ブロック暗号利用モードの
安全性に関する調査

２００３年１１月

茨城大学

岩田 哲

ブロック暗号利用モードの安全性に関する調査

平成 15 年 11 月

要旨： 本報告書では，現在利用されている，もしくは提案されているブロック暗号利用モードのうち，主に認証を目的にしているものについて，安全性の観点からその特徴をまとめる．効率についても言及する．

目 次

1 序論

本報告書では，現在利用されている，もしくは提案されているブロック暗号利用モードのうち，主に認証を目的にしているものについて，安全性の観点からその特徴をまとめる．効率についても言及する．

ブロック暗号利用モード　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** (privacy) と**認証** (authenticity) 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** (encryption scheme), 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) 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** (authenticated encryption scheme) 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．Fig. 1 参照.

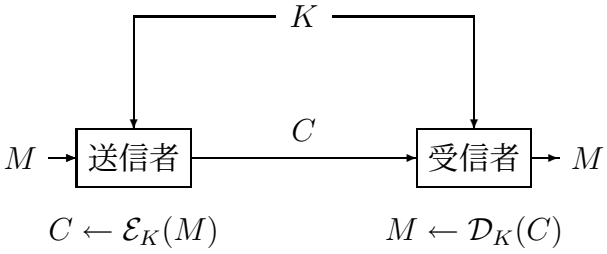

Fig. 1. 暗号化方式のモデル

- 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．Fig. 2 参照.

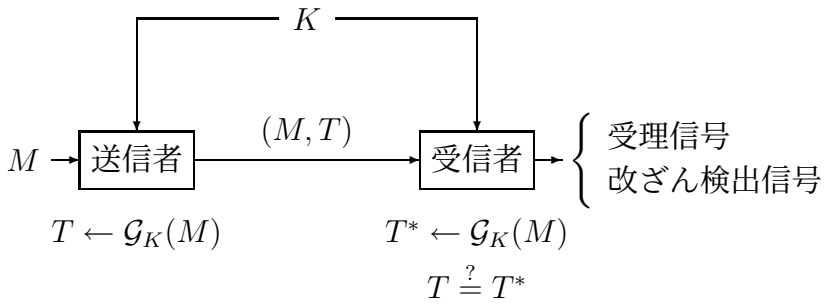

Fig. 2. メッセージ認証コードのモデル

- 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 $\mathcal{D}$ を用いて，暗号文 C と鍵 K から平文 $M = \mathcal{D}_K(C)$ 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て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．Fig. 3 参照.

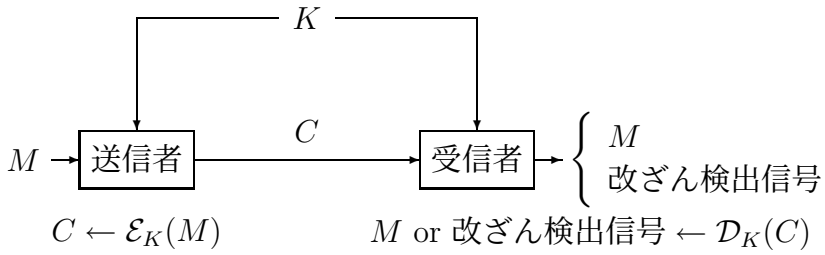

Fig. 3. メッセージ認証つき暗号化方式のモデル

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 [7]．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 [5] が，ランダムビット列からの識別不能性 (indistinguishability from random strings) 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 (unforgeability) 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，安全性を主眼においているが，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．

本報告書の構成　2 章でブロック暗号利用モードに関連する標準の文書についてまとめる．3 章では，ブロック暗号利用モードの一般的な定義を述べる．4 章では，ブロック暗号，およびメッセージ認証コードの一般的な安全性の定義を述べる．5 章では CBC MAC とその変形を扱う．CBC MAC, EMAC, RMAC, XCBC, TMAC, OMAC について，仕様と安全性，効率，標準化状況などについて述べる．6 章では並列計算可能なメッセージ認証コードを扱う．XOR MAC, XECB MAC, PMAC について，仕様と安全性，効率，標準化状況などについて述べる．7 章ではその他の MAC として，$f9$ について述べる．

