCB の、メッセージ認証に関するアドバンテージは以下のように示されている。

$$\text{Adv}_{\text{OCB}}^{\text{auth}} \leq 1.5(\mu_e + 3q + 5\mu_v + 11)^2/2^n + 2^{-\tau}.$$

ここで、 μ_e は攻撃者の q 回の暗号化オラクルへの質問で累積するブロック数、 μ_v は復号への試行の数、 τ はタグ長である。

一方、秘匿に関する安全性としては、暗号文-乱数不可識別性による秘匿の定義により評価を行っており、この場合、各パラメータ、変数はメッセージ認証と同じとして、ランダム関数を用いた OCB の秘匿に関する攻撃者のアドバンテージは以下ようになる。

$$\text{Adv}_{\text{OCB}}^{\text{ct}} \leq 1.5(\mu_e + 2q + 3)^2/2^n.$$

並列処理性など これらの利用モードはメッセージ認証可能な暗号方式であってかつ、暗号化、復号ともに並列処理性があることが大きな特徴である。ただし並列処理を行なう場合にも処理するブロックのブロック位置はプロセッサが知っておく必要がある。

復号 復号処理においても、ブロック暗号の暗号化関数が必要であるので、復号デバイスには両方を実装する必要がある。

6.7 k -PCFB

仕様の概要 k -CFB に変更を加えた利用モードである [H01c]. 従来の k -CFB モードは $k < n$ の時, 内部レジスタの更新に前の情報の内部レジスタ値を使っていた. このモードでは, ブロック暗号処理の出力の一部と暗号文を使って内部レジスタを更新する.

この利用モードとして, 特殊な平文 (前後に平文長がパディングされたもの) を使うことによりメッセージ認証も達成できると提案されている.

安全性 秘匿に関しては CFB の拡張の一種であり問題ないと考える.

メッセージ認証については特に安全性に関する技術的根拠が記載されていない. また, 提案者自身の評価も公開されていないため, あまり研究者の興味を集めたモードでない.

実際に, スキームへの改竄が可能であることは簡単に示すことができる. $k = n$ の時は CFB と等価であるため, ブロック単位のデータ欠損にはある程度の遅れを伴うもののすぐに同期が回復する. メッセージ中に長さ情報として読みとれる部分を二箇所, (その値で指定される) 適切な幅で挿入しておけば, データ欠損時にもその改竄が検出できない. 結果として部分だけを切りとる攻撃が既知平文攻撃により可能である.

7 ディスクセクタ向け暗号利用モード

IEEE の Security in Storage Working Group では, ハードディスクなどをセクタ単位で暗号化することを直接的な応用先として, 利用モードとその運用の観点から技術調査, 標準化を行なっている [WWW1].

この標準化における技術要件は, 平文が暗号文の長さから変化することがない暗号化であって, かつ暗号文に対するいかなる改竄によっても平文が攪拌されていることを保証するものである.

現在標準化が策定中であり, 提案された利用モードに関する情報は多くない. 本報告では, 提案方式を簡単に説明する.

7.1 EMD

EMD モードは Tweak 入力 (補助的な入力であって秘密情報ではないブロック暗号に対するパラメータのようなもの) をとりながら 2 パス処理により大きなブロックを攪拌する利用モードである [R02b].

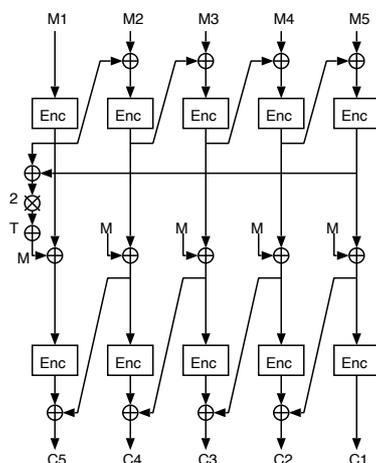


図 26: EMD モードの暗号化処理を示すブロック図

この利用モードは、証明にミスがあり、かつ効率的に PRP と識別可能であることが示された [J03].

7.2 EME

EME は、EMD の並列化可能なスキームに改良したものである。オリジナルのスキームは [R02b] で提案されたがこれについては攻撃方法が [J03] によって示された。これを改良した EME モードが [HR03a] で提案された。暗号化処理を図 27 に示す。

7.3 CMC

CMC は、EMD, EME モード [R02b, HR03a] に対して安全性の観点から改良した利用モードである [HR03b]。Tweak データ T と鍵 K をパラメータとして多ブロックを暗号化する方法である。構造は 3 層からなり、最初の層で CBC モードの暗号化のような処理を行なう。次の層では、その結果の一部を加工したデータを各ブロックにマスクする。最後の層では平文ブロックの逆の順序で CBC モードの復号のような処理を行なう。

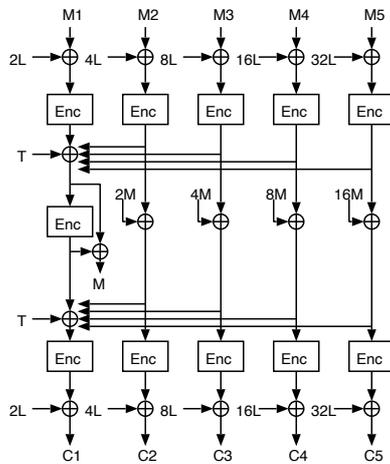


図 27: EME モードの暗号化の処理を示すブロック図

7.4 LRW

LRW は [LR02] で提案された Tweakable ブロック暗号である。これはディスクセクタ暗号の特にメッセージが 16 バイトと短いものに向けての操作モードとして提案された。

入力として鍵 K と Tweak I がある。鍵はさらにブロック暗号に使うための鍵 K_1 (ブロック暗号の鍵長) とマスク鍵を生成するための K_2 (ブロック長) からなる。暗号化処理では、有限体乗算により $K_2 \otimes I$ からマスク値 T を生成し、ブロック暗号処理前と処理後にそれぞれ T をマスクする。この処理を図 29 に示す。

7.5 NR

NR モードはブロック暗号の ECB モードに処理を加えた暗号処理方式である [NR03]。入力出力にそれぞれ拡張 Feistel 構造で構成される線形変換を導入する。入力側、出力側で二種類の線形変換を定義するが、各々の変換の内部では、3 つの universal hash を用いている。

8 認証に関する利用モード

本章では、認証に関する利用モードについて述べる。EMAC, RMAC, XCBC, TMAC, OMAC は CBC MAC の変形であり、XOR MAC, XECB

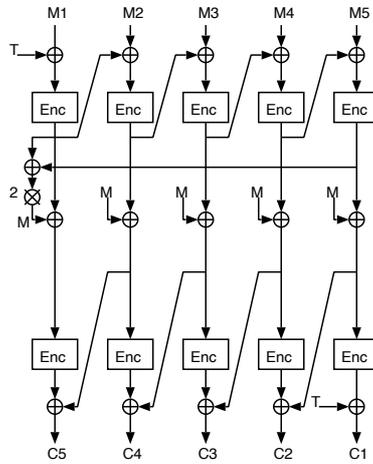


図 28: CMC モードの暗号化の処理を示すブロック図

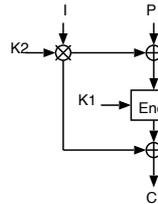


図 29: LRW モードの暗号化の処理を示すブロック図

MAC, PMAC は並列計算可能な方式である。また、 f_9 は 3GPP [3GPPa, 3GPPb] により標準化されている方式である。

以上の方式がブロック暗号を用いて構成されているのに対して、NMAC, HMAC はハッシュ関数を用いて構成される方式である。

8.1 CBC MAC

CBC MAC には、パディングの方法や、最終ブロックの処理など、いくつもの仕様がある。次に述べる仕様は、最も単純なものである。

方式 CBC MAC はブロック暗号 E 、タグ長 τ 、(メッセージ長を規定する) 定数 m をパラメータとする。ブロック長 n のブロック暗号 $E: \mathcal{K}_E \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ を用いた場合は、 $\tau \leq n$ でなくてはならな

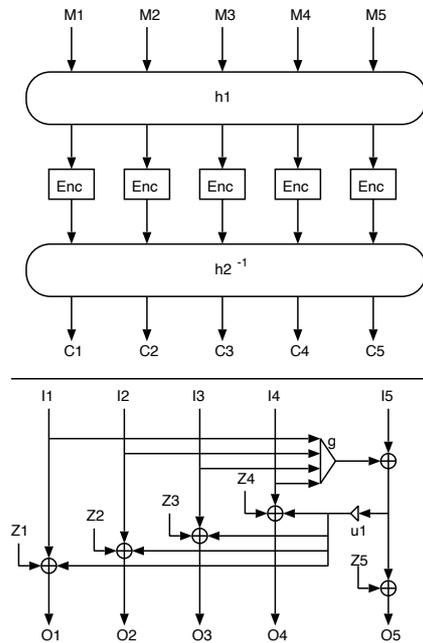


図 30: NR モードの暗号化処理 (上) と内部の universal hash の構成 (下) を示すブロック図

い. これらのパラメータを用いた CBC MAC を $\text{CBC}[E, \tau, m]$ と表記する. $\text{CBC}[E, \tau, m] = (\text{CBC-}\mathcal{K}, \text{CBC-}\mathcal{G}, \text{CBC-}\mathcal{V})$ の鍵生成アルゴリズム CBC- \mathcal{K} , タグ生成アルゴリズム CBC- \mathcal{G} , 確認アルゴリズム CBC- \mathcal{V} はそれぞれ以下のように動作する.

- 鍵生成アルゴリズム CBC- \mathcal{K} は確率的アルゴリズムであり,

$$K \stackrel{R}{\leftarrow} \mathcal{K}_E$$

を出力する.

- タグ生成アルゴリズム $\text{CBC-}\mathcal{G} : \mathcal{K}_E \times \{0, 1\}^{mn} \rightarrow \{0, 1\}^\tau$ は決定的アルゴリズムであり, 鍵空間は \mathcal{K}_E , メッセージ空間は $\{0, 1\}^{mn}$, タグ空間は $\{0, 1\}^\tau$ である. すなわち, 鍵 $K \in \mathcal{K}_E$ とメッセージ $M \in \{0, 1\}^{mn}$ を入力とし, タグ

$$T = \text{CBC-}\mathcal{G}_K(M) \in \{0, 1\}^\tau$$

を出力する. 図 31, 図 32 にあるように動作する. 図 32 において, trunc は n ビットの入力のうち, 左 τ ビットを出力する.

```

Algorithm CBC- $\mathcal{G}_K(M)$ 
 $Y[0] \leftarrow 0^n$ 
Partition  $M$  into  $M[1] \cdots M[m]$ 
for  $i \leftarrow 1$  to  $m$  do
     $X[i] \leftarrow M[i] \oplus Y[i-1]$ 
     $Y[i] \leftarrow E_K(X[i])$ 
 $T \leftarrow$  the left most  $\tau$  bits of  $Y[m]$ 
return  $T$ 

```

図 31: CBC MAC のタグ生成アルゴリズム $\text{CBC-}\mathcal{G}_K(\cdot)$.

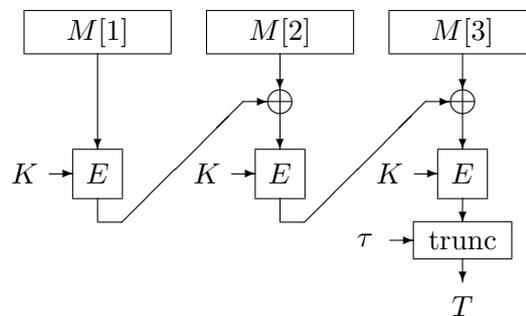


図 32: $M = M[1]M[2]M[3]$ の場合の $\text{CBC-}\mathcal{G}_K(M)$ の動作.

- 確認アルゴリズム $\text{CBC-}\mathcal{V} : \mathcal{K}_E \times \{0, 1\}^{mn} \times \{0, 1\}^\tau \rightarrow \text{accept or reject}$ は決定的アルゴリズムであり, 鍵 $K \in \mathcal{K}_E$, メッセージ $M \in \{0, 1\}^{mn}$, タグ $T \in \{0, 1\}^\tau$ を入力とし,

$$\text{accept or reject} = \text{CBC-}\mathcal{V}_K(M, T)$$

を出力する. 図 33 にあるように動作する.

Algorithm $\text{CBC-}\mathcal{V}_K(M, T)$
 $T' \leftarrow \text{CBC-}\mathcal{G}_K(M)$
if $T = T'$ **then return** accept
else return reject

図 33: CBC MAC の確認アルゴリズム $\text{CBC-}\mathcal{V}_K(\cdot, \cdot)$.

安全性 Bellare, Kilian, Rogaway [BKR00] と Bellare, Pietrzak, Rogaway [BPR05] とにより安全性が解析されている. ブロック暗号 E が安全な擬似ランダム置換族であれば, $\text{CBC}[E, \tau, m] = (\mathcal{K}_{\text{CBC}}, \mathcal{G}_{\text{CBC}}, \mathcal{V}_{\text{CBC}})$ は, 偽造不可能性の意味で安全な MAC であることが示されている. (タグ生成アルゴリズムは決定的なので, この場合, 弱偽造不可能性と強偽造不可能性は同一の定義になる.) 上記の解析結果より, 以下の定理が成立することがわかる.

定理 8.1. n, m, t, q を非負整数とし, $m^2 \leq 2^{n-2}$ とする. $E : \mathcal{K}_E \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ をブロック暗号とする. このとき,

$$\text{Adv}_{\text{CBC}[E, \tau, m]}^{\text{mac}}(t, q, mq) \leq \text{Adv}_E^{\text{prp}}(t', q') + \frac{m^2 q^2}{2^{n+1}} + \frac{12mq^2}{2^n} + \frac{8m^4 q^2}{2^{2n}} + \frac{1}{2^\tau}$$

である. ただし, $q' = mq$, $t' = t + O(nmq)$ であり, 質問の長さはブロック単位である.

直感的に, 定理 8.1 は, 以下のことを示している: 実行時間 t , 高々 q 回の質問の後,

$$\text{Adv}_{\text{CBC}[E, \tau, m]}^{\text{mac}}(A) = \epsilon$$

で偽造に成功する敵が存在すると仮定する. このとき, 実行時間 $t' = t + O(nmq)$, 高々 $q' = mq$ 回の質問の後,

$$\text{Adv}_E^{\text{prp}}(B) \geq \epsilon - \frac{m^2 q^2}{2^{n+1}} - \frac{12mq^2}{2^n} - \frac{8m^4 q^2}{2^{2n}} - \frac{1}{2^\tau}$$

なる敵 B が存在する.

しかし, 上記の定理はメッセージ空間が, ある定数 m に対し, $\{0, 1\}^{mn}$ となっていないなければならない. そうでない場合, 特に可変長のメッセージ空間 (たとえば $(\{0, 1\}^n)^+$) に対しては, CBC MAC は安全な MAC でなくなる. たとえば, 図 34 の敵 A は, メッセージ空間を $(\{0, 1\}^n)^+$ とした CBC MAC を偽造不可能性の意味で破る敵である. また, その成功確率は 1 である.

Algorithm $A^{\text{CBC-}\mathcal{G}_K(\cdot)}$ $M \leftarrow 0^n$ $T \leftarrow \text{CBC-}\mathcal{G}_K(M)$ $M' \leftarrow (M, T)$ return (M', T)

図 34: $A^{\text{CBC-}\mathcal{G}_K(\cdot)}$.

効率 CBC MAC の効率は, 以下のようにまとめられる.

- 鍵長: ブロック暗号の鍵 $K \in \mathcal{K}_E$ 一つのみである.
- ブロック暗号鍵スケジューリングの呼び出し回数: 1 回である.
- メッセージ M に対するタグを生成するのにかかるブロック暗号の呼び出し回数: $(|M|/n)$ 回の呼び出しである.
- 事前計算するべきブロック暗号の呼び出し回数: 必要ない.
- 並列処理性: 並列処理はできない.

標準化状況 広範囲にわたって標準化されている. FIPS 113, ISO 9797, ISO 8731-1, ISO 9807, ANSI X9.9, ANSI X9.19 に含まれている.

なお, メッセージ長の問題を解決するために, パディング, 最終出力の前に暗号化を施すなど, いくつかの変形がある. 正確な仕様については, 各標準の文書を参照されたい.

8.2 EMAC

方式 EMAC はブロック暗号 E とタグ長 τ をパラメータとする。ブロック長 n のブロック暗号 $E: \mathcal{K}_E \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ を用いた場合は, $\tau \leq n$ でなくてはならない。これらのパラメータを用いた EMAC を $\text{EMAC}[E, \tau]$ と表記する。 $\text{EMAC}[E, \tau] = (\text{EMAC-}\mathcal{K}, \text{EMAC-}\mathcal{G}, \text{EMAC-}\mathcal{V})$ の鍵生成アルゴリズム $\text{EMAC-}\mathcal{K}$, タグ生成アルゴリズム $\text{EMAC-}\mathcal{G}$, 確認アルゴリズム $\text{EMAC-}\mathcal{V}$ はそれぞれ以下のように動作する。

- 鍵生成アルゴリズム $\text{EMAC-}\mathcal{K}$ は確率的アルゴリズムであり,

$$K_1 \stackrel{R}{\leftarrow} \mathcal{K}_E$$

と

$$K_2 \stackrel{R}{\leftarrow} \mathcal{K}_E$$

を出力する。

- タグ生成アルゴリズム $\text{EMAC-}\mathcal{G}: (\mathcal{K}_E)^2 \times (\{0, 1\}^n)^+ \rightarrow \{0, 1\}^\tau$ は決定的アルゴリズムであり, 鍵空間は $(\mathcal{K}_E)^2$, メッセージ空間は $(\{0, 1\}^n)^+$, タグ空間は $\{0, 1\}^\tau$ である。すなわち, 鍵 $K_1, K_2 \in \mathcal{K}_E$ とメッセージ $M \in (\{0, 1\}^n)^+$ を入力とし, タグ

$$T = \text{EMAC-}\mathcal{G}_{K_1, K_2}(M) \in \{0, 1\}^\tau$$

を出力する。図 35, 図 36 にあるように動作する。

- 確認アルゴリズム $\text{EMAC-}\mathcal{V}: (\mathcal{K}_E)^2 \times (\{0, 1\}^n)^+ \times \{0, 1\}^\tau \rightarrow \text{accept or reject}$ は決定的アルゴリズムであり, 鍵 $K_1, K_2 \in \mathcal{K}_E$, メッセージ $M \in (\{0, 1\}^n)^+$, タグ $T \in \{0, 1\}^\tau$ を入力とし,

$$\text{accept or reject} = \text{EMAC-}\mathcal{V}_{K_1, K_2}(M, T)$$

を出力する。図 37 にあるように動作する。

安全性 Petrank, Rackoff により, 安全性が解析されている [PR00]。ブロック暗号 E が安全な擬似ランダム置換族であれば, $\text{EMAC}[E, \tau]$ は, 偽造不可能性の意味で安全な MAC であることが示されている。以下の定理が示されている。

```

Algorithm EMAC- $\mathcal{G}_{K_1, K_2}(M)$ 
 $Y[0] \leftarrow 0^n$ 
Partition  $M$  into  $M[1] \cdots M[m]$ 
for  $i \leftarrow 1$  to  $m$  do
     $X[i] \leftarrow M[i] \oplus Y[i-1]$ 
     $Y[i] \leftarrow E_{K_1}(X[i])$ 
 $T \leftarrow$  the left most  $\tau$  bits of  $E_{K_2}(Y[m])$ 
return  $T$ 

```

図 35: EMAC のタグ生成アルゴリズム EMAC- $\mathcal{G}_{K_1, K_2}(\cdot)$.

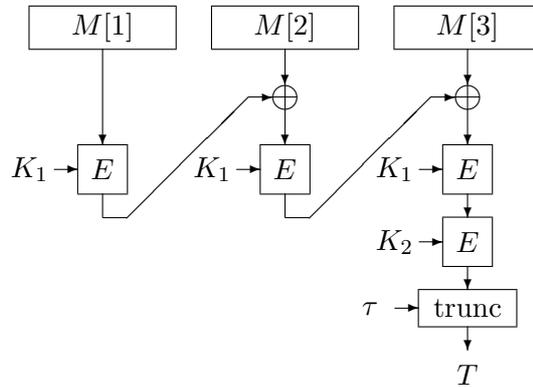


図 36: $M = M[1]M[2]M[3]$ の場合の EMAC- $\mathcal{G}_{K_1, K_2}(M)$ の動作.

```

Algorithm EMAC- $\mathcal{V}_{K_1, K_2}(M, T)$ 
 $T' \leftarrow$  EMAC- $\mathcal{G}_{K_1, K_2}(M)$ 
if  $T = T'$  then return accept
else return reject

```

図 37: EMAC の確認アルゴリズム EMAC- $\mathcal{V}_{K_1, K_2}(\cdot, \cdot)$.

定理 8.2. $n, \tau \geq 1$ を整数, $t, q, \sigma \geq 1$ を $\sigma^2 \leq 2^{(n+1)/2}$ なる整数とする. $E : \mathcal{K}_E \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ をブロック暗号とする. このとき,

$$\text{Adv}_{\text{EMAC}[E, \tau]}^{\text{mac}}(t, q, \sigma) \leq 2\text{Adv}_E^{\text{prp}}(t', q') + \frac{3\sigma^2}{2^n} + \frac{1}{2^{\tau-1}}$$

である. ただし, $q' = \sigma, t' = t + O(n\sigma)$ であり, 質問の長さはブロック単位である.