なお，本報告書の記述はブロック暗号利用モードの仕様を定義するものではない．これらモードの利用と実装にあたっては，各標準の文書などを参照して頂きたい．

2　標準化について

本章では，標準化について述べる．

2.1 NIST

米国 NIST (National Institute of Standards and Technology) は，FIPS (Federal Information Processing Standard) と SP (Special Publications) を策定しており，暗号技術関連の標準も含まれている．ブロック暗号利用モードに関する主なものは以下のとおりである．

FIPS 46-2　DES ANSI
FIPS 81　DES 利用モード，ECB, CBC, CFB, OFB
FIPS 113　CBC MAC
FIPS 197　AES

2001 年 11 月の AES の FIPS の策定に伴い，NIST は AES の利用モードを策定中である．2003 年 11 月現在で提案された利用モードを以下に挙げる．

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	その他
2DEM	OMAC	CCM	KFB
ABC	PMAC	CWC	AES-hash
CTR	RMAC	EAX	
IGE	TMAC	IACBC	
	XCBC	IAPM	
	XECB	OCB	
		PCFB	
		XCBC	

2003 年 11 月現在，5 つの秘匿に関する利用モードが SP800-38A [47] として定義されている．これには，FIPS 81 で定義した 4 つのモードと，あたらに CTR が加えられた．SP800-38B のドラフト版は，メッセージ認証コードとして，RMAC を，また，SP800-38C のドラフト版では，メッセージ認証つき暗号化方式として，CCM を定義している．以下にまとめる．

SP800-38A　暗号化方式 (ECB, CBC, CFB, OFB, CTR)
SP800-38B　メッセージ認証コード (RMAC, ドラフト版である)
SP800-38C　メッセージ認証つき暗号化方式 (CCM, ドラフト版である)

SP800-38B と，SP800-38C はドラフト版であり，今後の動向を注視する必要がある．

2.2 ISO/IEC

ISO (International Organization for Standardization), 及び，IEC (International Electrotechnical Commission) は，個々の標準の他に，一部の国際規格を共同で策定している．日本では ISO/IEC の標準化をうけて，日本語化したものを JIS として発行している．

ブロック暗号利用モードに関する標準化文書として主なものを挙げる．

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．

2.3　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．

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

ISO　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

2.4　NESSIE

NESSIE (New European Schemes for Signatures, Integrity, and Ecnryption) [40] 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．

2.5　3GPP

3GPP (3rd Generation Partnership Project) では，ブロック暗号 KASUMI 及び，その利用モードが策定されている．暗号化方式として $f8$ が，メッセージ認証コードとして $f9$ が策定されている [1, 2].

3 準備

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．

3.1 記法

A が集合である場合，$a \xleftarrow{R} A$ 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，ということをあらわす．A が確率的アルゴリズムでれば，$a \xleftarrow{R} A$ 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，$(\{0,1\}^l)^+ = \bigcup_{l'=1,2,\ldots} \{0,1\}^{ll'}$ 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，$\{0,1\}^* = \bigcup_{l=0,1,2,\ldots} \{0,1\}^l$ 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輪をあらわす．a がビット列のとき，$|a|$ は a のビット長をあらわす．

3.2 ブロック暗号

ブロック暗号 (block cipher) E とは，$E : \mathcal{K}_E \times \mathcal{M}_E \to \mathcal{M}_E$ 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．

3.3 暗号化方式

暗号化方式 (encryption scheme) 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 $\mathit{ENC} = (\mathit{ENC}\text{-}\mathcal{K}, \mathit{ENC}\text{-}\mathcal{E}, \mathit{ENC}\text{-}\mathcal{D})$ から成る．$\mathit{ENC}\text{-}\mathcal{K}$ を鍵生成アルゴリズム，$\mathit{ENC}\text{-}\mathcal{E}$ を暗号化アルゴリズム，$\mathit{ENC}\text{-}\mathcal{D}$ 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 $\mathit{ENC}\text{-}\mathcal{K}$ 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．$K \xleftarrow{R} \mathit{ENC}\text{-}\mathcal{K}$ と表記する．暗号化アルゴリズム $\mathit{ENC}\text{-}\mathcal{E}$ 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．$C \xleftarrow{R} \mathit{ENC}\text{-}\mathcal{E}(K;M)$ や $C \leftarrow \mathit{ENC}\text{-}\mathcal{E}(K;M)$ 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 $\mathit{ENC}\text{-}\mathcal{D}$ 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．$M \leftarrow \mathit{ENC}\text{-}\mathcal{D}(K; C)$ と表記する．

全ての鍵 K と全てのメッセージ M に対し，

$$\mathit{ENC}\text{-}\mathcal{D}(K; \mathit{ENC}\text{-}\mathcal{D}(K; M))) = M$$

でなければならない．

$\mathit{ENC}\text{-}\mathcal{E}(K; \cdot)$ と $\mathit{ENC}\text{-}\mathcal{D}(K; \cdot)$ を，$\mathit{ENC}\text{-}\mathcal{E}_K(\cdot)$ や $\mathit{ENC}\text{-}\mathcal{D}_K(\cdot)$ と表記する．

例として，一般的な ECB, CBC, OFB, CFB, CTR [47], ディスク暗号化用の NR [39], CMC [22], 3GPP の $f8$ [1, 2] などがある．

3.4 メッセージ認証コード

メッセージ認証コード (Message Authentication Code, MAC) 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 $\mathit{MAC} = (\mathit{MAC}\text{-}\mathcal{K}, \mathit{MAC}\text{-}\mathcal{G}, \mathit{MAC}\text{-}\mathcal{V})$ から成る．$\mathit{MAC}\text{-}\mathcal{K}$ を鍵生成アルゴリズム，$\mathit{MAC}\text{-}\mathcal{G}$ をタグ生成アルゴリズム，$\mathit{MAC}\text{-}\mathcal{V}$ を確認アルゴリズムという．また，メッセージ空間 $\mathcal{M}_{\mathit{MAC}}$ とタグ空間 $\mathcal{T}_{\mathit{MAC}}$ をもつ．

鍵生成アルゴリズム $\mathit{MAC}\text{-}\mathcal{K}$ 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．$K \xleftarrow{R} \mathit{MAC}\text{-}\mathcal{K}$ と表記する．

タグ生成アルゴリズム $\mathit{MAC}\text{-}\mathcal{G}$ 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 $M \in \mathcal{M}_{\mathit{MAC}}$ を入力とし，タグ $T \in \mathcal{T}_{\mathit{MAC}}$ を出力する．$T \xleftarrow{R} \mathit{MAC}\text{-}\mathcal{G}(K; M)$ や $T \leftarrow \mathit{MAC}\text{-}\mathcal{G}(K; M)$ 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．

$\mathit{MAC}\text{-}\mathcal{G}$ 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．

$\mathit{MAC}\text{-}\mathcal{G}$ 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 $MAC\text{-}\mathcal{V}$ 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．$MAC\text{-}\mathcal{V}(K; M; T) =$ accept や，$MAC\text{-}\mathcal{V}(K; M; T) =$ reject と表記する．

全ての鍵 K と全てのメッセージ M に対し，

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

でなければならない．

$MAC\text{-}\mathcal{G}(K; \cdot)$ と $MAC\text{-}\mathcal{V}(K; \cdot; \cdot)$ を，$MAC\text{-}\mathcal{G}_K(\cdot)$ や $MAC\text{-}\mathcal{V}_K(\cdot, \cdot)$ と表記する．

例として，CBC MAC [7], EMAC [10, 42], RMAC [29, 30, 31], XCBC [12], TMAC [36], OMAC [26], XOR MAC [6], XECB MAC [20], PMAC [14], $f9$ [1, 2] などがある．

3.5 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 (authenticated encryption scheme) 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 $AE = (AE\text{-}\mathcal{K}, AE\text{-}\mathcal{E}, AE\text{-}\mathcal{D})$ から成る．$AE\text{-}\mathcal{K}$ を鍵生成アルゴリズム，$AE\text{-}\mathcal{E}$ を暗号化アルゴリズム，$AE\text{-}\mathcal{D}$ 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 $AE\text{-}\mathcal{K}$ 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．$K \xleftarrow{R} AE\text{-}\mathcal{K}$ と表記する．暗号化アルゴリズム $AE\text{-}\mathcal{E}$ 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．$C \xleftarrow{R} AE\text{-}\mathcal{E}(K; M)$ や $C \leftarrow AE\text{-}\mathcal{E}(K; M)$ 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 $AE\text{-}\mathcal{D}$ 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．$M \leftarrow AE\text{-}\mathcal{D}(K; C)$ や，改ざん検出信号 $\leftarrow AE\text{-}\mathcal{D}(K; C)$ と表記する．