定理 8.2 は, 以下のことを示している: 実行時間 t , 高々 q 回の質問をし, それらの質問が合計で高々 σ ブロックであり,

$$\text{Adv}_{\text{EMAC}[E, \tau]}^{\text{mac}}(A) = \epsilon$$

なる敵 A が存在すると仮定する. このとき, 実行時間 $t' = t + O(n\sigma)$, 高々 $q' = \sigma$ 回の質問の後,

$$\text{Adv}_E^{\text{prp}}(B) \geq \frac{\epsilon}{2} - \frac{3\sigma^2}{2^{n+1}} - \frac{1}{2^\tau}$$

なる敵 B が存在する.

効率 EMAC の効率は, 以下のようにまとめられる.

- 鍵長: ブロック暗号の鍵 $K_1, K_2 \in \mathcal{K}_E$ の二つである.
- ブロック暗号鍵スケジューリングの呼び出し回数: 2 回である.
- メッセージ M に対するタグを生成するのにかかるブロック暗号の呼び出し回数: $(|M|/n) + 1$ 回の呼び出しである.
- 事前計算するべきブロック暗号の呼び出し回数: 必要ない.
- 並列処理性: 並列処理はできない.

標準化状況 ISO/IEC 9797-1 に含まれている [ISOIEC9797-1]. また, NESSIE の portfolio にも含まれている [WWW8]. 実装が容易であること, 証明可能安全であること, 効率と鍵スケジューリングが妥当であることなどが挙げられている [WWW8].

8.3 RMAC

RMAC にはいくつかのバージョンが存在する。Jaulmes, Joux, Vallete によって提案されたオリジナルの RMAC [JJ+02a, JJ+02b] と NIST が SP800-38B のドラフト版 [SP800-38B] で提案した RMAC である。それぞれ RMAC-JJV と RMAC-NIST と表記することにする。

Jaulmes, Joux, Vallete によって NIST に提案された文書 [JJ+02c] には二つの方式が提案されている。メッセージのパディングの仕方が異なり、それぞれ、RMAC-JJV1, RMAC-JJV2 と表記する。

方式 (RMAC-JJV1) はじめに、RMAC-JJV1 について述べる。

$$\text{RMAC-JJV1} = (\text{RMAC-JJV1-}\mathcal{K}, \text{RMAC-JJV1-}\mathcal{G}, \text{RMAC-JJV1-}\mathcal{V})$$

の鍵生成アルゴリズム RMAC-JJV1- \mathcal{K} , タグ生成アルゴリズム RMAC-JJV1- \mathcal{G} , 確認アルゴリズム RMAC-JJV1- \mathcal{V} はそれぞれ以下のように動作する。

$\text{Perm}(n)$ を n ビット上のすべての置換の集合とし、 r を整数とする。ランダム置換 f_1 とは、 $\text{Perm}(n)$ から一様ランダムに選んだ f_1 である。 $f_1 \stackrel{R}{\leftarrow} \text{Perm}(n)$ と表記する。置換族 F_2 を以下のように定義する。

$$F_2 = \left\{ f_2^{(R)} \mid R \in \{0, 1\}^r, f_2^{(R)} \in \text{Perm}(n) \right\}$$

すなわち、各 $R \in \{0, 1\}^r$ に対し、インデックス R を持つ置換 $f_2^{(R)} \in \text{Perm}(n)$ からなる集合である。

- 鍵生成アルゴリズム RMAC-JJV1- \mathcal{K} は確率的アルゴリズムであり、 $f_1 \stackrel{R}{\leftarrow} \text{Perm}(n)$ と、各 $R \in \{0, 1\}^r$ に対し、

$$f_2^{(R)} \stackrel{R}{\leftarrow} \text{Perm}(n)$$

を出力する。

すなわち、鍵空間は $\text{Perm}(n) \times F_2$ であり、鍵は、

$$f_1, f_2^{(0, \dots, 0)}, f_2^{(0, \dots, 1, 0)}, \dots, f_2^{(1, \dots, 1)}$$

となる。

- タグ生成アルゴリズム RMAC-JJV1- \mathcal{G} : $(\text{Perm}(n) \times F_2) \times \{0, 1\}^* \rightarrow (\{0, 1\}^n \times \{0, 1\}^r)$ は確率的アルゴリズムであり、鍵空間は $\text{Perm}(n) \times F_2$, メッセージ空間は $\{0, 1\}^*$, タグ空間は $\{0, 1\}^n \times \{0, 1\}^r$ である。

鍵 $f_1 \in \text{Perm}(n)$, $f_2^{(R)}$ ($R \in \{0, 1\}^r$) とメッセージ $M \in \{0, 1\}^*$ を入力とし, タグ

$$T = \text{RMAC-JJV1-}\mathcal{G}_{f_1, f_2^{(R)}}(M) \in \{0, 1\}^n \times \{0, 1\}^r$$

を出力する. 図 38, 図 39 にあるように動作する.

Algorithm RMAC-JJV1- $\mathcal{G}_{f_1, f_2^{(R)}}(M)$

$R \xleftarrow{R} \{0, 1\}^r$

Pad M

$Y[0] \leftarrow 0^n$

Partition M into $M[1] \cdots M[m]$

for $i \leftarrow 1$ **to** m **do**

$X[i] \leftarrow M[i] \oplus Y[i-1]$

$Y[i] \leftarrow f_1(X[i])$

$T' \leftarrow f_2^{(R)}(Y[m])$

$T \leftarrow (T', R)$

return T

図 38: RMAC-JJV1 のタグ生成アルゴリズム RMAC-JJV1- $\mathcal{G}_{f_1, f_2^{(\cdot)}}(\cdot)$.

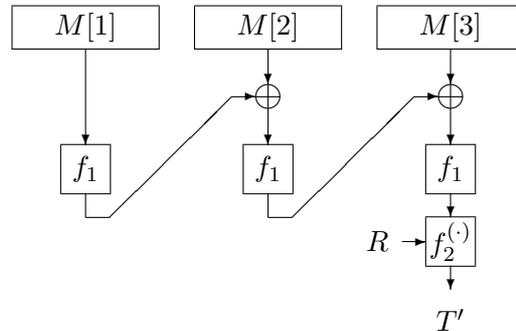


図 39: $M = M[1]M[2]M[3]$ の場合の RMAC-JJV1- $\mathcal{G}_{f_1, f_2^{(R)}}(M)$ の動作.

まず, 2 行目で r ビットの乱数 R を生成し, この R をインデックスにもつ $f_2^{(R)}$ を最終ブロックの暗号化の際に用いる. 3 行目では, M に 1 を連結してから, 全体が n ビットの倍数になるよう 0 を連

結する。すなわち、

$$M \leftarrow M \| 1 \| 0^{n-1-|M| \bmod n}$$

とする。 M がすでに n の整数倍である場合は、 10^{n-1} を連結する。

- 確認アルゴリズム $\text{RMAC-JJV1-}\mathcal{V}$: $(\text{Perm}(n) \times F_2) \times \{0, 1\}^* \times (\{0, 1\}^n \times \{0, 1\}^r) \rightarrow \text{accept or reject}$ は決定的アルゴリズムであり、鍵 $f_1 \in \text{Perm}(n)$, $f_2^{(\cdot)} \in F_2$, メッセージ $M \in \{0, 1\}^*$, タグ $T \in (\{0, 1\}^n \times \{0, 1\}^r)$ を入力とし、

$$\text{accept or reject} = \text{RMAC-JJV1-}\mathcal{V}_{f_1, f_2^{(R)}}(M, T)$$

を出力する。図 40 にあるように動作する。

<p>Algorithm $\text{RMAC-JJV1-}\mathcal{V}_{f_1, f_2^{(R)}}(M, (T', R))$ Pad M $Y[0] \leftarrow 0^n$ Partition M into $M[1] \cdots M[m]$ for $i \leftarrow 1$ to m do $X[i] \leftarrow M[i] \oplus Y[i-1]$ $Y[i] \leftarrow f_1(X[i])$ $T'' \leftarrow f_2^{(R)}(Y[m])$ if $T' = T''$ then return accept else return reject</p>
--

図 40: RMAC-JJV1 の確認アルゴリズム $\text{RMAC-JJV1-}\mathcal{V}_{f_1, f_2^{(\cdot)}}(\cdot, \cdot)$.

方式 (RMAC-JJV2) RMAC-JJV2 は、パディングの仕方が RMAC-JJV1 とは異なる。メッセージ長が n の倍数の場合は、 10^{n-1} を連結する必要がなく、ブロック暗号の呼び出し回数を 1 回削減できる。

メッセージ長が n の倍数である場合は、 $f_2^{(R)}$ を、そうでない場合は、それとは異なる $f_2'^{(R)}$ を用いる。

AES を用いた方式 (RMAC-JJV1) ブロック暗号として AES を用いた場合の RMAC-JJV1 の実装方法が提案されている。まず、 $r = 128$ として、128 ビット乱数 R を生成する。 K_1 を 128 ビット鍵、 K_2 を 128

ビット, もしくは 256 ビット鍵とする. f_1 として, AES_{K_1} を, $f_2^{(R)}$ として, $\text{AES}_{K_2 \oplus R}$ (K_2 が 128 ビットの場合), もしくは $\text{AES}_{K_2 \oplus (R \parallel 0^{128})}$ (K_2 が 256 ビットの場合) とする.

AES を用いた方式 (RMAC-JJV2) ブロック暗号として AES を用いた場合の RMAC-JJV2 の実装方法が提案されている. まず, K_1 を 128 ビット鍵, K_2 を 192 ビット, もしくは 256 ビット鍵とする. RMAC-JJV2 では K_2 の長さが R の長さよりも長くなくてはならないので, 128 ビットの K_2 を用いることはできない.

$r = 128$ として, 128 ビット乱数 R' を生成する. f_1 として, AES_{K_1} をもちいる. メッセージ長が n の倍数の場合, $f_2^{(R)}$ として, $\text{AES}_{K_2 \oplus R}$, $R = (R' \parallel 1 \parallel 0^{63})$ (K_2 が 192 ビットの場合) もしくは $\text{AES}_{K_2 \oplus R}$, $R = (R' \parallel 1 \parallel 0^{127})$ (K_2 が 256 ビットの場合) とする.

メッセージ長が n の倍数ではない場合, $f_2^{(R)}$ として, $\text{AES}_{K_2 \oplus R}$, $R = (R' \parallel 0 \parallel 0^{63})$ (K_2 が 192 ビットの場合) もしくは $\text{AES}_{K_2 \oplus R}$, $R = (R' \parallel 0 \parallel 0^{127})$ (K_2 が 256 ビットの場合) とする.

方式 (RMAC-NIST) NIST は SP800-38B のドラフト版で RMAC を提案した. R の扱いがオリジナルとは異なる. オリジナルでは, R が乱数であったのに対し, NIST の提案ではカウンタになることが許されていた. また, RMAC-JJV1 に対応する提案のみであり, RMAC-JJV2 に対応する提案はなかった.

RMAC-NIST はパラメータとして, ブロック暗号 $E : \{0, 1\}^k \times \{0, 1\}^n \rightarrow \{0, 1\}^n$, タグ長 τ , R の長さ r をとる. これらのパラメータを用いた場合, $\text{RMAC-NIST}[E, \tau, r]$ と表記する.

$\text{RMAC-NIST}[E, \tau, r] = (\text{RMAC-NIST-}\mathcal{K}, \text{RMAC-NIST-}\mathcal{G}, \text{RMAC-NIST-}\mathcal{V})$ の鍵生成アルゴリズム RMAC-NIST- \mathcal{K} , タグ生成アルゴリズム RMAC-NIST- \mathcal{G} , 確認アルゴリズム RMAC-NIST- \mathcal{V} はそれぞれ以下のように動作する.

- 鍵生成アルゴリズム RMAC-NIST- \mathcal{K} は確率的アルゴリズムであり,

$$K_1, K_2 \stackrel{R}{\leftarrow} \{0, 1\}^k$$

を出力する.

- タグ生成アルゴリズム RMAC-NIST- $\mathcal{G} : (\{0, 1\}^k)^2 \times \{0, 1\}^r \times \{0, 1\}^* \rightarrow \{0, 1\}^r \times \{0, 1\}^\tau$ は決定的アルゴリズムであり, 鍵空間は $(\{0, 1\}^k)^2$, メッセージ空間は $\{0, 1\}^*$, タグ空間は $\{0, 1\}^r \times \{0, 1\}^\tau$ である. さらに NIST の仕様では R はタグ生成アルゴリズムへの入力として

扱われる。すなわち，鍵 $K_1, K_2 \in \{0, 1\}^k$, $R \in \{0, 1\}^r$, メッセージ $M \in \{0, 1\}^*$ を入力とし，タグ

$$T = \text{RMAC-NIST-}\mathcal{G}_{K_1, K_2}(R, M) \in \{0, 1\}^r \times \{0, 1\}^\tau$$

を出力する。図 41, 図 42 にあるように動作する。

```

Algorithm RMAC-NIST- $\mathcal{G}_{K_1, K_2}(R, M)$ 
Pad  $M$ 
 $Y[0] \leftarrow 0^n$ 
Partition  $M$  into  $M[1] \cdots M[m]$ 
for  $i \leftarrow 1$  to  $m$  do
     $X[i] \leftarrow M[i] \oplus Y[i-1]$ 
     $Y[i] \leftarrow E_{K_1}(X[i])$ 
if  $r = 0$  then  $K_3 \leftarrow K_2$ 
    else  $K_3 \leftarrow K_2 \oplus (R \parallel 0^{k-r})$ 
 $T' \leftarrow$  the left most  $\tau$  bits of  $E_{K_3}(Y[m])$ 
 $T \leftarrow (R, T')$ 
return  $T$ 

```

図 41: RMAC-NIST のタグ生成アルゴリズム RMAC-NIST- $\mathcal{G}_{K_1, K_2}(\cdot, \cdot)$.

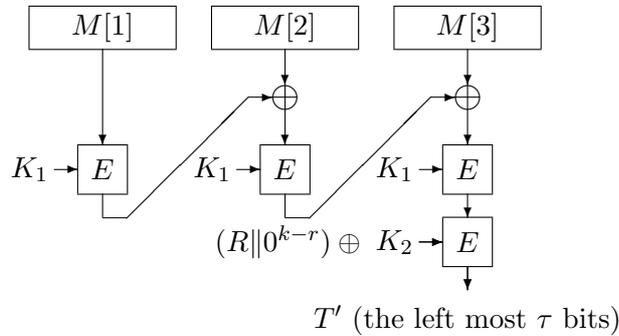


図 42: $M = M[1]M[2]M[3]$ の場合の RMAC-NIST- $\mathcal{G}_{K_1, K_2}(R, M)$ の動作.

3 行目では， M に 1 を連結してから，全体が n ビットの倍数になるよう 0 を連結する。すなわち，

$$M \leftarrow M \parallel 1 \parallel 0^{n-1-|M| \bmod n}$$

とする。 M がすでに n の整数倍である場合は， 10^{n-1} を連結する。

- 確認アルゴリズム $\text{RMAC-NIST-}\mathcal{V} : (\{0, 1\}^k)^2 \times \{0, 1\}^* \times (\{0, 1\}^r \times \{0, 1\}^\tau) \rightarrow \text{accept or reject}$ は決定的アルゴリズムであり, 鍵 $K_1, K_2 \in \{0, 1\}^k$, メッセージ $M \in \{0, 1\}^*$, タグ $T \in (\{0, 1\}^r \times \{0, 1\}^\tau)$ を入力とし,

$$\text{accept or reject} = \text{RMAC-NIST-}\mathcal{V}_{K_1, K_2}(M, T)$$

を出力する. 図 43 にあるように動作する.

Algorithm $\text{RMAC-NIST-}\mathcal{V}_{K_1, K_2}(M, (R, T'))$

Pad M

$Y[0] \leftarrow 0^n$

Partition M into $M[1] \cdots M[m]$

for $i \leftarrow 1$ **to** m **do**

$X[i] \leftarrow M[i] \oplus Y[i-1]$

$Y[i] \leftarrow E_{K_1}(X[i])$

if $r = 0$ **then** $K_3 \leftarrow K_2$

else $K_3 \leftarrow K_2 \oplus (R \parallel 0^{k-r})$

$T'' \leftarrow$ the left most τ bits of $E_{K_2}(Y[m])$

if $T' = T''$ **then return** accept

else return reject

図 43: RMAC-NIST の確認アルゴリズム $\text{RMAC-NIST-}\mathcal{V}_{K_1, K_2}(\cdot, \cdot)$.

パラメータについて RMAC-NIST はパラメータとして, ブロック暗号 $E : \{0, 1\}^k \times \{0, 1\}^n \rightarrow \{0, 1\}^n$, タグ長 τ , R の長さ r をとる. E としては, AES-128, AES-192, AES-256, Triple DES-112, Triple DES-168 のいずれかを, τ と r については, 図 44 を提案している.

Parameter Set II ~ V は一般的な使用に適している, と述べられている.

安全性 ランダム置換 f_1 とランダム置換族 $f_2^{(R)}$ ($R \in \{0, 1\}^R$) を用いた RMAC-JJV1 と RMAC-JJV2 について安全性が解析されている [JJ+02a, JJ+02b]. 安全性の定義は, 一般的な強偽造不能性に近いが, タグが異なっていない点で異なる.

敵は, タグ生成オラクルと確認オラクルをもつ. タグ生成オラクルに q 個のメッセージ M_1, \dots, M_q を質問し, その答え T_1, \dots, T_q を得たとする.

Parameter Set	$n = 128$		$n = 64$	
	r	τ	r	τ
I	0	32	0	32
II	0	64	64	64
III	16	80	n/a	
IV	64	96	n/a	
V	128	128	n/a	

図 44: RMAC-NIST のパラメータ.

また、確認オラクルに q' 個のメッセージ、タグのペア $(M'_1, T'_1), \dots, (M'_{q'}, T'_{q'})$ を質問したとする。

ある i に対し、 $MAC\text{-}\mathcal{V}_K(M'_i, T'_i) = \text{accept}$ であり、 $T'_i \notin \{T_1, \dots, T_j\}$ であれば、 A は偽造に成功した、という。 $\{T_1, \dots, T_j\}$ は、 (M'_i, T'_i) を確認オラクルに質問する以前に、タグ生成オラクルから返ってきた答えである。

直感的には、メッセージは見たことがあってもよいが、タグは見たことがあってはならない。たとえば、タグ生成オラクルに M_1 を送り、 T_1 を得たとする。次に、タグ生成オラクルに M_2 を送り、 T_2 を得たとする。強偽造不能性の定義では (M_1, T_2) を偽造文として許すが、上記安全性の定義では、これは偽造文ではない。

ここで、アドバンテージ $\text{Adv}^{\text{rmac-uf}}(A)$ を以下のように定義する。

$$\text{Adv}^{\text{rmac-uf}}(A) \stackrel{\text{def}}{=} \Pr(f_1 \stackrel{R}{\leftarrow} \text{Perm}(n), f_2 \stackrel{R}{\leftarrow} F_2 :$$

$A^{\text{RMAC-JJV-}\mathcal{G}_{f_1, f_2}^{(R)}(\cdot), \text{RMAC-JJV-}\mathcal{V}_{f_1, f_2}^{(R)}(\cdot, \cdot)}$ が上記の意味で偽造に成功)

以下の定理が示されている [JJ+02a, JJ+02b].

定理 8.3. $n \geq 2$ を整数、 $r = n$ とする。 A を高々 σ ブロックの質問をする敵とする。このとき、

$$\text{Adv}^{\text{rmac-uf}}(A) \leq \frac{4n\sigma + 4\sigma + 2}{2^n}$$

である。

上記の安全性のバウンドはほかに比べ、非常に小さいことがわかる。 $\sigma \approx 2^{n/2}$ とすると、XCBC や OMAC のバウンドは 1 を超えるのに対し、上記のアドバンテージは小さい値のままである。

ただし、上記の定理は帰着を示していない。ランダム置換やランダム置換族を鍵として持つのは、非現実的である。

AES を用いた実装に対しても安全性解析がなされている [JJ+02a, JJ+02b] が、該当箇所には議論の不備が指摘されている [R02a]。XCBC や OMAC のように、ブロック暗号の擬似ランダム性に安全性を帰着させる結果は知られていない。[R02b] では帰着することは不可能である、と述べられている。

効率 RMAC の効率は、以下のようにまとめられる。

- 鍵長：ブロック暗号の鍵 $K_1, K_2 \in \mathcal{K}_E$ の二つである。
- ブロック暗号鍵スケジューリングの呼び出し回数：鍵生成アルゴリズム実行時に 1 回必要であり、さらにタグ生成アルゴリズム、もしくは確認アルゴリズムを呼び出すごとに 1 回必要である。
- メッセージ M に対するタグを生成するのにかかるブロック暗号の呼び出し回数： $(|M|/n) + 1$ 回の呼び出しである。
- 事前計算するべきブロック暗号の呼び出し回数：必要ない。
- 並列処理性：並列処理はできない。

標準化状況 NIST に提案され、2002 年 10 月、SP800-38B のドラフト版が提案された。

これに対し、いくつかのコメントが寄せられた。Knudsen は、Triple DES を用いたときの安全性の問題点を指摘した [K02]。Rogaway [R02b]、Wagner [W02a]、Black [B02] は、いずれも、ブロック暗号の擬似ランダム性に RMAC の安全性が帰着できない点を指摘した。また、そもそも birthday bound をこえる安全性への疑問も出された。MAC は多くの場合、暗号化方式と組み合わせて使用される。多くの暗号化方式、たとえば CTR や CBC は birthday bound をこえる安全性を有していない。これらの暗号化方式は birthday bound に到達すると、平文に関する情報を漏洩する。これらのコメントはいずれも、RMAC の決定を見直すべきだと主張している。

NIST は 2003 年 6 月、RMAC の決定を見直し、OMAC を提案すると発表し [WWW9]、2005 年 5 月、OMAC を正式に推奨方式として採用した [SP800-38B]。

NESSIE でも考慮されたが、最終的には portfolio には含まれなかった [WWW8]。

8.4 XCBC

方式 XCBC はブロック暗号 E とタグ長 τ をパラメータとする。ブロック長 n のブロック暗号 $E: \mathcal{K}_E \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ を用いた場合は, $\tau \leq n$ でなくてはならない。これらのパラメータを用いた XCBC を $\text{XCBC}[E, \tau]$ と表記する。 $\text{XCBC}[E, \tau] = (\text{XCBC-}\mathcal{K}, \text{XCBC-}\mathcal{G}, \text{XCBC-}\mathcal{V})$ の鍵生成アルゴリズム $\text{XCBC-}\mathcal{K}$, タグ生成アルゴリズム $\text{XCBC-}\mathcal{G}$, 確認アルゴリズム $\text{XCBC-}\mathcal{V}$ はそれぞれ以下のように動作する。

- 鍵生成アルゴリズム $\text{XCBC-}\mathcal{K}$ は確率的アルゴリズムであり,

$$K_1 \stackrel{R}{\leftarrow} \mathcal{K}_E$$

$$K_2 \stackrel{R}{\leftarrow} \{0, 1\}^n$$

$$K_3 \stackrel{R}{\leftarrow} \{0, 1\}^n$$

を出力する。

- タグ生成アルゴリズム $\text{XCBC-}\mathcal{G}: (\mathcal{K}_E \times (\{0, 1\}^n)^2) \times \{0, 1\}^* \rightarrow \{0, 1\}^\tau$ は決定的アルゴリズムであり, 鍵空間は $\mathcal{K}_E \times (\{0, 1\}^n)^2$, メッセージ空間は $\{0, 1\}^*$, タグ空間は $\{0, 1\}^\tau$ である。すなわち, 鍵 $K_1 \in \mathcal{K}_E, K_2, K_3 \in \{0, 1\}^n$ とメッセージ $M \in \{0, 1\}^*$ を入力とし, タグ

$$T = \text{XCBC-}\mathcal{G}_{K_1, K_2, K_3}(M) \in \{0, 1\}^\tau$$

を出力する。図 45, 図 46 にあるように動作する。XCBC は M の長さが n の倍数でなくてもよい。図 45 の 3 行目において,

$$M = M[1]M[2] \cdots M[m-1]M[m]$$

は, $|M[1]| = |M[2]| = \cdots = |M[m-1]|$ かつ $1 \leq |M[m]| \leq n$ となるように分割される。 $M = \varepsilon$ のときは例外である。この場合, $|M[m]| = 0$ となる。

また, 図 45 の 7 行目の関数 $\text{pad}_n: \{0, 1\}^{\leq n} \rightarrow \{0, 1\}^n$ は以下のように定義される。 a を長さが高々 n ビットのビット列とする ($a = \varepsilon$ でもよい)。このとき,

$$\text{pad}_n(a) = \begin{cases} a10^{n-|a|-1} & \text{if } |a| < n, \\ a & \text{if } |a| = n. \end{cases} \quad (1)$$

```

Algorithm XCBC- $\mathcal{G}_{K_1, K_2, K_3}(M)$ 
 $Y[0] \leftarrow 0^n$ 
Partition  $M$  into  $M[1] \cdots M[m]$ 
for  $i \leftarrow 1$  to  $m - 1$  do
     $X[i] \leftarrow M[i] \oplus Y[i - 1]$ 
     $Y[i] \leftarrow E_{K_1}(X[i])$ 
 $X[m] \leftarrow \text{pad}_n(M[m]) \oplus Y[m - 1]$ 
if  $|M[m]| = n$  then  $X[m] \leftarrow X[m] \oplus K_2$ 
    else  $X[m] \leftarrow X[m] \oplus K_3$ 
 $T \leftarrow$  the left most  $\tau$  bits of  $E_{K_1}(Y[m])$ 
return  $T$ 

```

図 45: XCBC のタグ生成アルゴリズム XCBC- $\mathcal{G}_{K_1, K_2, K_3}(\cdot)$.

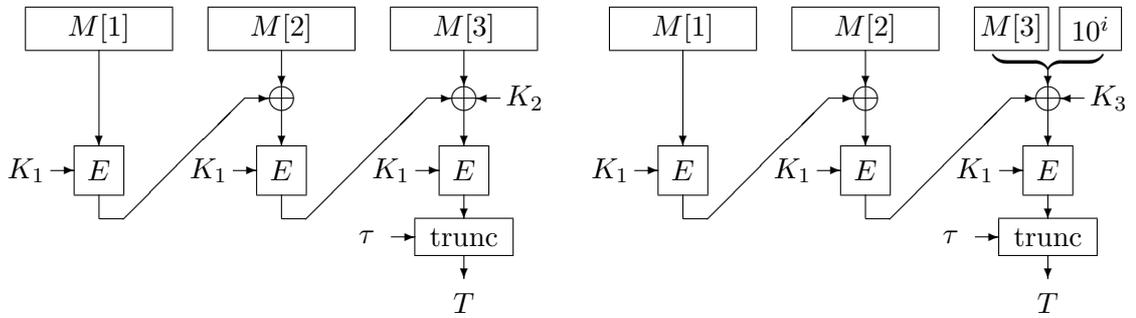


図 46: $M = M[1]M[2]M[3]$ の場合の XCBC- $\mathcal{G}_{K_1, K_2, K_3}(M)$ の動作.

- 確認アルゴリズム $\text{XCBC-}\mathcal{V} : (\mathcal{K}_E \times (\{0, 1\}^n)^2) \times \{0, 1\}^* \times \{0, 1\}^\tau \rightarrow \text{accept or reject}$ は決定的アルゴリズムであり, 鍵 $K_1 \in \mathcal{K}_E, K_2, K_3 \in \{0, 1\}^n$, メッセージ $M \in \{0, 1\}^*$, タグ $T \in \{0, 1\}^\tau$ を入力とし,

$$\text{accept or reject} = \text{XCBC-}\mathcal{V}_{K_1, K_2, K_3}(M, T)$$

を出力する. 図 47 にあるように動作する.

Algorithm $\text{XCBC-}\mathcal{V}_{K_1, K_2, K_3}(M, T)$
 $T' \leftarrow \text{XCBC-}\mathcal{G}_{K_1, K_2, K_3}(M)$
if $T = T'$ **then return** accept
else return reject

図 47: XCBC の確認アルゴリズム $\text{XCBC-}\mathcal{V}_{K_1, K_2, K_3}(\cdot, \cdot)$.

安全性 Black, Rogaway により, 安全性が解析されている [BR00]. ブロック暗号 E が安全な擬似ランダム置換族であれば, $\text{XCBC}[E, \tau]$ は, 偽造不可能性の意味で安全な MAC であることが示されている. 以下の定理が示されている [IK03b].

定理 8.4. $n, \tau \geq 1$ を整数, $t, q, \sigma \geq 1$ を $\sigma^2 \leq 2^{(n+1)/2}$ なる整数とする. $E : \mathcal{K}_E \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ をブロック暗号とする. このとき,

$$\text{Adv}_{\text{XCBC}[E, \tau]}^{\text{mac}}(t, q, \sigma) \leq \text{Adv}_E^{\text{prp}}(t', q') + \frac{3\sigma^2}{2^n} + \frac{1}{2^\tau}$$

である. ただし, $q' = \sigma, t' = t + O(n\sigma)$ であり, 質問の長さはブロック単位である.

定理 8.4 は, 以下のことを示している: 実行時間 t , 高々 q 回の質問をし, それらの質問が合計で高々 σ ブロックであり,

$$\text{Adv}_{\text{XCBC}[E, \tau]}^{\text{mac}}(A) = \epsilon$$

なる敵 A が存在すると仮定する. このとき, 実行時間 $t' = t + O(n\sigma)$, 高々 $q' = \sigma$ 回の質問をし,

$$\text{Adv}_E^{\text{prp}}(B) \geq \epsilon - \frac{3\sigma^2}{2^n} - \frac{1}{2^\tau}$$

なる敵 B が存在する.

効率 XCBC の効率は、以下のようにまとめられる。

- 鍵長：ブロック暗号の鍵 $K_1 \in \mathcal{K}_E$ と n ビットの鍵 $K_2, K_3 \in \{0, 1\}^n$ の計 3 つが必要である。
- ブロック暗号鍵スケジューリングの呼び出し回数：1 回である。
- メッセージ M に対するタグを生成するのにかかるブロック暗号の呼び出し回数： $\max\{1, \lceil |M|/n \rceil\}$ 回の呼び出しである。
- 事前計算するべきブロック暗号の呼び出し回数：必要ない。
- 並列処理性：並列処理はできない。

標準化状況 NIST に提案されている [WWW9]。また、[FH03] や [H03b] で議論されている。

8.5 TMAC

方式 TMAC はブロック暗号 E とタグ長 τ をパラメータとする。ブロック長 n のブロック暗号 $E: \mathcal{K}_E \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ を用いた場合は、 $\tau \leq n$ でなくてはならない。これらのパラメータを用いた TMAC を $\text{TMAC}[E, \tau]$ と表記する。TMAC $[E, \tau] = (\text{TMAC-}\mathcal{K}, \text{TMAC-}\mathcal{G}, \text{TMAC-}\mathcal{V})$ の鍵生成アルゴリズム TMAC- \mathcal{K} 、タグ生成アルゴリズム TMAC- \mathcal{G} 、確認アルゴリズム TMAC- \mathcal{V} はそれぞれ以下のように動作する。

- 鍵生成アルゴリズム TMAC- \mathcal{K} は確率的アルゴリズムであり、

$$K_1 \stackrel{R}{\leftarrow} \mathcal{K}_E$$

と

$$K_2 \stackrel{R}{\leftarrow} \{0, 1\}^n$$

を出力する。

- タグ生成アルゴリズム TMAC- $\mathcal{G}: (\mathcal{K}_E \times \{0, 1\}^n) \times \{0, 1\}^* \rightarrow \{0, 1\}^\tau$ は決定的アルゴリズムであり、鍵空間は $\mathcal{K}_E \times \{0, 1\}^n$ 、メッセージ空間は $\{0, 1\}^*$ 、タグ空間は $\{0, 1\}^\tau$ である。すなわち、鍵 $K_1 \in \mathcal{K}_E$ 、 $K_2 \in \{0, 1\}^n$ とメッセージ $M \in \{0, 1\}^*$ を入力とし、タグ

$$T = \text{TMAC-}\mathcal{G}_{K_1, K_2}(M) \in \{0, 1\}^\tau$$

を出力する. 図 48, 図 49 にあるように動作する. 図 48, 図 49 において, $K_2 \cdot u$ は, $\text{GF}(2^n)$ 上の乗算である. 一般的に $a \in \{0, 1\}^n$ に対し,

$$a \cdot u = \begin{cases} a \ll 1 & \text{if } \text{msb}(a) = 0, \\ (a \ll 1) \oplus \text{Cst}_n & \text{otherwise} \end{cases} \quad (2)$$

となる. ここで, (2) において, $a \ll 1$ は a の左 1 ビットシフトを表し, $a = a_{n-1}a_{n-2} \cdots a_1a_0$ と a をビット表現した場合, $a \ll 1 = a_{n-2}a_{n-3} \cdots a_00$ となる. すなわち, 最上位ビットはなくなり, 最下位ビットに 0 が補充される. また, $\text{msb}(a)$ は a の最上位ビットを表し, Cst_n は n ビットの定数である. たとえば, $\text{Cst}_{128} = 0^{120}10000111$ であり, $\text{Cst}_{64} = 0^{59}11011$ である. TMAC も XCBC と同様, M

Algorithm $\text{TMAC-}\mathcal{G}_{K_1, K_2}(M)$
 $Y[0] \leftarrow 0^n$
Partition M into $M[1] \cdots M[m]$
for $i \leftarrow 1$ **to** $m - 1$ **do**
 $X[i] \leftarrow M[i] \oplus Y[i - 1]$
 $Y[i] \leftarrow E_{K_1}(X[i])$
 $X[m] \leftarrow \text{pad}_n(M[m]) \oplus Y[m - 1]$
if $|M[m]| = n$ **then** $X[m] \leftarrow X[m] \oplus K_2 \cdot u$
 else $X[m] \leftarrow X[m] \oplus K_2$
 $T \leftarrow$ the left most τ bits of $E_{K_1}(Y[m])$
return T

図 48: TMAC のタグ生成アルゴリズム $\text{TMAC-}\mathcal{G}_{K_1, K_2}(\cdot)$.

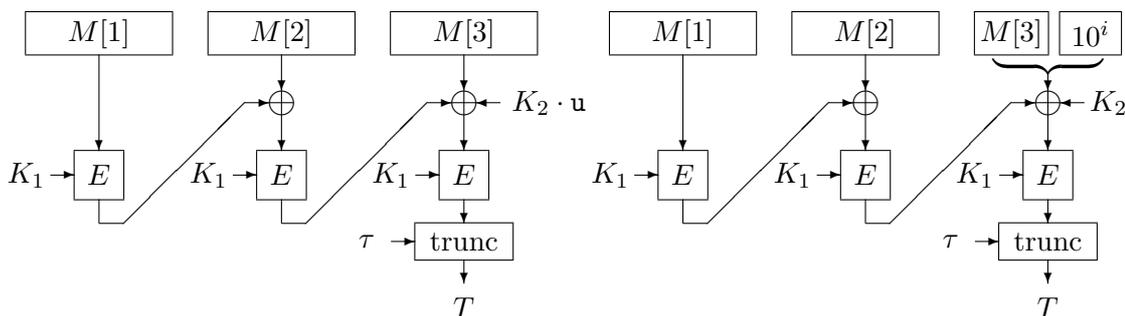


図 49: $M = M[1]M[2]M[3]$ の場合の $\text{TMAC-}\mathcal{G}_{K_1, K_2}(M)$ の動作.

の長さが n の倍数でなくてもよい。図 48 の 3 行目において、

$$M = M[1]M[2] \cdots M[m-1]M[m]$$

は、 $|M[1]| = |M[2]| = \cdots = |M[m-1]|$ かつ $1 \leq |M[m]| \leq n$ となるように分割される。 $M = \varepsilon$ のときは例外で、この場合、 $|M[m]| = 0$ となる。

また、図 48 の 7 行目の関数 $\text{pad}_n : \{0, 1\}^{\leq n} \rightarrow \{0, 1\}^n$ は (1) のように定義される。

- 確認アルゴリズム $\text{TMAC-}\mathcal{V} : (\mathcal{K}_E \times \{0, 1\}^n) \times \{0, 1\}^* \times \{0, 1\}^\tau \rightarrow \text{accept or reject}$ は決定的アルゴリズムであり、鍵 $K_1 \in \mathcal{K}_E, K_2 \in \{0, 1\}^n$, メッセージ $M \in \{0, 1\}^*$, タグ $T \in \{0, 1\}^\tau$ を入力とし、

$$\text{accept or reject} = \text{TMAC-}\mathcal{V}_{K_1, K_2}(M, T)$$

を出力する。図 50 にあるように動作する。

Algorithm $\text{TMAC-}\mathcal{V}_{K_1, K_2}(M, T)$
 $T' \leftarrow \text{TMAC-}\mathcal{G}_{K_1, K_2}(M)$
if $T = T'$ **then return** accept
else return reject

図 50: TMAC の確認アルゴリズム $\text{TMAC-}\mathcal{V}_{K_1, K_2}(\cdot, \cdot)$.

安全性 Kurosawa, Iwata により、安全性が解析されている [KI03]. ブロック暗号 E が安全な擬似ランダム置換族であれば、 $\text{TMAC}[E, \tau]$ は、偽造不可能性の意味で安全な MAC であることが示されている。以下の定理が示されている [IK03b].

定理 8.5. $n, \tau \geq 1$ を整数, $t, q, \sigma \geq 1$ を $\sigma^2 \leq 2^{(n+1)/2}$ なる整数とする。 $E : \mathcal{K}_E \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ をブロック暗号とする。このとき、

$$\text{Adv}_{\text{TMAC}[E, \tau]}^{\text{mac}}(t, q, \sigma) \leq \text{Adv}_E^{\text{prp}}(t', q') + \frac{3\sigma^2}{2^n} + \frac{1}{2^\tau}$$

である。ただし、 $q' = \sigma, t' = t + O(n\sigma)$ であり、質問の長さはブロック単位である。

定理 8.5 は、以下のことを示している： 実行時間 t , 高々 q 回の質問をし、それらの質問が合計で高々 σ ブロックであり、

$$\mathbf{Adv}_{\text{TMAC}[E,\tau]}^{\text{mac}}(A) = \epsilon$$

なる敵 A が存在すると仮定する。このとき、実行時間 $t' = t + O(n\sigma)$, 高々 $q' = \sigma$ 回の質問をし、

$$\mathbf{Adv}_E^{\text{prp}}(B) \geq \epsilon - \frac{3\sigma^2}{2^n} - \frac{1}{2^\tau}$$

なる敵 B が存在する。

効率 TMAC の効率は、以下のようにまとめられる。

- 鍵長：ブロック暗号の鍵 $K_1 \in \mathcal{K}_E$ と n ビットの鍵 $K_2 \in \{0,1\}^n$ の計 2 つが必要である。
- ブロック暗号鍵スケジューリングの呼び出し回数：1 回である。
- メッセージ M に対するタグを生成するのにかかるブロック暗号の呼び出し回数： $\max\{1, \lceil |M|/n \rceil\}$ 回の呼び出しである。
- 事前計算するべきブロック暗号の呼び出し回数：必要ない。
- 並列処理性：並列処理はできない。

標準化状況 NIST に提案されている [WWW9].

8.6 OMAC/CMAC

OMAC は 2 つの方式 OMAC1 と OMAC2 の総称である。OMAC1 は NIST の SP800-38B に採用され、CMAC と呼ばれている。

方式 (OMAC1) OMAC1 はブロック暗号 E とタグ長 τ をパラメータとする。ブロック長 n のブロック暗号 $E: \mathcal{K}_E \times \{0,1\}^n \rightarrow \{0,1\}^n$ を用いた場合は、 $\tau \leq n$ でなくてはならない。これらのパラメータを用いた OMAC1 を $\text{OMAC1}[E, \tau]$ と表記する。

$$\text{OMAC1}[E, \tau] = (\text{OMAC1-}\mathcal{K}, \text{OMAC1-}\mathcal{G}, \text{OMAC1-}\mathcal{V})$$

の鍵生成アルゴリズム OMAC1- \mathcal{K} , タグ生成アルゴリズム OMAC1- \mathcal{G} , 確認アルゴリズム OMAC1- \mathcal{V} はそれぞれ以下のように動作する。

- 鍵生成アルゴリズム $\text{OMAC1-}\mathcal{K}$ は確率的アルゴリズムであり,

$$K \stackrel{R}{\leftarrow} \mathcal{K}_E$$

を出力する.

- タグ生成アルゴリズム $\text{OMAC1-}\mathcal{G} : \mathcal{K}_E \times \{0,1\}^* \rightarrow \{0,1\}^\tau$ は決定的アルゴリズムであり, 鍵空間は \mathcal{K}_E , メッセージ空間は $\{0,1\}^*$, タグ空間は $\{0,1\}^\tau$ である. すなわち, 鍵 $K \in \mathcal{K}_E$ とメッセージ $M \in \{0,1\}^*$ を入力とし, タグ

$$T = \text{OMAC1-}\mathcal{G}_K(M) \in \{0,1\}^\tau$$

を出力する. 図 51, 図 52 にあるように動作する. 図 51, 図 52 において, $L \cdot u$ は (2) によって得られ, $L \cdot u^2$ は $(L \cdot u) \cdot u$ として, (2) によって得られる. OMAC1 も XCBC , TMAC と同様, M の長さが n の倍数でなくてもよい. 図 51 の 3 行目において,

$$M = M[1]M[2] \cdots M[m-1]M[m]$$

は, $|M[1]| = |M[2]| = \cdots = |M[m-1]|$ かつ $1 \leq |M[m]| \leq n$ となるように分割される. $M = \varepsilon$ のときは例外で, この場合, $|M[m]| = 0$ となる.

また, 図 51 の 7 行目の関数 $\text{pad}_n : \{0,1\}^{\leq n} \rightarrow \{0,1\}^n$ は (1) のように定義される.

- 確認アルゴリズム $\text{OMAC1-}\mathcal{V} : \mathcal{K}_E \times \{0,1\}^* \times \{0,1\}^\tau \rightarrow \text{accept or reject}$ は決定的アルゴリズムであり, 鍵 $K \in \mathcal{K}_E$, メッセージ $M \in \{0,1\}^*$, タグ $T \in \{0,1\}^\tau$ を入力とし,

$$\text{accept or reject} = \text{OMAC1-}\mathcal{V}_K(M, T)$$

を出力する. 図 53 にあるように動作する.