全ての鍵 K と全てのメッセージ M に対し，

$$AE\text{-}\mathcal{D}(K; AE\text{-}\mathcal{D}(K; M))) = M$$

でなければならない．

$AE\text{-}\mathcal{E}(K; \cdot)$ と $AE\text{-}\mathcal{D}(K; \cdot)$ を，$AE\text{-}\mathcal{E}_K(\cdot)$ や $AE\text{-}\mathcal{D}_K(\cdot)$ と表記する．

例として，IACBC, IAPM [33], OCB [46], XCBC [20], CCM [51, 32], CWC [34], EAX [9] などがある．

4 安全性定義

本章では，ブロック暗号とメッセージ認証コードの一般的な安全性定義を述べる．

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，擬似ランダム置換 (pseudorandom permutation) としての安全性と強擬似ランダム置換 (super-pseudorandom permutation, あるいは strong-pseudorandom permutation) としての安全性がある．

4.1.1 擬似ランダム置換族

ブロック暗号 $E : \mathcal{K}_E \times \mathcal{M}_E \to \mathcal{M}_E$ は，$\mathcal{M}_E$ 上の置換族 $\{E_K(\cdot) \in \mathrm{Perm}(\mathcal{M}_E) \mid K \in \mathcal{K}_E\}$ と捉えることができる．ここで，$\mathrm{Perm}(\mathcal{M}_E)$ 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擬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 $\{E_K(\cdot) \in \mathrm{Perm}(\mathcal{M}_E) \mid K \in \mathcal{K}_E\}$ と $\mathcal{M}_E$ 上のすべての置換の集合 $\mathrm{Perm}(\mathcal{M}_E)$ 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 $E : \mathcal{K}_E \times \mathcal{M}_E \to \mathcal{M}_E$ の，敵 A に対する，擬似ランダム置換としての安全性は，アドバンテージ $\mathbf{Adv}^{\mathrm{prp}}_E(A)$ によって評価される．ここで,

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

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．

$\mathrm{Perm}(\mathcal{M}_E)$ から一様ランダムにえばられた 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．

$$\mathbf{Adv}^{\mathrm{prp}}_E(t, q) \stackrel{\mathrm{def}}{=} \max_A \{\mathbf{Adv}^{\mathrm{prp}}_E(A)\}$$

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 $\mathbf{Adv}^{\mathrm{prp}}_E(t, q)$ 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，$\mathbf{Adv}^{\mathrm{prp}}_E(t,q)$ 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 $\{E_K(\cdot) \in \mathrm{Perm}(\mathcal{M}_E) \mid K \in \mathcal{K}_E\}$ と $\mathcal{M}_E$ 上のすべての置換の集合 $\mathrm{Perm}(\mathcal{M}_E)$ 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 $E : \mathcal{K}_E \times \mathcal{M}_E \to \mathcal{M}_E$ 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 $\mathbf{Adv}^{\mathrm{sprp}}_E(A)$ によって評価される．ここで,

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

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．

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

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，$\mathbf{Adv}^{\mathrm{sprp}}_E(t,q_e,q_d)$ 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 [8] などがこれに含まれる．

また，理想的ブロック暗号モデル (ideal-block cipher model) 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 [29, 30] の安全性解析に用いられた．

4.2 メッセージ認証コードの安全性

メッセージ認証コード $MAC = (MAC\text{-}\mathcal{K}, MAC\text{-}\mathcal{G}, MAC\text{-}\mathcal{V})$ の安全性には，弱偽造不可能性 (weak unforgeability) と強偽造不可能性 (strong unforgeability) 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．$A^{MAC\text{-}\mathcal{G}_K(\cdot),MAC\text{-}\mathcal{V}_K(\cdot,\cdot)}$ は，メッセージ M に対し，タグ $T = MAC\text{-}\mathcal{G}_K(M)$ を返すタグ生成オラクル $MAC\text{-}\mathcal{G}_K(\cdot)$ と，メッセージ，タグのペア (M,T) に対し，accept or reject $= MAC\text{-}\mathcal{V}_K(M,T)$ を返す確認オラクル $MAC\text{-}\mathcal{V}_K(\cdot,\cdot)$ 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．