方式 (**OMAC2**) OMAC2 は OMAC1 とほぼ同様である. OMAC1 中の $L \cdot u^2$ を $L \cdot u^{-1}$ としたものが OMAC2 である. 一般に $a \in \{0,1\}^n$ に対し,

$$a \cdot u^{-1} = \begin{cases} a \gg 1 & \text{if } \text{lsb}(a) = 0, \\ (a \gg 1) \oplus \text{Cst}'_n & \text{otherwise.} \end{cases} \quad (3)$$

となる. ここで, 上記 (3) において, $a \gg 1$ は a の右 1 ビットシフトを表す. $a = a_{n-1}a_{n-2} \cdots a_1a_0$ と a をビット表現した場合, $a \gg 1 =$

```

Algorithm OMAC1- $\mathcal{G}_K(M)$ 
 $L \leftarrow E_K(0^n)$ 
 $Y[0] \leftarrow 0^n$ 
Partition  $M$  into  $M[1] \cdots M[m]$ 
for  $i \leftarrow 1$  to  $m - 1$  do
     $X[i] \leftarrow M[i] \oplus Y[i - 1]$ 
     $Y[i] \leftarrow E_K(X[i])$ 
 $X[m] \leftarrow \text{pad}_n(M[m]) \oplus Y[m - 1]$ 
if  $|M[m]| = n$  then  $X[m] \leftarrow X[m] \oplus L \cdot u$ 
    else  $X[m] \leftarrow X[m] \oplus L \cdot u^2$ 
 $T \leftarrow$  the left most  $\tau$  bits of  $E_K(Y[m])$ 
return  $T$ 

```

図 51: OMAC1 のタグ生成アルゴリズム OMAC1- $\mathcal{G}_K(\cdot)$.

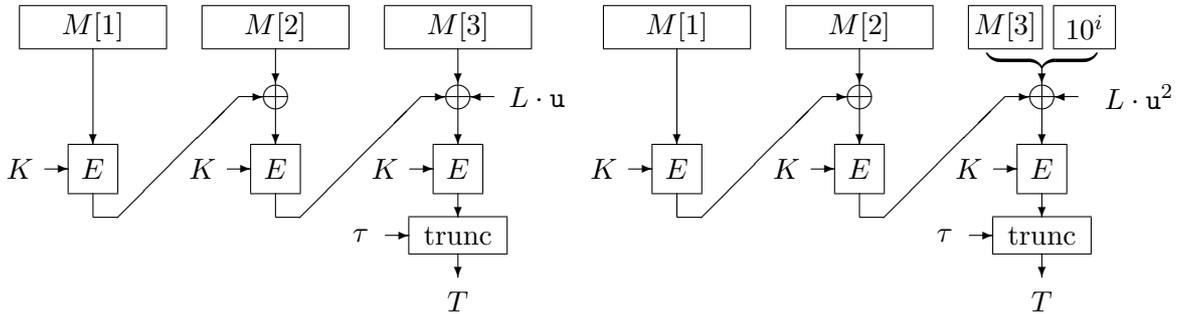


図 52: $M = M[1]M[2]M[3]$ の場合の OMAC1- $\mathcal{G}_K(M)$ の動作.

```

Algorithm OMAC1- $\mathcal{V}_K(M, T)$ 
 $T' \leftarrow$  OMAC1- $\mathcal{G}_K(M)$ 
if  $T = T'$  then return accept
    else return reject

```

図 53: OMAC1 の確認アルゴリズム OMAC1- $\mathcal{V}_K(\cdot, \cdot)$.

$0a_{n-1}a_{n-2}\cdots a_2a_1$ となる。すなわち、最下位ビットはなくなり、最上位ビットに 0 が補充される。また、 $\text{lsb}(a)$ は a の最下位ビットを表し、 Cst'_n は n ビットの定数である。たとえば、 $\text{Cst}'_{128} = 10^{120}1000011$ である。

安全性 Iwata, Kurosawa により、安全性が解析されている [IK03a]。ブロック暗号 E が安全な擬似ランダム置換族であれば、 $\text{OMAC1}[E, \tau]$ と $\text{OMAC2}[E, \tau]$ は、偽造不可能性の意味で安全な MAC であることが示されている。以下の定理が示されている [IK03b]。

定理 8.6. $n, \tau \geq 1$ を整数、 $t, q, \sigma \geq 1$ を $\sigma^2 \leq 2^{(n+1)/2}$ なる整数とする。 $E : \mathcal{K}_E \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ をブロック暗号とする。このとき、

$$\begin{cases} \text{Adv}_{\text{OMAC1}[E, \tau]}^{\text{mac}}(t, q, \sigma) \leq \text{Adv}_E^{\text{PRP}}(t', q') + \frac{4\sigma^2}{2^n} + \frac{1}{2^\tau} \\ \text{Adv}_{\text{OMAC2}[E, \tau]}^{\text{mac}}(t, q, \sigma) \leq \text{Adv}_E^{\text{PRP}}(t', q') + \frac{4\sigma^2}{2^n} + \frac{1}{2^\tau} \end{cases}$$

である。ただし、 $q' = \sigma, t' = t + O(n\sigma)$ であり、質問の長さはブロック単位である。

定理 8.6 は、以下のことを示している：実行時間 t 、高々 q 回の質問をし、それらの質問が合計で高々 σ ブロックであり、

$$\text{Adv}_{\text{OMAC1}[E, \tau]}^{\text{mac}}(A) = \epsilon$$

なる敵 A が存在すると仮定する。このとき、実行時間 $t' = t + O(n\sigma)$ 、高々 $q' = \sigma$ 回の質問をし、

$$\text{Adv}_E^{\text{PRP}}(B) \geq \epsilon - \frac{4\sigma^2}{2^n} - \frac{1}{2^\tau}$$

なる敵 B が存在する。OMAC2 についても同様である。

効率 OMAC の効率は、以下のようにまとめられる。

- 鍵長：ブロック暗号の鍵 $K \in \mathcal{K}_E$ の一つのみである。
- ブロック暗号鍵スケジューリングの呼び出し回数：1 回である。
- メッセージ M に対するタグを生成するのにかかるブロック暗号の呼び出し回数： $\max\{1, \lceil |M|/n \rceil\}$ 回の呼び出しである。
- 事前計算するべきブロック暗号の呼び出し回数： $L = E_K(0^n)$ を計算するのに 1 回必要である。
- 並列処理性：並列処理はできない。

標準化状況 NIST は 2005 年 5 月, OMAC1 を推奨方式として採用した [SP800-38B]. なお, SP 800-38B では, OMAC1 は CMAC (Cipher-based MAC) と呼ばれている.

8.7 XOR MAC

2 つの方式が提案されており, 一つは乱数を用いる XMACR 方式, もう一つはカウンタを用いる XMACC 方式である. どちらも関数族 $F : \{0, 1\}^k \times \{0, 1\}^n \rightarrow \{0, 1\}^{n'}$ を用いる. F としてブロック暗号を用いてもよいが, 入力長 n と出力長 n' は異なってもよい. XMACR, XMACC どちらも, パラメータとして, F と整数 b をとる. ただし, $b \leq n-1$ でなくてはならない. メッセージ M は $|M| \leq b \cdot 2^{n-b-1}$ であり, 長さが b の整数倍になるようにパディングされているとする. たとえば,

$$M \leftarrow M \parallel 10^{(b-|M|-1) \bmod b} \quad (4)$$

とする. 関数 $\text{tag} : \{0, 1\}^k \times (\{0, 1\}^b)^+ \times \{0, 1\}^{n-1} \rightarrow \{0, 1\}^{n'}$ を以下のように定義する.

$$\text{tag}(K, M, r) \stackrel{\text{def}}{=} F_K(0 \parallel r) \oplus F_K(1 \parallel \langle 1 \rangle_{n-b-1} \parallel M[1]) \oplus \cdots \oplus F_K(1 \parallel \langle m \rangle_{n-b-1} \parallel M[m])$$

ただし, $M = M[1] \cdots M[m]$ はパディングされていて, 各 $M[1], \dots, M[m]$ は b ビット, r は $(n-1)$ ビット, $\langle i \rangle_{n-b-1}$ は整数 i の $(n-b-1)$ ビット表現である. 以下, $\text{tag}(K, M, r)$ を $\text{tag}_K(M, r)$ と表記する. 図 54 参照.

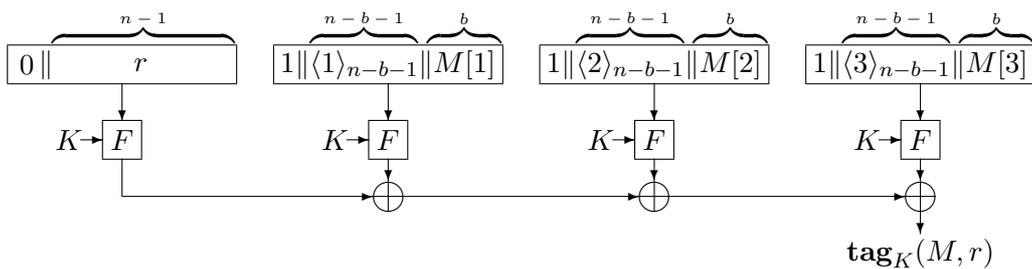


図 54: $M = M[1]M[2]M[3]$ の場合の $\text{tag}_K(M, r)$ の動作.

方式 (XMACR) XMACR は、関数族 F とブロック長 n をパラメータとする。関数族 $F : \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^{n'}$ を用いた場合は、 $b \leq n-1$ でなくてはならない。これらのパラメータを用いた XMACR を $\text{XMACR}[F, b]$ と表記する。 $\text{XMACR}[F, b] = (\text{XMACR-}\mathcal{K}, \text{XMACR-}\mathcal{G}, \text{XMACR-}\mathcal{V})$ の鍵生成アルゴリズム XMACR- \mathcal{K} , タグ生成アルゴリズム XMACR- \mathcal{G} , 確認アルゴリズム XMACR- \mathcal{V} はそれぞれ以下のように動作する。

- 鍵生成アルゴリズム XMACR- \mathcal{K} は確率的アルゴリズムであり、

$$K \stackrel{R}{\leftarrow} \{0,1\}^k$$

を出力する。

- タグ生成アルゴリズム $\text{XMACR-}\mathcal{G} : \{0,1\}^k \times \{0,1\}^* \rightarrow (\{0,1\}^{n-1} \times \{0,1\}^{n'})$ は確率的アルゴリズムであり、鍵空間は $\{0,1\}^k$, メッセージ空間は $\{0,1\}^*$, タグ空間は $(\{0,1\}^{n-1} \times \{0,1\}^{n'})$ である。すなわち、鍵 $K \in \{0,1\}^k$ とメッセージ $M \in \{0,1\}^*$ を入力とし、タグ

$$T = \text{XMACR-}\mathcal{G}_K(M) \in (\{0,1\}^{n-1} \times \{0,1\}^{n'})$$

を出力する。図 55 にあるように動作する。

Algorithm XMACR- $\mathcal{G}_K(M)$
 Pad M
 $r \stackrel{R}{\leftarrow} \{0,1\}^{n-1}$
 $T' \leftarrow \text{tag}_K(M, r)$
 $T \leftarrow (r, T')$
return T

図 55: XMACR のタグ生成アルゴリズム XMACR- $\mathcal{G}_K(\cdot)$.

ただし、2 行目は $M \in \{0,1\}^*$ に対しパディングを施し、 $M \in (\{0,1\}^n)^+$ となるようにする。たとえば、(4) のようにする。3 行目では $n-1$ ビットの乱数 r を選ぶ。

- 確認アルゴリズム $\text{XMACR-}\mathcal{V} : \{0,1\}^k \times \{0,1\}^* \times (\{0,1\}^{n-1} \times \{0,1\}^{n'}) \rightarrow \text{accept or reject}$ は決定的アルゴリズムであり、鍵 $K \in$

$\{0, 1\}^k$, メッセージ $M \in \{0, 1\}^*$, タグ $T \in (\{0, 1\}^{n-1} \times \{0, 1\}^{n'})$ を入力とし,

$$\text{accept or reject} = \text{XMACR-}\mathcal{V}_K(M, T)$$

を出力する. 図 56 にあるように動作する.

Algorithm XMACR- $\mathcal{V}_K(M, (r, T'))$
 Pad M
 $T'' \leftarrow \text{tag}_K(M, r)$
if $T' = T''$ **then return** accept
else return reject

図 56: XMACR の確認アルゴリズム XMACR- $\mathcal{V}_K(\cdot, \cdot)$.

ただし, 2 行目は $M \in \{0, 1\}^*$ に対し, (4) のようにする.

方式 (XMACC) XMACC は, XMACR と同様, 関数族 F とブロック長 n をパラメータとする. 関数族 $F : \{0, 1\}^k \times \{0, 1\}^n \rightarrow \{0, 1\}^{n'}$ を用いた場合は, $b \leq n-1$ でなくてはならない. これらのパラメータを用いた XMACC を XMACC[F, b] と表記する.

$$\text{XMACC}[F, b] = (\text{XMACC-}\mathcal{K}, \text{XMACC-}\mathcal{G}, \text{XMACC-}\mathcal{V})$$

の鍵生成アルゴリズム XMACC- \mathcal{K} , タグ生成アルゴリズム XMACC- \mathcal{G} , 確認アルゴリズム XMACC- \mathcal{V} はそれぞれ以下のように動作する.

- 鍵生成アルゴリズム XMACC- \mathcal{K} は確率的アルゴリズムであり,

$$K \stackrel{R}{\leftarrow} \{0, 1\}^k$$

を出力する.

- タグ生成アルゴリズム XMACC- $\mathcal{G} : \{0, 1\}^k \times \{0, 1\}^* \times \text{Count} \rightarrow (\text{Count} \times \{0, 1\}^{n'})$ は決定的アルゴリズムであり, 鍵空間は $\{0, 1\}^k$, メッセージ空間は $\{0, 1\}^*$, タグ空間は $(\text{Count} \times \{0, 1\}^{n'})$ である. さらに入力として, カウンタ $C \in \text{Count}$ をとる. Count はカウンタの空間であり, $\text{Count} = \{0, 1, \dots, 2^{n-1} - 1\}$ である. カウンタは送信者により保持されており, 0 に初期化され, タグ生成アルゴリズムによって更新される. 受信者はこれを保持しない. すなわち, 鍵

$K \in \{0, 1\}^k$, メッセージ $M \in \{0, 1\}^*$, カウンタ $C \in \text{Count}$ を入力とし, タグ

$$T = \text{XMACC-}\mathcal{G}_K(M) \in (\text{Count} \times \{0, 1\}^{n'})$$

を出力する. 図 57 にあるように動作する.

Algorithm XMACC- $\mathcal{G}_K(M, C)$
 Pad M
 $C \leftarrow C + 1$
 $T' \leftarrow \text{tag}_K(M, \langle C \rangle_{n-1})$
 $T \leftarrow (C, T')$
return T

図 57: XMACC のタグ生成アルゴリズム XMACC- $\mathcal{G}_K(\cdot)$.

ただし, $\langle C \rangle_{n-1}$ はカウンタ C の $(n-1)$ ビット表現であり, 2 行目は $M \in \{0, 1\}^*$ に対し, (4) のようにしてから, $M = M[1] \cdots M[m]$ とする.

- 確認アルゴリズム XMACC- $\mathcal{V} : \{0, 1\}^k \times \{0, 1\}^* \times (\text{Count} \times \{0, 1\}^{n'}) \rightarrow \text{accept or reject}$ は決定的アルゴリズムであり, 鍵 $K \in \{0, 1\}^k$, メッセージ $M \in \{0, 1\}^*$, タグ $T \in (\text{Count} \times \{0, 1\}^{n'})$ を入力とし,

$$\text{accept or reject} = \text{XMACC-}\mathcal{V}_K(M, T)$$

を出力する. 図 58 にあるように動作する.

Algorithm XMACC- $\mathcal{V}_K(M, (C, T'))$
 $T'' \leftarrow \text{tag}_K(M, C)$
if $T' = T''$ **then return** accept
else return reject

図 58: XMACC の確認アルゴリズム XMACC- $\mathcal{V}_K(\cdot, \cdot)$.

安全性 Bellare, Guérin, Rogaway により, 安全性が解析されている [BGR95]. 関数族 F が安全な擬似ランダム関数族であれば, $\text{XMCCR}[F, b]$ と $\text{XMCCR}[F, b]$ は, 強偽造不可能性の意味で安全な MAC であることが示されている.

擬似ランダム関数族 関数族 $F : \{0, 1\}^k \times \{0, 1\}^n \rightarrow \{0, 1\}^{n'}$ に対し、「 F が擬似ランダム関数族である」とは、適応的選択平文攻撃を行う任意の敵が、関数族 $\{F_K(\cdot) \in \text{Rand}(n, n') \mid K \in \{0, 1\}^k\}$ と $\{0, 1\}^n$ から $\{0, 1\}^{n'}$ へのすべての関数の集合 $\text{Rand}(n, n')$ を区別できないことをいう。

より厳密には、敵 A として、オラクルにアクセスできるアルゴリズムを考える。何回かの質問の後、 A は 1 ビットを出力する。関数族 $F : \{0, 1\}^k \times \{0, 1\}^n \rightarrow \{0, 1\}^{n'}$ の、敵 A に対する、擬似ランダム関数としての安全性は、アドバンテージ $\text{Adv}_F^{\text{prf}}(A)$ によって評価される。ここで、

$$\text{Adv}_F^{\text{prf}}(A) \stackrel{\text{def}}{=} |\Pr(K \stackrel{R}{\leftarrow} \{0, 1\}^k : A^{F_K(\cdot)} = 1) - \Pr(R \stackrel{R}{\leftarrow} \text{Rand}(n, n') : A^{R(\cdot)} = 1)|$$

と定義され、 $A^{F_K(\cdot)}$ は質問 X に対し、 $Y = F_K(X)$ を返すオラクル $F_K(\cdot)$ を持つ敵 A を表し、 $A^{R(\cdot)}$ は質問 X に対し、 $Y = R(X)$ を返すオラクル $R(\cdot)$ を持つ敵 A を表す。

計算量理論的安全性 関数族 F の擬似ランダム関数族としての安全性を考える場合に扱う資源は、実行時間 t とオラクルへの質問回数 q である。

$$\text{Adv}_F^{\text{prf}}(t, q) \stackrel{\text{def}}{=} \max_A \left\{ \text{Adv}_E^{\text{prf}}(A) \right\}$$

と定義される。ただし、最大値は実行時間 t 、オラクルへの質問回数 q のすべての敵 A についてとる。

XMACR の安全性 $\text{Adv}_{\text{XMACR}[F,b]}^{\text{s-uf}}(t, q_g, q_v, m)$ を

$$\text{Adv}_{\text{XMACR}[F,b]}^{\text{s-uf}}(t, q_g, q_v, m) \stackrel{\text{def}}{=} \max_A \left\{ \text{Adv}_{\text{XMACR}[F,b]}^{\text{s-uf}}(A) \right\}$$

と定義する。ただし、最大値は実行時間 t 、タグ生成オラクルへ高々 q_g 回、確認オラクルへ高々 q_v 回、それぞれの質問の長さが高々 m ブロックであるすべての敵 A についてとる。

XMACR については、以下の定理が示されている [BGR95]。

定理 8.7. $n, n' \geq 1$ を整数、 b を $b \leq n - 1$ なる整数、 $t, q_g, q_v, m \geq 1$ を整数とする。 $F : \{0, 1\}^k \times \{0, 1\}^n \rightarrow \{0, 1\}^{n'}$ を関数族とする。このとき、

$$\text{Adv}_{\text{XMACR}[F,b]}^{\text{s-uf}}(t, q_g, q_v, m) \leq \text{Adv}_F^{\text{prf}}(t', q') + \frac{2q_s^2}{2^n} + \frac{2q_v^2}{2^{n'}}$$

である。ただし、 $q' = (q_g + q_v) \cdot (m + 1)$ 、 $t' = t + O((n + n')q')$ である。

定理 8.7 は、以下のことを示している： 実行時間 t , タグ生成オラクルに高々 q_g 回, 確認オラクルに高々 q_v 回の質問をし, それぞれの質問が高々 m ブロック (1 ブロックは b ビット) であり,

$$\mathbf{Adv}_{\text{XMACR}[F,b]}^{\text{s-uf}}(A) = \epsilon$$

なる敵 A が存在すると仮定する. このとき, 実行時間 $t' = t + O((n+n')q')$, 高々 $q' = (q_g + q_v) \cdot (m + 1)$ 回の質問をし,

$$\mathbf{Adv}_E^{\text{pp}}(B) \geq \epsilon - \frac{2q_g^2}{2^n} - \frac{2q_v^2}{2^{n'}}$$

なる敵 B が存在する.

XMACC の安全性 $\mathbf{Adv}_{\text{XMACC}[F,b]}^{\text{s-uf}}(t, q_g, q_v, m)$ を

$$\mathbf{Adv}_{\text{XMACC}[F,b]}^{\text{s-uf}}(t, q_g, q_v, m) \stackrel{\text{def}}{=} \max_A \{ \mathbf{Adv}_{\text{XMACC}[F,b]}^{\text{s-uf}}(A) \}$$

と定義する. 最大値のとりかたは, XMACR と同様である.

XMACC については, 以下の定理が示されている [BGR95].

定理 8.8. $n, n' \geq 1$ を整数, b を $b \leq n - 1$ なる整数, $t, q_g, q_v, m \geq 1$ を $q_g \leq 2^{n-1}$ なる整数とする. $F : \{0, 1\}^k \times \{0, 1\}^n \rightarrow \{0, 1\}^{n'}$ を関数族とする. このとき,

$$\mathbf{Adv}_{\text{XMACC}[F,b]}^{\text{s-uf}}(t, q_g, q_v, m) \leq \mathbf{Adv}_F^{\text{prf}}(t', q') + \frac{2q_v^2}{2^{n'}}$$

である. ただし, $q' = (q_g + q_v) \cdot (m + 1)$, $t' = t + O((n + n')q')$ である.

定理 8.8 は、以下のことを示している： 実行時間 t , タグ生成オラクルに高々 q_g 回, 確認オラクルに高々 q_v 回の質問をし, それぞれの質問が高々 m ブロック (1 ブロックは b ビット) であり,

$$\mathbf{Adv}_{\text{XMACC}[F,b]}^{\text{s-uf}}(A) = \epsilon$$

なる敵 A が存在すると仮定する. このとき, 実行時間 $t' = t + O((n+n')q')$, 高々 $q' = (q_g + q_v) \cdot (m + 1)$ 回の質問をし,

$$\mathbf{Adv}_E^{\text{pp}}(B) \geq \epsilon - \frac{2q_v^2}{2^{n'}}$$

なる敵 B が存在する.

XMACC は, 送信者がカウンタを保持しなければならない代わりに, 安全性のバウンドが q_g に依存しない, という利点を持っている.

効率 XMACR, XMACC の効率は、以下のようにまとめられる。

- 鍵長：関数族 F (ブロック暗号) の鍵 $K \in \{0, 1\}^k$ の一つのみである。
- F の鍵スケジューリングの呼び出し回数：1 回である。
- メッセージ M に対するタグを生成するのにかかる F の呼び出し回数：パディングの定義により異なるが、(4) のようにした場合は、 $\lceil (|M|+1)/b \rceil + 1$ 回必要である。 F として、ブロック暗号を用いた場合、CBC MAC では $|M|/n$ なので、CBC MAC と比べ、 n/b 倍程度必要である。たとえば、 $b = n/2$ とした場合、およそ 2 倍の呼び出し回数が必要である。
- 事前計算するべき F の呼び出し回数：XMACC における送信者は $F_K(0||C)$ を計算できる。
- 並列処理性：並列処理可能である。CBC MAC とその変形は、ブロック暗号の並列処理ができない。

標準化状況 標準化された実績はない。

8.8 XECB MAC

3つの方式：乱数を用いる XECB\$-MAC 方式、送信者が状態を用いる XECBC-MAC 方式、状態と乱数を用いる XECBS-MAC 方式が提案されている。

どの方式もブロック暗号 $E: \mathcal{K}_E \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ を用いる。

方式 (XECB\$-MAC) XECB\$-MAC は乱数を用いる方式である。XECB\$-MAC は、ブロック暗号 E をパラメータとする。これを XECB\$-MAC[E] と表記する。

$$\text{XECB\$-MAC}[E] = (\text{XECB\$-}\mathcal{K}, \text{XECB\$-}\mathcal{G}, \text{XECB\$-}\mathcal{V})$$

の鍵生成アルゴリズム XECB\$- \mathcal{K} 、タグ生成アルゴリズム XECB\$- \mathcal{G} 、確認アルゴリズム XECB\$- \mathcal{V} はそれぞれ以下のように動作する。

- 鍵生成アルゴリズム XECB\$- \mathcal{K} は確率的アルゴリズムであり、

$$K \stackrel{R}{\leftarrow} \mathcal{K}_E$$

を出力する。

- タグ生成アルゴリズム $\text{XECB}\$-\mathcal{G} : \mathcal{K}_E \times \{0, 1\}^* \rightarrow (\{0, 1\}^n \times \{0, 1\}^n)$ は確率的アルゴリズムであり，鍵空間は \mathcal{K}_E ，メッセージ空間は $\{0, 1\}^*$ ，タグ空間は $(\{0, 1\}^n \times \{0, 1\}^n)$ である．すなわち，鍵 $K \in \mathcal{K}_E$ とメッセージ $M \in \{0, 1\}^*$ を入力とし，タグ

$$T = \text{XECB}\$-\mathcal{G}_K(M) \in (\{0, 1\}^n \times \{0, 1\}^n)$$

を出力する．図 59 にあるように動作する．

```

Algorithm  $\text{XECB}\$-\mathcal{G}_K(M)$ 
 $r_0 \xleftarrow{R} \{0, 1\}^n$ 
 $y_0 \leftarrow E_K(r_0)$ 
 $z_0 \leftarrow E_K(r_0 + 1)$ 
Partition  $M$  into  $M[1] \cdots M[m]$ 
if  $|M[m]| = n$  then  $Z \leftarrow \bar{z}_0$ 
           else  $Z \leftarrow z_0$ 
 $M[m] \leftarrow \text{pad}_n(M[m])$ 
 $M[m+1] \leftarrow Z$ 
for  $i \leftarrow 1$  to  $m+1$  do
            $X[i] \leftarrow M[i] \oplus i \times y_0$ 
            $Y[i] \leftarrow E_K(X[i])$ 
 $T' \leftarrow Y[1] \oplus \cdots \oplus Y[m+1]$ 
 $T \leftarrow (r_0, T')$ 
return  $T$ 

```

図 59: XECB\$-MAC のタグ生成アルゴリズム $\text{XECB}\$-\mathcal{G}_K(\cdot)$.

ただし， $+$ や \times の演算は $\text{mod } 2^n$ 上で行われる．2 行目の r_0 は n ビットの乱数である．6 行目の \bar{z}_0 は z_0 のビットごとの反転である．8 行目の $\text{pad}_n(\cdot)$ は (1) で定義されている．

- 確認アルゴリズム $\text{XECB}\$-\mathcal{V} : \mathcal{K}_E \times \{0, 1\}^* \times (\{0, 1\}^n \times \{0, 1\}^n) \rightarrow \text{accept or reject}$ は決定的アルゴリズムであり，鍵 $K \in \mathcal{K}_E$ ，メッセージ $M \in \{0, 1\}^*$ ，タグ $T \in (\{0, 1\}^n \times \{0, 1\}^n)$ を入力とし，

$$\text{accept or reject} = \text{XECB}\$-\mathcal{V}_K(M, T)$$

を出力する．図 60 にあるように動作する．

r_0 を用いてタグ生成を行い，一致していれば accept を返す．

```

Algorithm XECB$\mathcal{V}_K(M, (r_0, T'))
y_0 \leftarrow E_K(r_0)
z_0 \leftarrow E_K(r_0 + 1)
Partition $M$ into $M[1] \cdots M[m]$
if $|M[m]| = n$ then $Z \leftarrow \bar{z}_0$
    else $Z \leftarrow z_0$
$M[m] \leftarrow \text{pad}_n(M[m])$
$M[m+1] \leftarrow Z$
for $i \leftarrow 1$ to $m+1$ do
    $X[i] \leftarrow M[i] \oplus i \times y_0$
    $Y[i] \leftarrow E_K(X[i])$
$T'' \leftarrow Y[1] \oplus \cdots \oplus Y[m+1]$
$T''' \leftarrow (r_0, T')$
if $T' = T'''$ then return accept
    else return reject

```

図 60: XECB\$-MAC の確認アルゴリズム XECB\$\mathcal{V}_K(\cdot, \cdot)\$.

方式 (XECBC-MAC) XECBC-MAC は, 送信者が状態を用いる方式である. XECB\$-MAC と同様, ブロック暗号 E をパラメータとする. これを, XECBC-MAC[E] と表記する.

$$\text{XECBC-MAC}[E] = (\text{XECBC-}\mathcal{K}, \text{XECBC-}\mathcal{G}, \text{XECBC-}\mathcal{V})$$

の鍵生成アルゴリズム XECBC- \mathcal{K} , タグ生成アルゴリズム XECBC- \mathcal{G} , 確認アルゴリズム XECBC- \mathcal{V} はそれぞれ以下のように動作する.

- 鍵生成アルゴリズム XECBC- \mathcal{K} は確率的アルゴリズムであり,

$$K \stackrel{R}{\leftarrow} \mathcal{K}_E$$

を出力する.

- タグ生成アルゴリズム XECBC- $\mathcal{G} : \mathcal{K}_E \times \{0, 1\}^* \times \text{Count} \rightarrow (\text{Count} \times \{0, 1\}^n)$ は決定的アルゴリズムであり, 鍵空間は \mathcal{K}_E , メッセージ空間は $\{0, 1\}^*$, タグ空間は $(\text{Count} \times \{0, 1\}^n)$ である. さらに入力として, カウンタ $C \in \text{Count}$ をとる. Count はカウンタの空間であり, $\text{Count} = \{1, 2, \dots\}$ である. カウンタは送信者により保持され

ており, 1 に初期化され, タグ生成アルゴリズムによって更新される. 受信者はこれを保持しない. すなわち, 鍵 $K \in \mathcal{K}_E$, メッセージ $M \in \{0, 1\}^*$, カウンタ $C \in \text{Count}$ を入力とし, タグ

$$T = \text{XECBC-}\mathcal{G}_K(M) \in (\text{Count} \times \{0, 1\}^n)$$

を出力する. 図 61 にあるように動作する.

```

Algorithm XECBC- $\mathcal{G}_K(C, M)$ 
 $y_0 \leftarrow E_K(C)$ 
 $z_0 \leftarrow E_K(y_0)$ 
Partition  $M$  into  $M[1] \cdots M[m]$ 
if  $|M[m]| = n$  then  $Z \leftarrow \bar{z}_0$ 
      else  $Z \leftarrow z_0$ 
 $M[m] \leftarrow \text{pad}_n(M[m])$ 
 $M[m+1] \leftarrow Z$ 
for  $i \leftarrow 1$  to  $m+1$  do
       $X[i] \leftarrow M[i] \oplus i \times y_0$ 
       $Y[i] \leftarrow E_K(X[i])$ 
 $T' \leftarrow Y[1] \oplus \cdots \oplus Y[m+1]$ 
 $C' \leftarrow C + 1$ 
 $T \leftarrow (C, T')$ 
return  $T$ 

```

図 61: XECBC-MAC のタグ生成アルゴリズム XECBC- $\mathcal{G}_K(\cdot)$.

ただし, C' は更新されたカウンタの値であり, $+$ と \times の演算は, $\text{mod } 2^n$ 上で行われる.

- 確認アルゴリズム XECBC- $\mathcal{V}: \mathcal{K}_E \times \{0, 1\}^* \times (\text{Count} \times \{0, 1\}^n) \rightarrow \text{accept or reject}$ は決定的アルゴリズムであり, 鍵 $K \in \mathcal{K}_E$, メッセージ $M \in \{0, 1\}^*$, タグ $T \in (\text{Count} \times \{0, 1\}^n)$ を入力とし,

$$\text{accept or reject} = \text{XMACC-}\mathcal{V}_K(M, T)$$

を出力する. 図 62 にあるように動作する.

```

Algorithm XMACC- $\mathcal{V}_K(M, (C, T'))$ 
 $y_0 \leftarrow E_K(C)$ 
 $z_0 \leftarrow E_K(y_0)$ 
Partition  $M$  into  $M[1] \cdots M[m]$ 
if  $|M[m]| = n$  then  $Z \leftarrow \bar{z}_0$ 
           else  $Z \leftarrow z_0$ 
 $M[m] \leftarrow \text{pad}_n(M[m])$ 
 $M[m+1] \leftarrow Z$ 
for  $i \leftarrow 1$  to  $m+1$  do
            $X[i] \leftarrow M[i] \oplus i \times y_0$ 
            $Y[i] \leftarrow E_K(X[i])$ 
 $T'' \leftarrow Y[1] \oplus \cdots \oplus Y[m+1]$ 
if  $T' = T''$  then return accept
           else return reject

```

図 62: XECBC の確認アルゴリズム XECBC- $\mathcal{V}_K(\cdot, \cdot)$.

方式 (XECBS-MAC) XECBS-MAC は状態と乱数を用いる方式である。これも、ブロック暗号 E をパラメータとする。XECBS-MAC[E] と表記する。XECBS-MAC[E] = (XECBS- \mathcal{K} , XECBS- \mathcal{G} , XECBS- \mathcal{V}) の鍵生成アルゴリズム XECBS- \mathcal{K} , タグ生成アルゴリズム XECBS- \mathcal{G} , 確認アルゴリズム XECBS- \mathcal{V} はそれぞれ以下のように動作する。

- 鍵生成アルゴリズム XECBS- \mathcal{K} は確率的アルゴリズムであり,

$$K \stackrel{R}{\leftarrow} \mathcal{K}_E$$

を出力する。

- タグ生成アルゴリズム XECBS- $\mathcal{G} : \mathcal{K}_E \times \{0, 1\}^* \times \text{Count} \rightarrow (\text{Count} \times \{0, 1\}^n)$ は決定的アルゴリズムであり、鍵空間は \mathcal{K}_E , メッセージ空間は $\{0, 1\}^*$, タグ空間は $(\text{Count} \times \{0, 1\}^n)$ である。さらに入力として、カウンタ $C \in \text{Count}$ をとる。Count はカウンタの空間であり、 $\text{Count} = \{1, 2, \dots\}$ である。カウンタは送信者により保持されており、1 に初期化され、タグ生成アルゴリズムによって更新される。受信者はこれを保持しない。すなわち、鍵 $K \in \mathcal{K}_E$, メッセージ $M \in \{0, 1\}^*$, カウンタ $C \in \text{Count}$ を入力とし、タグ

$$T = \text{XECBS-}\mathcal{G}_K(M) \in (\text{Count} \times \{0, 1\}^n)$$

を出力する。さらにアルゴリズム内部では、状態 R と R^* を用いる。これらは、 n ビットのビット列であり、鍵ごとに更新される。メッセージごとに更新されるわけではない。この状態は送信者だけでなく、受信者も保持している。秘密鍵 K から導出してもよいと記されている [GD01a]。図 63 にあるように動作する。

```

Algorithm XECBS- $\mathcal{G}_K(C, M)$ 
Partition  $M$  into  $M[1] \cdots M[m]$ 
if  $|M[m]| = n$  then  $Z \leftarrow \bar{R}$ 
      else  $Z \leftarrow R$ 
 $M[m] \leftarrow \text{pad}_n(M[m])$ 
 $M[m+1] \leftarrow Z$ 
for  $i \leftarrow 1$  to  $m$  do
       $X[i] \leftarrow M[i] \oplus C \times Z \oplus i \times R^*$ 
       $Y[i] \leftarrow E_K(X[i])$ 
 $T' \leftarrow Y[1] \oplus \cdots \oplus Y[m]$ 
 $C' \leftarrow C + 1$ 
 $T \leftarrow (C, T')$ 
return  $T$ 

```

図 63: XECBS-MAC のタグ生成アルゴリズム XECBS- $\mathcal{G}_K(\cdot)$.

ただし、 C' は更新されたカウンタの値であり、 $+$ と \times の演算は、 $\text{mod } 2^n$ 上で行われる。

- 確認アルゴリズム XECBS- $\mathcal{V} : \mathcal{K}_E \times \{0, 1\}^* \times (\text{Count} \times \{0, 1\}^n) \rightarrow \text{accept or reject}$ は決定的アルゴリズムであり、鍵 $K \in \mathcal{K}_E$ 、メッセージ $M \in \{0, 1\}^*$ 、タグ $T \in (\text{Count} \times \{0, 1\}^n)$ を入力とし、

$$\text{accept or reject} = \text{XMACC-}\mathcal{V}_K(M, T)$$

を出力する。さらにアルゴリズム内部では、状態 R と R^* を用いる。これらは、 n ビットのビット列であり、鍵ごとに更新される。メッセージごとに更新されるわけではない。この状態は、受信者も保持している。図 64 にあるように動作する。

安全性 Gligor, Donescu により、安全性が解析されている [GD01a]。ブロック暗号 E が安全な擬似ランダム置換族であれば、XECBS-MAC[E],

```

Algorithm XMACC- $\mathcal{V}_K(M, (C, T'))$ 
Partition  $M$  into  $M[1] \cdots M[m]$ 
if  $|M[m]| = n$  then  $Z \leftarrow \bar{R}$ 
           else  $Z \leftarrow R$ 
 $M[m] \leftarrow \text{pad}_n(M[m])$ 
 $M[m+1] \leftarrow Z$ 
for  $i \leftarrow 1$  to  $m$  do
     $X[i] \leftarrow M[i] \oplus C \times Z \oplus i \times R^*$ 
     $Y[i] \leftarrow E_K(X[i])$ 
 $T'' \leftarrow Y[1] \oplus \cdots \oplus Y[m]$ 
if  $T' = T''$  then return accept
           else return reject

```

図 64: XECBS の確認アルゴリズム XECBS- $\mathcal{V}_K(\cdot, \cdot)$.

XECBC-MAC[E], XECBS-MAC[E] は, いずれも, 強偽造不可能性の意味で安全な MAC であることが示されている.

XECBS-MAC の安全性 $\text{Adv}_{\text{XECBS-MAC}[E]}^{\text{s-uf}}(t, q_g, \mu_g, q_v, \mu_v)$ を

$$\text{Adv}_{\text{XECBS-MAC}[E]}^{\text{s-uf}}(t, q_g, \mu_g, q_v, \mu_v) \stackrel{\text{def}}{=} \max_A \{ \text{Adv}_{\text{XECBS-MAC}[E]}^{\text{s-uf}}(A) \}$$

と定義する. ただし, 最大値は実行時間 t , タグ生成オラクルへ高々 q_g 回の質問を合計で高々 μ_g ブロック, 確認オラクルへ高々 q_v 回の質問を合計で高々 μ_v ブロックであるすべての敵 A についてとる.

XECBS-MAC については, 以下の定理が示されている [GD01a].

定理 8.9. $n \geq 1$ を整数, $t, q_g, \mu_g, q_v, \mu_v \geq 1$ を整数とする. $E : \mathcal{K}_E \times \{0, 1\}^n \rightarrow \{0, 1\}^{n'}$ をブロック暗号とする. このとき,

$$\begin{aligned} & \text{Adv}_{\text{XECBS-MAC}[E]}^{\text{s-uf}}(t, q_g, \mu_g, q_v, \mu_v) \\ & \leq \text{Adv}_E^{\text{prf}}(t', q') + \frac{\mu_v((\log \mu_v) + 3)}{2^n} + \frac{q_g \mu_v^2}{2^n} \\ & \quad + \left(q_g + 2q_v + \frac{\mu_g}{2} \right) \frac{\mu_g((\log \mu_v) + 3)}{2^{n+1}} \end{aligned}$$

である. ただし, $\mu_s + \mu_g \leq q'$, $t \leq t'$ である.

バウンドは最後の項が最も大きく, ほかの MAC と比べると, 通常の birthday paradox バウンドよりも \log スケールで悪いことがわかる.

XECBS-MAC の安全性 $\text{Adv}_{\text{XECBS-MAC}[E]}^{\text{s-uf}}(t, q_g, \mu_g, q_v, \mu_v)$ も $\text{XECBS-MAC}[E]$ と同様に定義する.

XECBS-MAC については, 以下の定理が示されている [GD01a].

定理 8.10. $n \geq 1$ を整数, $t, q_g, \mu_g, q_v, \mu_v \geq 1$ を整数とする. $E : \mathcal{K}_E \times \{0, 1\}^n \rightarrow \{0, 1\}^{n'}$ をブロック暗号とする. このとき,

$$\begin{aligned} & \text{Adv}_{\text{XECBS-MAC}[E]}^{\text{s-uf}}(t, q_g, \mu_g, q_v, \mu_v) \\ & \leq \text{Adv}_E^{\text{prf}}(t', q') + \frac{q_v}{2^n} + \frac{\mu_v((\log \mu_v) + 3)}{2^{n+1}} \\ & \quad + (q_v + \mu_g) \frac{q_s((\log q_s) + 3)}{2^{n+1}} + (q_v + \mu_g) \frac{\mu_g((\log \mu_v) + 3)}{2^{n+1}} \end{aligned}$$

である. ただし, $\mu_s + \mu_g \leq q', t \leq t'$ である.

XECBC-MAC の安全性 $\text{Adv}_{\text{XECBC-MAC}[E]}^{\text{s-uf}}(t, q_g, \mu_g, q_v, \mu_v)$ も上記二つと同様に定義する.

XECBC-MAC については, 以下の定理が示されている [GD01a].

定理 8.11. $n \geq 1$ を整数, $t, q_g, \mu_g, q_v, \mu_v \geq 1$ を整数とする. $E : \mathcal{K}_E \times \{0, 1\}^n \rightarrow \{0, 1\}^{n'}$ をブロック暗号とする. このとき,

$$\begin{aligned} & \text{Adv}_{\text{XECBC-MAC}[E]}^{\text{s-uf}}(t, q_g, \mu_g, q_v, \mu_v) \\ & \leq \text{Adv}_E^{\text{prf}}(t', q') + \frac{\mu_v^2}{2^{n+1}} + \frac{q_v}{2^n} + \frac{\mu_v((\log \mu_v) + 3)}{2^{n+1}} \\ & \quad + (q_v + \mu_g) \frac{q_g((\log q_g) + 3)}{2^{n+1}} + (q_v + \mu_g) \frac{\mu_g((\log \mu_v) + 3)}{2^{n+1}} + \frac{\mu_g^2}{2^{n+1}} \end{aligned}$$

である. ただし, $\mu_s + \mu_g \leq q', t \leq t'$ である.

効率 XECB MAC の効率は, 以下のようによまとめられる.

- 鍵長: ブロック暗号 E の鍵 $K \in \mathcal{K}_E$ の一つのみである.
- E の鍵スケジューリングの呼び出し回数: 1 回である.
- メッセージ M に対するタグを生成するのにかかる F の呼び出し回数: XECBS-MAC と XECBC-MAC では, $\lceil |M|/n \rceil + 3$ 回, XECBS-MAC では, $\lceil |M|/n \rceil$ 回必要である. しかし, XECBS-MAC では, R と R^* の生成にブロック暗号を呼び出す必要がある場合がある.

- 事前計算すべき F の呼び出し回数：XECBS-MAC では、 R と R^* の生成にブロック暗号を呼び出す必要がある場合がある。
- 並列処理性：並列処理可能である。

標準化状況 NIST に提案されている [WWW9].

8.9 PMAC

方式 PMAC はブロック暗号 E とタグ長 τ をパラメータとする。ブロック長 n のブロック暗号 $E: \mathcal{K}_E \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ を用いた場合は、 $\tau \leq n$ でなくてはならない。これらのパラメータを用いた PMAC を $\text{PMAC}[E, \tau]$ と表記する。 $\text{PMAC}[E, \tau] = (\text{PMAC-}\mathcal{K}, \text{PMAC-}\mathcal{G}, \text{PMAC-}\mathcal{V})$ の鍵生成アルゴリズム $\text{PMAC-}\mathcal{K}$ 、タグ生成アルゴリズム $\text{PMAC-}\mathcal{G}$ 、確認アルゴリズム $\text{PMAC-}\mathcal{V}$ はそれぞれ以下のように動作する。

- 鍵生成アルゴリズム $\text{PMAC-}\mathcal{K}$ は確率的アルゴリズムであり、

$$K \xleftarrow{R} \mathcal{K}_E$$

を出力する。

- タグ生成アルゴリズム $\text{PMAC-}\mathcal{G}: \mathcal{K}_E \times \{0, 1\}^* \rightarrow \{0, 1\}^\tau$ は決定的アルゴリズムであり、鍵空間は \mathcal{K}_E 、メッセージ空間は $\{0, 1\}^*$ 、タグ空間は $\{0, 1\}^\tau$ である。すなわち、鍵 $K \in \mathcal{K}_E$ とメッセージ $M \in \{0, 1\}^*$ を入力とし、タグ

$$T = \text{PMAC-}\mathcal{G}_K(M) \in \{0, 1\}^\tau$$

を出力する。図 65, 図 66 にあるように動作する。PMAC は M の長さが n の倍数でなくてもよい。図 65 の 3 行目において、

$$M = M[1]M[2] \cdots M[m-1]M[m]$$

は、 $|M[1]| = |M[2]| = \cdots = |M[m-1]|$ かつ $1 \leq |M[m]| \leq n$ となるように分割される。 $M = \varepsilon$ のときは例外である。この場合、 $|M[m]| = 0$ となる。

5 行目の $\gamma_i \cdot L$ は、以下のように計算される。

$$\begin{cases} \gamma_1 \cdot L = L \\ \gamma_i \cdot L = (\gamma_{i-1} \cdot L) \oplus (L \cdot \mathbf{u}^{\text{ntz}(i)}) \quad (i \geq 2) \end{cases}$$

```

Algorithm PMAC- $\mathcal{G}_K(M)$ 
 $L \leftarrow E_K(0^n)$ 
Partition  $M$  into  $M[1] \cdots M[m]$ 
for  $i \leftarrow 1$  to  $m - 1$  do
     $X[i] \leftarrow M[i] \oplus \gamma_i \cdot L$ 
     $Y[i] \leftarrow E_K(X[i])$ 
 $\Sigma \leftarrow Y[1] \oplus Y[2] \oplus \cdots \oplus Y[m - 1] \oplus \mathbf{pad}_n(M[m])$ 
if  $|M[m]| = n$  then  $X[m] \leftarrow \Sigma \oplus L \cdot \mathbf{u}^{-1}$ 
    else  $X[m] \leftarrow \Sigma$ 
 $T \leftarrow$  the left most  $\tau$  bits of  $E_K(X[m])$ 
return  $T$ 

```

図 65: PMAC のタグ生成アルゴリズム PMAC- $\mathcal{G}_K(\cdot)$.

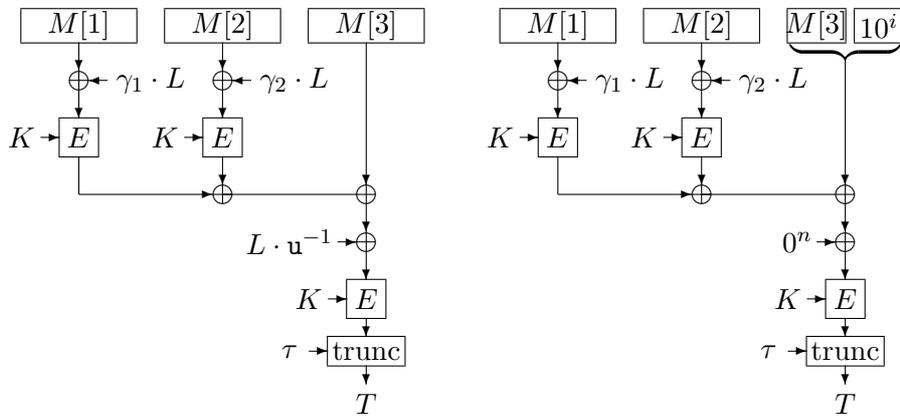


図 66: $M = M[1]M[2]M[3]$ の場合の PMAC- $\mathcal{G}_K(M)$ の動作.

ここで、 $\text{ntz}(i)$ は、 i をビット表現したときの、最下位ビットから連続して並ぶ 0 の個数である。もしくは、 $\text{ntz}(i)$ は、 2^z が i を割り切る最大の z である。たとえば、 $\text{ntz}(7) = 0$ 、 $\text{ntz}(8) = 3$ である。 $\gamma_i \cdot L$ は事前計算でいくつか計算しておいてもよいし、必要に応じて計算してもよい。

また、図 65 の 7 行目の関数 $\text{pad}_n : \{0, 1\}^{\leq n} \rightarrow \{0, 1\}^n$ は (1) と同様である。8 行目の $L \cdot u^{-1}$ は、(3) のように計算される。

- 確認アルゴリズム $\text{PMAC-}\mathcal{V} : \mathcal{K}_E \times \{0, 1\}^* \times \{0, 1\}^\tau \rightarrow \text{accept or reject}$ は決定的アルゴリズムであり、鍵 $K \in \mathcal{K}_E$ 、メッセージ $M \in \{0, 1\}^*$ 、タグ $T \in \{0, 1\}^\tau$ を入力とし、

$$\text{accept or reject} = \text{PMAC-}\mathcal{V}_K(M, T)$$

を出力する。図 67 にあるように動作する。

Algorithm $\text{PMAC-}\mathcal{V}_K(M, T)$
 $T' \leftarrow \text{PMAC-}\mathcal{G}_K(M)$
if $T = T'$ **then return** accept
else return reject

図 67: PMAC の確認アルゴリズム $\text{PMAC-}\mathcal{V}_K(\cdot, \cdot)$.

安全性 Black, Rogaway により、安全性が解析されている [BR02]。ブロック暗号 E が安全な擬似ランダム置換族であれば、 $\text{PMAC}[E, \tau]$ は、偽造不可能性の意味で安全な MAC であることが示されている。具体的には、以下の定理が示されている [BR02]。

定理 8.12. $n, \tau \geq 1$ を整数、 $t, q, \sigma \geq 1$ を整数とする。 $E : \mathcal{K}_E \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ をブロック暗号とする。このとき、

$$\text{Adv}_{\text{PMAC}[E, \tau]}^{\text{mac}}(t, q, \sigma) \leq \text{Adv}_E^{\text{prp}}(t', q') + \frac{1.5\sigma'^2}{2^n} + \frac{1}{2^\tau}$$

である。ただし、 $\sigma' = \sigma + q + 1$ 、 $q' = \sigma + 1$ 、 $t' = t + O(n\sigma)$ であり、質問の長さはブロック単位である。

定理 8.12 は、以下のことを示している： 実行時間 t , 高々 q 回の質問をし、それらの質問が合計で高々 σ ブロックであり、

$$\text{Adv}_{\text{PMAC}[E,\tau]}^{\text{mac}}(A) = \epsilon$$

なる敵 A が存在すると仮定する。 $\sigma' = \sigma + q + 1$ とする。 このとき、実行時間 $t' = t + O(n\sigma)$, 高々 $q' = \sigma$ 回の質問をし、

$$\text{Adv}_E^{\text{PRP}}(B) \geq \epsilon - \frac{1.5\sigma'^2}{2^n} - \frac{1}{2^\tau}$$

なる敵 B が存在する。

効率 PMAC の効率は、以下のようにまとめられる。

- 鍵長：ブロック暗号の鍵 $K \in \mathcal{K}_E$ の計 1 つが必要である。
- ブロック暗号鍵スケジューリングの呼び出し回数：1 回である。
- メッセージ M に対するタグを生成するのにかかるブロック暗号の呼び出し回数： $\max\{1, \lceil |M|/n \rceil\}$ 回の呼び出しである。
- 事前計算するべきブロック暗号の呼び出し回数：1 回である。
- 並列処理性：並列処理可能である。

標準化状況 NIST に提案されている [WWW9].

8.10 f_9 (3GPP)

方式 f_9 はブロック暗号 KASUMI: $\{0, 1\}^{128} \times \{0, 1\}^{64} \rightarrow \{0, 1\}^{64}$ を用いる。 鍵生成アルゴリズム $f_9\text{-K}$, タグ生成アルゴリズム $f_9\text{-G}$, 確認アルゴリズム $f_9\text{-V}$ はそれぞれ以下のように動作する。

- 鍵生成アルゴリズム $f_9\text{-K}$ は確率的アルゴリズムであり、

$$K \stackrel{R}{\leftarrow} \{0, 1\}^{128}$$

を出力する。

- タグ生成アルゴリズム $f9-G : \mathcal{K}_E \times \{0,1\}^* \rightarrow \{0,1\}^\tau$ は決定的アルゴリズムであり，鍵空間は $\{0,1\}^{64}$ ，メッセージ空間は $\{0,1\}^*$ ，タグ空間は $\{0,1\}^{32}$ である．さらに，32 ビットのカウンタ COUNT，32 ビットの乱数 FRESH，1 ビットの direction identifier DIRECTION を入力にもつ．これらは，送受信者間で共有されている．図 68，図 69 にあるように動作する．

```

Algorithm  $f9-G_K(\text{COUNT}, \text{FRESH}, \text{DIRECTION}, M)$ 
 $M \leftarrow \text{pad}_{64}(\text{COUNT}, \text{FRESH}, \text{DIRECTION}, M)$ 
Break  $M$  into 64-bit blocks  $M[1] \parallel \dots \parallel M[m]$ 
 $Y[0] \leftarrow 0^{64}$ 
For  $i = 1$  to  $m$  do:
     $X[i] \leftarrow M[i] \oplus Y[i-1]$ 
     $Y[i] \leftarrow \text{KASUMI}_K(X[i])$ 
 $T \leftarrow \text{KASUMI}_{K \oplus KM}(Y[1] \oplus \dots \oplus Y[m])$ 
 $T \leftarrow$  the leftmost 32 bits of  $T$ 
Return  $T$ 

```

図 68: $f9$ のタグ生成アルゴリズム $f9-G_K(\cdot)$.

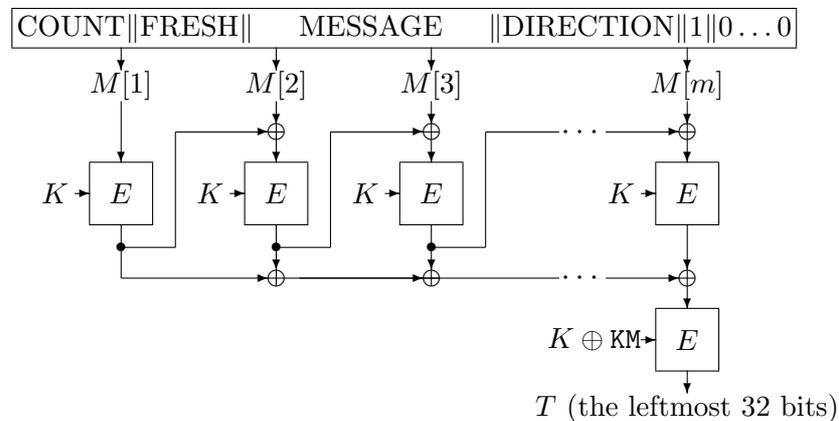


図 69: $f9-G_K(M)$ の動作. E は KASUMI である.

2 行目の $\text{pad}_{64}(\text{COUNT}, \text{FRESH}, \text{DIRECTION}, M)$ は以下のように動作する：まず，COUNT, FRESH, M, DIRECTION を連結し，次に 1 ビットの “1” を連結し，最後に，全体の長さが 64 ビットの

整数倍になるように，“0”を連結する。すなわち，

$$\begin{aligned} \text{pad}_{64}(\text{COUNT}, \text{FRESH}, \text{DIRECTION}, M) \\ = \text{COUNT} \parallel \text{FRESH} \parallel M \parallel \text{DIRECTION} \parallel 1 \parallel 0^{63 - (|M| + 1 \bmod 64)} \end{aligned}$$

とする。KM は，128 ビットの定数であり， $\text{KM} = 0x\text{AA}\dots\text{AA}$ である。

- 確認アルゴリズム $f9\text{-}\mathcal{V} : \{0, 1\}^{64} \times \{0, 1\}^* \times \{0, 1\}^{32} \rightarrow \text{accept or reject}$ は決定的アルゴリズムであり，鍵 $K \in \{0, 1\}^{64}$ ，メッセージ $M \in \{0, 1\}^*$ ，タグ $T \in \{0, 1\}^{32}$ を入力とし，

$$\text{accept or reject} = f9\text{-}\mathcal{V}_K(M, T)$$

を出力する。さらに，送受信者間で共有されている 32 ビットのカウンタ COUNT，32 ビットの乱数 FRESH，1 ビットの direction identifier DIRECTION を入力にもつ。

図 70 にあるように動作する。

Algorithm $f9\text{-}\mathcal{V}_K(\text{COUNT}, \text{FRESH}, \text{DIRECTION}, M, T)$
 $T' \leftarrow f9\text{-}\mathcal{G}_K(\text{COUNT}, \text{FRESH}, \text{DIRECTION}, M)$
if $T = T'$ **then return** accept
else return reject

図 70: $f9$ の確認アルゴリズム $f9\text{-}\mathcal{V}_K(\cdot, \cdot)$.

安全性 Hong, Kang, Preneel, Ryu により，KASUMI が安全な擬似ランダム置換であれば， $f9$ が偽造不能性の意味で安全な MAC であることが主張された [HKPR03] が，証明の不備が指摘されている [IK03c].

効率 $f9$ の効率は，以下のようにまとめられる。

- 鍵長：KASUMI の鍵 $K \in \{0, 1\}^{64}$ 一つのみである。
- ブロック暗号鍵スケジューリングの呼び出し回数： K と $K \oplus \text{KM}$ の 2 回である。
- メッセージ M に対するタグを生成するのにかかるブロック暗号の呼び出し回数： $\lceil |M|/n \rceil + 2$ 回の呼び出しである。
- 事前計算するべきブロック暗号の呼び出し回数：必要ない。
- 並列処理性：並列処理はできない。

標準化状況 3GPP により標準化されている [3GPPa, 3GPPb].

8.11 NMAC, HMAC

8.11.1 NMAC

方式 NMAC (Nested MAC) は反復型ハッシュ関数 F とタグ長 τ をパラメータとする. 出力長 n の反復型ハッシュ関数 $F : \{0, 1\}^* \rightarrow \{0, 1\}^n$ を用いた場合は, $\tau \leq n$ でなければならない. これらのパラメータを用いた NMAC を $\text{NMAC}[F, \tau]$ と表記する. $\text{NMAC}[F, \tau] = (\text{NMAC-}\mathcal{K}, \text{NMAC-}\mathcal{G}, \text{NMAC-}\mathcal{V})$ の鍵生成アルゴリズム NMAC- \mathcal{K} , タグ生成アルゴリズム NMAC- \mathcal{G} , 確認アルゴリズム NMAC- \mathcal{V} はそれぞれ以下のように動作する.

- 鍵生成アルゴリズム NMAC- \mathcal{K} は確率的アルゴリズムであり,

$$K_1 \stackrel{R}{\leftarrow} \{0, 1\}^n$$

と

$$K_2 \stackrel{R}{\leftarrow} \{0, 1\}^n$$

を出力する.

- タグ生成アルゴリズム $\text{NMAC-}\mathcal{G} : (\{0, 1\}^n \times \{0, 1\}^n) \times \{0, 1\}^* \rightarrow \{0, 1\}^\tau$ は決定的アルゴリズムであり, 鍵空間は $\{0, 1\}^n \times \{0, 1\}^n$, メッセージ空間は $\{0, 1\}^*$, タグ空間は $\{0, 1\}^\tau$ である. すなわち, 鍵 $K_1 \in \{0, 1\}^n$, $K_2 \in \{0, 1\}^n$ とメッセージ $M \in \{0, 1\}^*$ を入力とし, タグ

$$T = \text{NMAC-}\mathcal{G}_{K_1, K_2}(M) \in \{0, 1\}^\tau$$

を出力する. 図 71 にあるように動作する. なお, 図 71 で, F_{K_1}, F_{K_2} はそれぞれ反復型ハッシュ関数 F の初期値を K_1, K_2 として計算を行うことを表す.

Algorithm $\text{NMAC-}\mathcal{G}_{K_1, K_2}(M)$
 $T \leftarrow$ the left most τ bits of $F_{K_1}(F_{K_2}(M))$
 Return T

図 71: NMAC のタグ生成アルゴリズム $\text{NMAC-}\mathcal{G}_{K_1, K_2}(\cdot)$.

- 確認アルゴリズム $\text{NMAC-}\mathcal{V} : (\{0, 1\}^n \times \{0, 1\}^n) \times \{0, 1\}^* \times \{0, 1\}^\tau \rightarrow \text{accept or reject}$ は決定的アルゴリズムであり、鍵 $K_1 \in \{0, 1\}^n, K_2 \in \{0, 1\}^n$, メッセージ $M \in \{0, 1\}^*$, タグ $T \in \{0, 1\}^\tau$ を入力とし,

$$\text{accept or reject} = \text{NMAC-}\mathcal{V}_{K_1, K_2}(M, T)$$

を出力する。図 72 にあるように動作する。

Algorithm $\text{NMAC-}\mathcal{V}_{K_1, K_2}(M, T)$
 $T' \leftarrow \text{NMAC-}\mathcal{G}_{K_1, K_2}(M)$
if $T = T'$ **then return** accept
else return reject

図 72: NMAC の確認アルゴリズム $\text{NMAC-}\mathcal{V}_{K_1, K_2}(\cdot, \cdot)$.

安全性 Bellare, Canetti, Krawczyk により安全性が解析されている [BCK96].

まず準備として弱衝突計算困難性 (weak collision resistance) の定義を述べる。関数 $G : \mathcal{K} \times \mathcal{D} \rightarrow \mathcal{R}$ について、 \mathcal{K} を鍵の集合、 \mathcal{D}, \mathcal{R} をそれぞれ G の定義域、値域とする。

敵 A として、 G を計算するオラクルにアクセスできるアルゴリズムを考える。 $A^{G(K, \cdot)}$ は $x \in \mathcal{D}$ に対して $G(K, x)$ を返すオラクル $G(K, \cdot)$ をもつ敵を表す。質問は適応的に行う。すなわち、ある質問に対する答を得た後、次の質問を行う。

$A^{G(K, \cdot)}$ が $G(K, x) = G(K, x')$ かつ $x \neq x'$ を満たす x, x' を出力したとき、 A は弱衝突計算困難性の意味で関数 G の衝突計算に成功したという。 $G : \mathcal{K} \times \mathcal{D} \rightarrow \mathcal{R}$ の敵 A に対する弱衝突計算困難性の意味での安全性はアドバンテージ $\text{Adv}_G^{\text{wcr}}(A)$ によって評価される。ここで、

$$\text{Adv}_G^{\text{wcr}}(A) =$$

$$\Pr[K \xleftarrow{R} \mathcal{K} : A^{G(K, \cdot)} \text{ が弱衝突計算困難性の意味で衝突計算に成功}]$$

と定義される。

$G : \mathcal{K} \times \mathcal{D} \rightarrow \mathcal{R}$ の弱衝突計算困難性の意味での安全性を考える場合に扱う資源は、実行時間 t , オラクルへの質問回数 q , それらの質問の長さ σ である。

$$\text{Adv}_G^{\text{wcr}}(t, q, \sigma) \stackrel{\text{def}}{=} \max_A \{\text{Adv}_G^{\text{wcr}}(A)\}$$

と定義される。ただし、実行時間 t 、オラクルへの質問回数 q 、それらの質問の長さ σ のすべての敵 A について最大値をとる。

[BCK96]における NMAC の安全性についての議論より、以下の定理が成立することが分かる。

定理 8.13.

$$\mathbf{Adv}_{\text{NMAC}[F,\tau]}^{\text{mac}}(t, q, \sigma) \leq \mathbf{Adv}_{f^{(\tau)}}^{\text{mac}}(t + O(q\sigma), q, b) + \mathbf{Adv}_F^{\text{wcr}}(t, q, \sigma)$$

ここで $f^{(\tau)}$ は圧縮関数 f の出力の左から τ ビットのみを出力する関数である。また $O(q\sigma)$ については、圧縮関数 f の 1 回の計算時間を定数と仮定している。

効率 NMAC の効率は以下のようにまとめられる。

- 鍵長：ハッシュ関数の初期値として使用される鍵 $K_1, K_2 \in \{0, 1\}^n$ の二つである。
- メッセージ M に対するタグを生成するのにかかるハッシュ関数の圧縮関数の呼び出し回数：ハッシュ関数 $F : \{0, 1\}^* \rightarrow \{0, 1\}^n$ の圧縮関数を $f : \{0, 1\}^n \times \{0, 1\}^b \rightarrow \{0, 1\}^n$ とする。ここで、 b は圧縮関数に入力されるメッセージのブロック長である。 F が MD5, RIPEMD-160, SHA-1, SHA-256, SHA-384, SHA-512 などのハッシュ関数であることを想定すれば、 F_{K_2} の圧縮関数の呼び出し回数は $\lceil |M|/b \rceil$ 回または、 $\lceil |M|/b \rceil + 1$ 回である。また、 F_{K_1} の圧縮関数の呼び出し回数は 1 回である。
- 並列処理性：並列処理はできない。

標準化状況 NMAC は応用を指向した HMAC の理論的基礎として設計されたものであり、それ自体については標準化の実績はない。

8.11.2 HMAC

方式 HMAC (Hash-based MAC) は反復型ハッシュ関数 F とタグ長 τ をパラメータとする。出力長 n の反復型ハッシュ関数 $F : \{0, 1\}^* \rightarrow \{0, 1\}^n$ を用いた場合は、 $\tau \leq n$ でなければならない。これらのパラメータを用いた HMAC を $\text{HMAC}[F, \tau]$ と表記する。 $\text{HMAC}[F, \tau] = (\text{HMAC-}\mathcal{K}, \text{HMAC-}\mathcal{G}, \text{HMAC-}\mathcal{V})$ の鍵生成アルゴリズム $\text{HMAC-}\mathcal{K}$ 、タグ生成アルゴリズム $\text{HMAC-}\mathcal{G}$ 、確認アルゴリズム $\text{HMAC-}\mathcal{V}$ はそれぞれ以下のように動作する。

- 鍵生成アルゴリズム HMAC- \mathcal{K} は確率的アルゴリズムであり,

$$K \stackrel{R}{\leftarrow} \{0, 1\}^n$$

を出力する.

- タグ生成アルゴリズム HMAC- $\mathcal{G} : \{0, 1\}^n \times \{0, 1\}^* \rightarrow \{0, 1\}^\tau$ は決定的アルゴリズムであり, 鍵空間は $\{0, 1\}^n$, メッセージ空間は $\{0, 1\}^*$, タグ空間は $\{0, 1\}^\tau$ である. すなわち, 鍵 $K \in \{0, 1\}^n$ とメッセージ $M \in \{0, 1\}^*$ を入力とし, タグ

$$T = \text{HMAC-}\mathcal{G}_K(M) \in \{0, 1\}^\tau$$

を出力する. 図 73 にあるように動作する.

Algorithm HMAC- $\mathcal{G}_K(M)$
 $T \leftarrow$ the left most τ bits of $F(\overline{K} \oplus \text{opad}, (F(\overline{K} \oplus \text{ipad}, M)))$
 Return T

図 73: HMAC のタグ生成アルゴリズム HMAC- $\mathcal{G}_K(\cdot)$.

ハッシュ関数 $F : \{0, 1\}^* \rightarrow \{0, 1\}^n$ の圧縮関数を $f : \{0, 1\}^n \times \{0, 1\}^b \rightarrow \{0, 1\}^n$ とする. ここで, b は圧縮関数に入力されるメッセージのブロック長である. 図 73 で, \overline{K} は, 長さが b ビットになるまで K に 0 を付加して得られる系列である. opad は, 00110110 を長さが b ビットになるまで繰り返して得られる系列である. ipad は, 01011100 を長さが b ビットになるまで繰り返して得られる系列である.

- 確認アルゴリズム HMAC- $\mathcal{V} : \{0, 1\}^n \times \{0, 1\}^* \times \{0, 1\}^\tau \rightarrow \text{accept or reject}$ は決定的アルゴリズムであり, 鍵 $K \in \{0, 1\}^n$, メッセージ $M \in \{0, 1\}^*$, タグ $T \in \{0, 1\}^\tau$ を入力とし,

$$\text{accept or reject} = \text{HMAC-}\mathcal{V}_K(M, T)$$

を出力する. 図 74 にあるように動作する.

<p>Algorithm HMAC-$\mathcal{V}_K(M, T)$ $T' \leftarrow \text{HMAC-}\mathcal{G}_K(M)$ if $T = T'$ then return accept else return reject</p>
--

図 74: HMAC の確認アルゴリズム HMAC- $\mathcal{V}_K(\cdot, \cdot)$.

安全性 Bellare, Canetti, Krawczyk により, 安全性が解析されている [BCK96].

HMAC の安全性は NMAC の安全性に基づく. $K_1 = f(IV, \bar{K} \oplus \text{opad})$, $K_2 = f(IV, \bar{K} \oplus \text{ipad})$ とすれば,

$$F(\bar{K} \oplus \text{opad}, (F(\bar{K} \oplus \text{ipad}, M))) = F_{K_1}(F_{K_2}(M))$$

となる². ここで, IV はハッシュ関数 F で定義されている初期値である. NMAC では鍵 K_1, K_2 が無作為に選択されるので, f が K を鍵とする疑似ランダム関数であるという仮定を設ければ, NMAC の安全性の解析は HMAC に適用できることになる.

opad, ipad はハミング距離が大きくなるように選択されている.

効率 HMAC の効率は以下のようにまとめられる.

- 鍵長: ハッシュ関数の入力の一部として使用される鍵 $K \in \{0, 1\}^n$ の一つである. この鍵の長さはハッシュ関数 F の出力長と等しい.
- メッセージ M に対するタグを生成するのにかかるハッシュ関数の圧縮関数の呼び出し回数: 一カブロック $\bar{K} \oplus \text{opad}$ と $\bar{K} \oplus \text{ipad}$ の処理のため, NMAC での呼び出し回数より 2 回多い. ただし, K を変更しない限り $K_1 = f(IV, \bar{K} \oplus \text{opad})$, $K_2 = f(IV, \bar{K} \oplus \text{ipad})$ を計算して保持しておけば, NMAC と同じ回数となる.
- 並列処理性: 並列処理はできない.

²[BCK96] ではこのように主張されているが, NMAC と HMAC とでは, F への入力長が異なることになるので, パディングを考慮すると, 通常はこの等式は成立しない. ただし, この等式が成立するように NMAC のパディングを変更することは可能であり, このような変更は NMAC の安全性の証明に影響を与えない.

標準化状況 米国情報処理標準であり，FIPS198に制定されている [FIPS198]. NESSIEのportfolioに含まれている [WWW8]. IPSecでは，任意の認証アルゴリズムの実装が許されているが，最低限MD5とSHA-1によるHMACの実装がなされなければならない [RFC2402].

9 まとめ

本報告では，暗号利用モードとMACに関して，その動作原理や安全性の考え方などについて解説すると共に，各種標準化作業や学術出版物などで知られている方式を調査し，それらの工業的，学術的性質，安全性，処理効率などについてまとめた。また，それらを用途別に，秘匿に関する利用モード，認証暗号に関する利用モード，ディスクセクタ向け暗号利用モード認証に関する利用モードに分類した。

提供される機能は，利用モードにより異なるため，利用モードを選択する際にはその性質について十分理解しておく必要がある。また，その際の参考として本報告書が役立つことを期待する。

表 5: 秘匿に関する暗号利用モードのまとめ

	秘匿性に関する注意点	1ビット エラーの 伝播範囲	処理 速度	並列性		復号 関数 の実装	機能に 関する コメント
				暗号化	復号		
ECB	<ul style="list-style-type: none"> ・暗号文を見るだけで平文ブロックが同じ値であるか否かを判定できる。よって、1) 平文がブロックサイズより小さい、2) 平文ブロックが全て異なる、などの特別な理由がない限り用いるべきでない。特に、長い文章を暗号化する際の利用は避けた方がよい。 	1ブロック	1	有り	有り	必要	
CBC	<ul style="list-style-type: none"> ・$2^{n/2}$ ブロック程度以上の平文を暗号化すると、中間一致攻撃により、暗号文を見るだけで平文に関する1ブロック分の情報が得られる可能性がある。 	1ブロック +1ビット	1	無し	有り	必要	
k -CFB	<ul style="list-style-type: none"> ・$2^{n/2}$ ブロック程度以上の平文を暗号化すると、中間一致攻撃により、暗号文を見るだけで平文に関する1ブロック分 (kビット) の情報が得られる可能性がある。 ・kが小さい場合、初期値と平文の組み合わせによっては、鍵ストリームの周期が極端に小さくなる恐れがある。 	$(\lfloor \frac{n}{k} \rfloor \sim \lceil \frac{n}{k} \rceil)$ 平文ブロック +1ビット (1 平文 ブロック = k ビット)	$\frac{k}{n}$	無し	有り	不要	k ビット単位の自動同期回復機能がある。
OFB (k -OFB)	<ul style="list-style-type: none"> ・$k < n$ の場合 $2^{n/2}$ ブロック程度で周期を形成。(k が小さい場合、IVによっては、暗号文から容易に鍵の候補を絞り込むことが可能。) ・初期値の運用を誤ると重大な欠陥につながる恐れがあり、注意が必要。 	1ビット	1	無し	無し	不要	
CTR	<ul style="list-style-type: none"> ・初期値の運用を誤ると重大な欠陥につながる恐れがあり、注意が必要。 	1ビット	1	有り	有り	不要	

参考文献

- [3GPPa] 3GPP TS 35.201 v 3.1.1. Specification of the 3GPP confidentiality and integrity algorithms, Document 1: *f8* and *f9* specification. Available at <http://www.3gpp.org/tb/other/algorithms.htm>.
- [3GPPb] 3GPP TS 35.202 v 3.1.1. Specification of the 3GPP confidentiality and integrity algorithms, Document 2: KASUMI specification. Available at <http://www.3gpp.org/tb/other/algorithms.htm>.
- [AGS97] V. Afanassiev, C. Gehrman, and B. Smeets, “Fast Message Authentication Using Efficient Polynomial Evaluation,” *Fast Software Encryption, 4th International Workshop, FSE'97, Haifa, Israel, January 20–22, 1997, Proceedings*, ed. E. Biham, pp. 190–204, Lecture Notes in Computer Science vol. 1267, Springer-Verlag, 1997.
- [AGPS02] A. Alkassar, A. GERALDY, B. Pfitzmann, and A. R. Sadeghi, “Optimized Self-synchronizing Mode of Operation,” *Fast Software Encryption, 8th International Workshop, FSE2001, Yokohama, Japan, April 2–4, 2001, Revised Papers*, ed. M. Matsui, pp. 78–91, Lecture Notes in Computer Science vol. 2355, Springer-Verlag, 2002.
- [ANSIX3.106] ANSI X 3.106, American National Standard for Information Systems – Data Encryption Algorithm – Modes of Operation,” American National Standard Institute, 1983.
- [ANSIX3.92] ANSI X 3.92, American National Standard for Information Systems – Data Encryption Algorithm,” American National Standard Institute, 1981.
- [AB99] J.H. An and M. Bellare, “Constructing VIL-MACs from FIL-MACs: Message Authentication under Weakened Assumptions,” *Advances in Cryptology — CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15–19, 1999. Proceedings*, ed. M. Wiener, pp. 252–269, Lecture Notes in Computer Science vol. 1666, Springer-Verlag, 1999.
- [AB01] J.H. An and M. Bellare, “Does Encryption with Redundancy Provide Authenticity?” *Advances in Cryptology, — EUROCRYPT*

2001, *International Conference on the Theory and Application of Cryptographic Techniques, Innsbruck, Austria, May 6–10, 2001, Proceedings*, ed. B. Pfitzmann, pp. 512–528, Lecture Notes in Computer Science vol. 2045, Springer-Verlag, 2001.

- [BA01] A. A. Belal and M.A.Abdel-Gawad, “2D-Encryption Mode,” Submitted document for Modes of Operation for Symmetric Key Block Ciphers, Computer Security Resource Center of Information Technology Laboratory, National Institute of Standards and Technology, 2001, available at <http://csrc.nist.gov/CryptoToolkit/modes/proposedmodes/>.
- [BGR95] M. Bellare, R. Guérin, and P. Rogaway, “XOR MACs: New Methods for Message Authentication Using Finite Pseudorandom Functions,” *Advances in Cryptology — CRYPTO ’95, 15th Annual International Cryptology Conference, Santa Barbara, California, USA, August 1995. Proceedings*, ed. D. Coppersmith, pp. 15–28, Lecture Notes in Computer Science vol. 963, Springer-Verlag, 1995.
- [BCK96] M. Bellare, R. Canetti, and H. Krawczyk, “Keying Hash Functions for Message Authentication,” *Advances in Cryptology — CRYPTO ’96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 1996. Proceedings*, ed. N. Koblitz, pp. 1–15, Lecture Notes in Computer Science vol. 1109, Springer-Verlag, 1996.
- [BDJR97] M. Bellare, A. Desai, E. Jorjipii, and P. Rogaway. A concrete security treatment of symmetric encryption. *Proceedings of The 38th Annual Symposium on Foundations of Computer Science, FOCS ’97*, pp. 394–405, IEEE, 1997.
- [BN00] M. Bellare and C. Namprempre, “Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm,” *Advances in Cryptology — ASIACRYPT 2000, 6th International Conference on the Theory and Application of Cryptology and Information Security, Kyoto, Japan, December 3–7, 2000 Proceedings*, ed. T. Okamoto, pp. 531–545, Lecture Notes in Computer Science vol. 1976, Springer-Verlag, 2000.
- [BKR00] M. Bellare, J. Kilian, and P. Rogaway. The security of the cipher block chaining message authentication code. *JCSS*, Vol. 61,

- No. 3, pp. 362–399, 2000. Earlier version in *Advances in Cryptology — CRYPTO '94, LNCS 839*, pp. 341–358, Springer-Verlag, 1994.
- [BPR05] M. Bellare, K. Pietrzak, and P. Rogaway. Improved Security Analyses for CBC MACs. *Advances in Cryptology — CRYPTO 2005, LNCS 3621*, pp. 527–545, Springer-Verlag, 2005.
- [BBKN01] M. Bellare, A. Boldyreva, L. Knudsen, and C. Namprempre, “Online Ciphers and the Hash-CBC Construction,” *Advances in Cryptology — CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, Aug. 2001, Proceedings*, ed. J. Kilian, pp.292–309, Lecture Notes in Computer Science vol. 2139, Springer-Verlag, 2001.
- [BK03] M. Bellare, and T. Kohno. A theoretical treatment of related-key attacks: RKA-PRPs, RKA-PRFs, and applications. *Advances in Cryptology — EUROCRYPT 2003, LNCS 2656*, pp. 491–506, Springer-Verlag, 2003.
- [BRW03] M. Bellare, P. Rogaway, and D. Wagner, “A Conventional Authenticated-Encryption Mode,” Submitted document for Modes of Operation for Symmetric Key Block Ciphers, Computer Security Resource Center of Information Technology Laboratory, National Institute of Standards and Technology, 2003, available at <http://csrc.nist.gov/CryptoToolkit/modes/proposedmodes/>.
- [BB+95] A. Berendschot, B. den Boer, J. Boly, A. Bosselaers, J. Brandt, D. Chaum, I. Damgaard, M. Dichtl, W. Fumy, M. van der Ham, C. Jansen, P. Landrock, B. Preneel, G. Roelofsen, P. de Rooij, J. Vandewalle, *Final Report of Race Integrity Primitives, Lecture Notes in Computer Science*, vol. 1007, Springer-Verlag, 1995.
- [B96] E. Biham, “Cryptanalysis of Triple-Modes of Operation,” Technion technical report CS885, 1996, available at <http://www.cs.technion.ac.il/~biham/publications.html>.
- [BHKKR99] J. Black, S. Halevi, H. Krawczyk, T. Krovets, and P. Rogaway, “UMAC: Fast and Secure Message Authentication,” *Advances in Cryptology — CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15–19,*

1999. *Proceedings*, ed. M. Wiener, pp. 216–233, Lecture Notes in Computer Science vol. 1666, Springer-Verlag, 1999.
- [BR00] J. Black and P. Rogaway, “CBC MACs for Arbitrary-Length Messages: The Three-Key Constructions,” *Advances in Cryptology — CRYPTO 2000, 20th Annual International Cryptology Conference, Santa Barbara, California, USA, August 2000. Proceedings*, ed. M. Bellare, pp. 197–215, Lecture Notes in Computer Science vol. 1880, Springer-Verlag, 2000.
- [BR01] J. Black and P. Rogaway. Comments to NIST concerning AES modes of operations: A suggestion for handling arbitrary-length messages with the CBC MAC. *Second Modes of Operation Workshop*. Available at <http://www.cs.ucdavis.edu/~rogaway/>.
- [BR02] J. Black and P. Rogaway, “A Block-Cipher Modes of Operation for Parallelizable Message Authentication,” *Advances in Cryptology — EUROCRYPT 2002, International Conference on the Theory and Application of Cryptographic Techniques, Amsterdam, the Netherlands, April 28–May 2, 2002. Proceedings*, ed. L. Knudsen, pp. 384–397, Lecture Notes in Computer Science vol. 2332, Springer-Verlag, 2002.
- [BRS02] J. Black, P. Rogaway, and T. Shrimpton, “Black-Box Analysis of the Block-Cipher-Based Hash-Function Constructions from PGV,” *Advances in Cryptology — CRYPT 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 2002, Proceedings*, ed. M. Yung, pp. 320–335, Lecture Notes in Computer Science vol. 2442, Springer-Verlag, 2002.
- [B02] J. Black. Comments on the RMAC algorithm. Comments to NIST. Available at <http://csrc.nist.gov/CryptoToolkit/modes/comments/>.
- [C78] C. M. Campbell, “Design and Specification of Cryptographic Capabilities,” *Computer Security and the Data Encryption Standard*, (ed.) D. K. Brandstad, National Bureau of Standards Special Publications 500-27, U. S. Department of Commerce, February 1978, pp. 54–66.

- [CW79] L. Carter and M. Wegman, “Universal Hash Functions,” *Journal of Computer and System Sciences*, vol. 18, 1979.
- [D93] J. Daemen, “Limitations of the Even-Mansour Construction,” *Advances in Cryptology — ASIACRYPT ’91, International Conference on the Theory and Application of Cryptology*, eds. H. Imai, R.L. Rivest, and T. Matsumoto, pp. 495–499, Lecture Notes in Computer Science vol. 739, Springer-Verlag, 1993.
- [DR99] J. Daemen and V. Rijmen, *AES Proposal: Rijndael*, AES Algorithm Submission, September 3, 1999, available at <http://www.nist.gov/CryptoToolkit>.
- [DH79] W. Diffie and M. E. Hellman, “Privacy and Authentication: An Introduction to Cryptography,” *Proceedings of the IEEE* 67/3, 1979, pp. 397–427.
- [SP800-38A] M. Dworkin, National Institute of Standards and Technology, Special Publication 800-38A, 2001 Edition, Recommendation for Block Cipher Modes of Operation: Methods and Techniques, December 2001.
- [SP800-38B] M. Dworkin, National Institute of Standards and Technology, Special Publication 800-38B, Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication, May 2005.
- [SP800-38C] M. Dworkin, National Institute of Standards and Technology, Special Publication 800-38C, Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality, May, 2004.
- [EM97] S. Even and Y. Mansour, “A Construction of a Cipher from a Single Pseudorandom Permutation,” *J. of Cryptology*, 10(3) 151–161, Summer 1997.
- [FIPS46-3] National Institute of Standards and Technology, Federal Information Processing Standards Publication 46-3, Data Encryption Standard (DES).

- [FIPS81] National Institute of Standards and Technology, Federal Information Processing Standards Publication 81, DES Modes of Operation (DES), 1980.
- [FIPS113] National Institute of Standards and Technology, Federal Information Processing Standards Publication 113, Computer data authentication, 1994.
- [FIPS197] National Institute of Standards and Technology, Federal Information Processing Standards Publication 197, Advanced Encryption Standard (AES).
- [FIPS198] National Institute of Standards and Technology, Federal Information Processing Standards Publication 198, The Keyed-Hash Message Authentication Code (HMAC), 2002.
- [FH03] S. Frankel, and H. Herbert. The AES-XCBC-MAC-96 algorithm and its use with IPsec. Available at <http://www.ieft.org/>.
- [FMP03] P.-A. Fouque, G. Martinet, and G. Poupard, “Practical Symmetric On-line Encryption,” *FSE2003, Tenth Annual Workshop on Fast Software Encryption, February 24–26, 2003, AF-Borgen, Lund, Sweden. Pre-proceedings*, pp. 379–392, Department of Information Technology of Lund Institute of Technology, Lund University, 2003.
- [GD99] V. Gligor and P. Donescu, “Integrity-aware PCBC Encryption Schemes,” *Security Protocols, 7th International Workshop Cambridge, UK, April 19–21, 1999 Proceedings*, eds. B. Christianson, B. Crispo, J.A. Malcolm, and M. Roe, Lecture Notes in Computer Science vol. 1796, Springer-Verlag, 2000.
- [GD00] V. D. Gligor and P. Donescu, “On Message Integrity in Symmetric Encryption,” Submitted document for Modes of Operation for Symmetric Key Block Ciphers, Computer Security Resource Center of Information Technology Laboratory, National Institute of Standards and Technology, 2000, available at <http://csrc.nist.gov/CryptoToolkit/modes/proposedmodes/>.
- [GD01a] V.D. Gligor and P. Donescu, “Fast Encryption and Authentication: XCBC Encryption and XECB Authentication Modes,”

Fast Software Encryption, 8th International Workshop, FSE2001, Yokohama, Japan, April 2–4, 2001. Revised Papers, ed. M. Matsui, pp. 92–108, Lecture Notes in Computer Science vol. 2355, Springer-Verlag, 2002.

- [GD01b] V. D. Gligor and P. Donescu, “Fast Encryption and Authentication: XCBC Encryption and XECB Authentication Modes,” Submitted document for Modes of Operation for Symmetric Key Block Ciphers, Computer Security Resource Center of Information Technology Laboratory, National Institute of Standards and Technology, 2001, available at <http://csrc.nist.gov/CryptoToolkit/modes/proposedmodes/>.
- [GGM86] O. Goldreich, S. Goldwasser, and S. Micali, “How to Construct Random Functions,” *Journal of the ACM*, Vol. 33, No. 4, 792–807, October 1986.
- [GM84] S. Goldwasser and S. Micali, “Probabilistic Encryption,” *J. Computer & System Sciences*, 28: pp.270–299, 1984.
- [HK97] S. Halevi and H. Krawczyk, “MMH: Software Message Authentication in the Gbit/second Rates,” *Fast Software Encryption, 4th International Workshop, FSE’97, Haifa, Israel, January 20–22, 1997, Proceedings*, ed. E. Biham, pp. 172–189, Lecture Notes in Computer Science vol. 1267, Springer-Verlag, 1997.
- [HR03a] S. Halevi and P. Rogaway, “A Parallelizable Enciphering Mode,” Working draft for SISWG, Security in Storage Working Group, March 2003, document available at <http://www.siswg.org/docs/>.
- [HR03b] S. Halevi, and P. Rogaway. A tweakable enciphering mode. *Advances in Cryptology — CRYPTO 2003, LNCS 2729*, pp. 482–499, Springer-Verlag, 2003.
- [HP99] H. Handschuh and B. Preneel, “On the Security of Double and 2-key Triple Modes of Operation,” *Fast Software Encryption, 6th International Workshop, FSE’99, Rome, Italy, March 1999, Proceedings*, ed. L. Knudsen, pp. 215–230, Lecture Notes in Computer Science vol. 1636, Springer-Verlag, 1999.

- [H01c] H. Hellström, “Propagating Cipher Feedback,” Submitted document for Modes of Operation for Symmetric Key Block Ciphers, Computer Security Resource Center of Information Technology Laboratory, National Institute of Standards and Technology, 2001, available at <http://csrc.nist.gov/CryptoToolkit/modes/proposedmodes/>.
- [H01a] H. M. Heys, “Delay Characteristics of Statistical Cipher Feedback Mode,” *IEEE Pacific Rim Conference on Communications, Computers, and Signal Processing — PACRIM 2001, Victoria, British Columbia*, 2001, available at <http://www.engr.mun.ca/~howard/Research/Papers/index.html>.
- [H01b] H. M. Heys, “An Analysis of the Statistical Self-Synchronization of Stream Ciphers,” *Proceedings of INFOCOM 2001, Anchorage, Alaska*, pp. 897–904, 2001, available at <http://www.engr.mun.ca/~howard/Research/Papers/index.html>.
- [H03a] H. M. Heys, “Analysis of the Statistical Cipher Feedback Mode of Block Ciphers,” *IEEE Transactions on Computers*, vol.=52, no. 1, pp. 77–92, January 2003, available at <http://www.engr.mun.ca/~howard/Research/Papers/index.html>.
- [H03b] P. Hoffman. The AES-XCBC-PRF-128 algorithm for IKE. Available at <http://www.ietf.org/>.
- [HKPR03] D. Hong, J-S. Kang, B. Preneel, and H. Ryu. A concrete security analysis for 3GPP-MAC. *Fast Software Encryption, FSE 2003, LNCS 2887*, pp. 154–169, Springer-Verlag, 2003.
- [ISO8372] ISO 8372: 1987, Information Processing – Modes of operation for a 64-bit block cipher algorithm (ANSI X3.92-1981 を参照している).
- [ISOIEC9797-1] ISO/IEC 9797-1. Information technology — security techniques — data integrity mechanism using a cryptographic check function employing a block cipher algorithm. International Organization for Standards, Geneva, Switzerland, 1999. Second edition.
- [ISO10116] ISO/IEC 10116:1997, Information technology – Security techniques – Modes of operation for an n-bit block cipher algorithm, 2002-6-26.

- [IK03a] T. Iwata and K. Kurosawa. OMAC: One-Key CBC MAC. *Fast Software Encryption, FSE 2003, LNCS 2887*, pp. 129–153, Springer-Verlag, 2003.
- [IK03b] T. Iwata and K. Kurosawa. Stronger security bounds for OMAC, TMAC and XCBC. *Progress in Cryptology — INDOCRYPT 2003, LNCS 2904*, pp. 402–415, 2003.
- [IK03c] T. Iwata and K. Kurosawa. On the correctness of security proofs for the 3GPP confidentiality and integrity algorithms. *Ninth IMA International Conference on Cryptography and Coding, LNCS 2898*, pp. 306–318, Springer-Verlag, 2003.
- [JJ+02a] É. Jaulmes, A. Joux, and F. Valette. On the security of randomized CBC-MAC beyond the birthday paradox limit: A new construction. *Fast Software Encryption, FSE 2002, LNCS 2365*, pp. 237–251, Springer-Verlag, 2002.
- [JJ+02b] É. Jaulmes, A. Joux, and F. Valette. On the security of randomized CBC-MAC beyond the birthday paradox limit: A new construction. Full version. Available at Cryptology ePrint Archive, Report 2001/074, <http://eprint.iacr.org/>.
- [JJ+02c] É. Jaulmes, A. Joux, and F. Valette. RMAC, A randomized MAC beyond the birthday paradox limit. *Second Modes of Operation Workshop*, 2001. Available at <http://csrc.nist.gov/CryptoToolkit/modes>.
- [JMV02] A. Joux, G. Martinet, and F. Valette, “Blockwise Adaptive Attackers: Revisiting the (In)security of Some Provably Secure Encryption Modes: CBC, GEM, IACBC,” *Advances in Cryptology — CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, Aug. 18–22, 2002. Proceedings*, ed. M. Yung, pp. 17–30, Lecture Notes in Computer Science vol. 2442, Springer-Verlag, 2002.
- [J03] A. Joux, “Cryptanalysis of the EMD Mode of Operation,” *Advances in Cryptology — EUROCRYPT 2003, International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4–8, 2003, Proceedings*, ed. E. Biham, pp. 1–16, Lecture Notes in Computer Science vol. 2656, Springer-Verlag, 2003.

- [JKRW01] O. Jung, S. Kuhn, C. Ruland, and K. Wollenweber, “Enhanced modes of operation for the encryption in high-speed networks and their impact on QoS,” *Information Security and Privacy: 6th Australasian Conference, ACISP 2001, Sydney, Australia, July 11–13, 2001, Proceedings*, eds. V. Varadharajan and Y. Mu, pp. 344–359, Lecture Notes in Computer Science vol. 2119, Springer-Verlag, 2001.
- [JR99] O. Jung and C. Ruland, “Encryption with statistical self-synchronization in synchronous broadband networks,” *Cryptographic Hardware and Embedded Systems, First International Workshop, CHES ’99, Worcester, MA, USA, August 12–13, 1999, Proceedings*, eds. C.K.Koç and C.Paar, pp. 340–352, Lecture Notes in Computer Science vol. 1717, Springer-Verlag, 1999.
- [J02] J. Jonsson, “On the Security of CTR + CBC-MAC,” *Selected Areas in Cryptography, 9th Annual Workshop, SAC 2002, St. John’s, Newfoundland, Canada, Aug. 2002, Revised Papers*, ed. K. Nyberg and H. Heys, pp. 76–93, Lecture Notes in Computer Science vol. 2595, Springer-Verlag, 2002.
- [J00] C. S. Jutla, “Encryption Modes with Almost Free Message Integrity,” Submitted document for Modes of Operation for Symmetric Key Block Ciphers, Computer Security Resource Center of Information Technology Laboratory, National Institute of Standards and Technology, 2000, available at <http://csrc.nist.gov/CryptoToolkit/modes/proposedmodes/>.
- [J01] C.S. Jutla, “Encryption Modes with Almost Free Message Integrity,” *Advances in Cryptology — EUROCRYPT 2001, International Conference on the Theory and Application of Cryptographic Techniques, Innsbruck, Austria, May 6–10, 2001. Proceedings*, ed. B. Pfitzmann, pp. 529–544, Lecture Notes in Computer Science vol. 2045, Springer-Verlag, 2001.
- [KY00a] J. Katz and M. Yung, “Complete Characterization of Security Notions for Probabilistic Private-key Encryption,” Proceedings of the Thirty-second Annual ACM Symposium on Theory of Computing, ACM, 2000.

- [KY00b] J. Katz and M. Yung, “Unforgeable Encryption and Chosen Cipher Secure Modes of Operation,” *Fast Software Encryption, 7th International Workshop, FSE 2000, New York, NY, USA, April 2000, Proceedings*, ed. B. Schneier, pp. 284–299, Lecture Notes in Computer Science vol. 1978, Springer-Verlag, 2001.
- [KR96] J. Kilian and P. Rogaway, “How to Protect DES against Exhaustive Search (an Analysis of DESX),” *Advances in Cryptology — CRYPTO ’96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 1996. Proceedings*, ed. N. Koblitz, pp. 252–267, Lecture Notes in Computer Science vol. 1109, Springer-Verlag, 1996.
- [K00] L. R. Knudsen, “Block Chaining Modes of Operation,” Submitted document for Modes of Operation for Symmetric Key Block Ciphers, Computer Security Resource Center of Information Technology Laboratory, National Institute of Standards and Technology, 2000, available at <http://csrc.nist.gov/CryptoToolkit/modes/proposedmodes/>.
- [K02] L.R. Knudsen. Analysis of RMAC. Comments to NIST. Available at <http://csrc.nist.gov/CryptoToolkit/modes/comments/>.
- [KVV03] T. Kohno, J. Viega, and D. Whiting, “The CWC Authenticated Encryption (Associated Data) Mode,” Submitted document for Modes of Operation for Symmetric Key Block Ciphers, Computer Security Resource Center of Information Technology Laboratory, National Institute of Standards and Technology, 2003, available at <http://csrc.nist.gov/CryptoToolkit/modes/proposedmodes/>.
- [KI03] K. Kurosawa and T. Iwata, “TMAC, Two-Key CBC MAC,” *Topics in Cryptology — CT-RSA 2003, The Cryptographers’ Track at the RSA Conference 2003, San Francisco, CA, USA, April 13–17, 2003. Proceedings*, ed. M. Joye, pp. 33–49, Lecture Notes in Computer Science vol. 2612, Springer-Verlag, 2003.
- [LN94] R. Lidl and H. Niederreiter. Introduction to finite fields and their applications, revised edition. Cambridge University Press, 1994.

- [LRW00] H. Lipmaa, P. Rogaway, and D. Wagner, “Comments to NIST Concerning AES Modes of Operations: CTR-mode Encryption,” available at <http://csrc.nist.gov/>.
- [LR02] M. Liskov, R.L. Rivest, and D. Wagner, “Tweakable Block Ciphers,” *Advances in Cryptology — CRYPTO 2002, 22nd Annual International Cryptology Conference Santa Barbara, California, USA, August 18–22, 2002. Proceedings*, ed. M. Yung, pp. 31–46, Lecture Notes in Computer Science vol. 2442, Springer-Verlag, 2002.
- [LR88] M. Luby and C. Rackoff, “How to Construct Pseudorandom Permutations from Pseudorandom Functions,” *SIAM J. Comput.*, vol. 17, no. 2, April 1988.
- [L96] S. Lucks, “Faster Luby-Rackoff Ciphers,” *Fast Software Encryption, Third International Workshop, Cambridge, UK, February 1996, Proceedings*, ed. D. Gollmann, pp. 189–203, Lecture Notes in Computer Science vol. 1039, Springer-Verlag, 1996.
- [M91] U. M. Maurer, “New Approaches to the Design of Self-Synchronizing Stream Ciphers,” *Advances in Cryptology — EURO-CRYPT ’91, Brighton, UK*, ed. D.W.Davies, pp. 458–471, Lecture Notes in Computer Science vol. 547, Springer-Verlag, 1991.
- [MRS88] S. Micali, C. Rackoff, and R. Sloan, “The notion of security for probabilistic cryptosystems,” *SIAM J. of Computing*, April 1988.
- [M02] C.J. Mitchell, “The Security of Two-key DESX,” COSIC Seminar, Katholieke Universiteit Leuven, 15th March 2002, Leuven, Belgium.
- [MI02a] S. Moriai and H. Imai, “2-Key XCBC: The CBC-MAC for Arbitrary Length Messages by the Two-key Construction,” a talk at the *Recent Results* session of *Fast Software Encryption, 9th International Workshop, FSE 2002, Leuven, Belgium, February 4–6, 2002*.
- [MI02b] S. Moriai and H. Imai, “2-Key XCBC: The CBC MAC for Arbitrary-Length Messages by the Two-Key Construction,” *Proceedings of SCIS2002, The 2002 Symposium on Cryptography and Information Security*, The Institute of Electronics, Information and Communication Engineers, 2002 (in Japanese).

- [MV04] D. McGrew, J. Viega, “The Security and Performance of the Galois/Counter Mode (GCM) of Operation,” *Progress in Cryptology — INDOCRYPT 2004, 5th International Conference on Cryptology in India, Chennai, India, December 20-22, 2004, Proceedings*. pp. 343-355, Lecture Notes in Computer Science vol. 3348, Springer-Verlag, 2004. Full version is available at <http://eprint.iacr.org/2004/193/>.
- [MV05] D. McGrew, J. Viega, “The Galois/Counter Mode of Operation (GCM),” May 2005, available at <http://csrc.nist.gov/CryptoToolkit/modes/proposedmodes/>.
- [NR99] M. Naor and O. Reingold. On the construction of pseudorandom permutations: Luby-Rackoff revised. *J. Cryptology*, vol. 12, no. 1, pp. 29–66, Springer-Verlag, 1999.
- [NR03] M. Naor and O. Reingold, “A Pseudo-Random Encryption Mode,” Working draft for SISWG, Security in Storage Working Group, document available at <http://www.siswg.org/docs/>.
- [PR00] E. Petrank and C. Rackoff. CBC MAC for real-time data sources. *J. Cryptology*, Vol. 13, No. 3, pp. 315–338, Springer-Verlag, 2000.
- [PGV94] B. Preneel, R. Govaerts, and J. Vandewalle, “Hash functions based on block ciphers: A synthetic approach,” *Advances in Cryptology — CRYPTO ’93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 1993. Proceedings*, ed. D.R. Stinson, pp. 368–378, Lecture Notes in Computer Science vol. 773, Springer-Verlag, 1994.
- [PvO96] B. Preneel and P. C. van Oorschot. On the security of two MAC algorithms. *Advances in Cryptology — EUROCRYPT ’96, LNCS 1070*, pp. 19–32, Springer-Verlag, 1996.
- [RFC2040] R. Baldwin and R. Rivest, “The RC5, RC5-CBC, RC5-CBC-Pad, and RC5-CTS Algorithms,” RFC 2040 (1996), available at <http://www.ietf.org/rfc/rfc2040.txt>.
- [RFC2402] S. Kent, “IP Authentication Header,” RFC 2402 (1998), available at <http://www.ietf.org/rfc/rfc2402.txt>.

- [R97] R.L. Rivest, “All-Or-Nothing Encryption and the Package Transform,” *Fast Software Encryption, 4th International Workshop, FSE’97, Haifa, Israel, January 20–22, 1997, Proceedings*, ed. E. Biham, pp. 210–218, Lecture Notes in Computer Science vol. 1267, Springer-Verlag, 1997.
- [R95] P. Rogaway. Bucket hashing and its application to fast message authentication. *Advances in Cryptology — CRYPTO ’95, LNCS 963*, pp. 29–42, Springer-Verlag, 1995.
- [RBBK01a] P. Rogaway, M. Bellare, J. Black, and T. Krovetz, “OCB: A Block-cipher Mode of Operation for Efficient Authenticated Encryption,” *Eighth ACM conference on computer and communications security CCS-8*, ACM Press, 2001.
- [RBBK01b] P. Rogaway, M. Bellare, J. Black, and T. Krovetz, “OCB: A Block-cipher Mode of Operation for Efficient Authenticated Encryption,” Submitted document for Modes of Operation for Symmetric Key Block Ciphers, Computer Security Resource Center of Information Technology Laboratory, National Institute of Standards and Technology, 2001, available at <http://csrc.nist.gov/CryptoToolkit/modes/proposedmodes/>.
- [R02a] P. Rogaway. Comments on NIST’s RMAC proposal. Comments to NIST. Available at <http://www.cs.ucdavis.edu/~rogaway/xcbc/index.html>. Also available at <http://csrc.nist.gov/CryptoToolkit/modes/comments/>.
- [R02b] P. Rogaway, “The EMD Mode of Operation (A Tweaked, Wide-Blocksize, Strong PRP),” *Cryptology ePrint Archive 2002/148*, <http://eprint.iacr.org/2002/148/>.
- [RW03] P. Rogaway and D. Wagner, “A Critique of CCM,” available at <http://www.cs.berkeley.edu/~daw/papers/ccm.html>.
- [V02] S. Vaudenay, “Security Flaws Induced by CBC Padding — Applications to SSL, IPSEC, WTLS...,” *Advances in Cryptology — EUROCRYPT 2002, International Conference on the Theory and Application of Cryptographic Techniques, Amsterdam, the Netherlands, April 28–May 2, 2002, proceedings*, ed. L. Knudsen, pp. 534–

546, Lecture Notes in Computer Science vol. 2332, Springer-Verlag, 2002.

- [WC81] M. Wegman and L. Carter, “New Hash Functions And Their Use in Authentication And Set Equality,” *Journal of Computer and System Sciences*, vol. 22, 1981.
- [W98] D. Wagner, “Cryptanalysis of Some Recently-proposed Multiple Modes of Operation,” *Fast Software Encryption, 5th International Workshop, FSE’98, Paris, France, March 1998. Proceedings*, ed. S. Vaudenay, pp. 254–269, Lecture Notes in Computer Science vol. 1372, Springer-Verlag, 1998.
- [W02a] D. Wagner, Comments on RMAC. Comments to NIST. Available at <http://csrc.nist.gov/CryptoToolkit/modes/comments/>.
- [W02b] D. Wagner, “OFB and CFB modes: A Cautionary Note Regarding IV Selection,” Rump-session Talk at CRYPTO 2002, 22nd Annual International Cryptology Conference Santa Barbara, California, USA, Aug. 18–22, 2002.
- [WHF02] D. Whiting, R. Housley, and F. Ferguson, “Counter with CBC-MAC (CCM) — AES Mode of Operation,” Submitted document for Modes of Operation for Symmetric Key Block Ciphers, Computer Security Resource Center of Information Technology Laboratory, National Institute of Standards and Technology, 2002, available at <http://csrc.nist.gov/CryptoToolkit/modes/proposedmodes/>.
- [WHR02] D. Whiting, R. Housley, and N. Ferguson, “AES Encryption & Authentication Using CTR Mode & CBC-MAC,” *IEEE P802.11 doc 02/001r2*, May 2002.
- [WWW1] SISWG, Security in Storage Working Group, An IEEE Information Assurance Activity, URL at <http://www.siswg.org/>
- [WWW2] CRYPTREC, Cryptography Research and Evaluation Committees, <http://www.ipa.go.jp/security/enc/CRYPTREC/>
- [WWW3] CRYPTREC, Cryptography Research and Evaluation Committees, <http://cryptrec.nict.go.jp/>

[WWW4] <http://www.itl.nist.gov/fipspubs/>

[WWW5] <http://www.nist.gov/>

[WWW6] <http://csrc.nist.gov/publications/nistpubs/>

[WWW7] <http://www.ansi.org/>

[WWW8] <http://www.cosic.esat.kuleuven.ac.be/nessie/>

[WWW9] <http://www.nist.gov/>