4.2.1 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 $MAC = (MAC\text{-}\mathcal{K}, MAC\text{-}\mathcal{G}, MAC\text{-}\mathcal{V})$ 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 $M_1, \ldots, M_q$ を質問し，その答え $T_1, \ldots, T_q$ 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 $(M'_1, T'_1), \ldots, (M'_{q'}, T'_{q'})$ 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，$M'_i \notin \{M_1, \ldots, M_j\}$ 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，$\{M_1, \ldots, M_j\}$ 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 $\mathbf{Adv}^{\text{w-uf}}_{E}(A)$ によって評価される．ここで，

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

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．

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

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．

4.2.2 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 $M_1, \ldots, M_q$ を質問し，その答え $T_1, \ldots, T_q$ 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 $(M'_1, T'_1), \ldots, (M'_{q'}, T'_{q'})$ 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，$(M'_i, T'_i) \notin \{(M_1, T_1), \ldots, (M_j, T_j)\}$ 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．$\{(M_1, T_1), \ldots, (M_j, T_j)\}$ 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 $MAC = (MAC\text{-}\mathcal{K}, MAC\text{-}\mathcal{G}, MAC\text{-}\mathcal{V})$ 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 $\mathbf{Adv}^{\text{s-uf}}_{E}(A)$ によって評価される．ここで，

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

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 $MAC = (MAC\text{-}\mathcal{K}, MAC\text{-}\mathcal{G}, MAC\text{-}\mathcal{V})$ 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．

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

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．

4.2.3 $MAC\text{-}\mathcal{G}$ が決定的アルゴリズムである場合の安全性

$MAC\text{-}\mathcal{G}$ 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 accpet 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いで，q と σ にこれらを含めるのが一般的である．$MAC\text{-}\mathcal{G}$ 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 (unforgeability) 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．$A^{MAC\text{-}\mathcal{G}_K(\cdot)}$ は，メッセージ M に対し，タグ $T = MAC\text{-}\mathcal{G}_K(M)$ を返すタグ生成オラクル $MAC\text{-}\mathcal{G}_K(\cdot)$ 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 $MAC = (MAC\text{-}\mathcal{K}, MAC\text{-}\mathcal{G}, MAC\text{-}\mathcal{V})$ を破ろうとする敵 A がタグ生成オラクルにメッセージ $M_1, \ldots, M_j$ を質問し，その答え $T_1, \ldots, T_j$ 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．

$MAC\text{-}\mathcal{V}_K(M_{j+1}, T_{j+1}) = \text{accept}$ であり，$M_{j+1} \notin \{M_1, , \ldots, M_j\}$ 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．$\{M_1, \ldots, M_j\}$ 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 $MAC = (MAC\text{-}\mathcal{K}, MAC\text{-}\mathcal{G}, MAC\text{-}\mathcal{V})$ 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 $\mathbf{Adv}^{\mathrm{mac}}_{MAC}(A)$ によって評価される．ここで，

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

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 $MAC = (MAC\text{-}\mathcal{K}, MAC\text{-}\mathcal{G}, MAC\text{-}\mathcal{V})$ 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．

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

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．

4.2.4　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，そのつど説明をする．

5　CBC MAC とその変形

本章では，CBC MAC とその変形として，CBC MAC, EMAC, RMAC, XCBC, TMAC, OMAC について述べる．

5.1　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 $E : \mathcal{K}_E \times \{0,1\}^n \to \{0,1\}^n$ 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 CBC MAC を $\mathrm{CBC}[E, \tau, m]$ と表記する．$\mathrm{CBC}[E, \tau, m] = (\text{CBC-}\mathcal{K}, \text{CBC-}\mathcal{G}, \text{CBC-}\mathcal{V})$ 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，$K \xleftarrow{R} \mathcal{K}_E$ を出力する．
- タグ生成アルゴリズム CBC-$\mathcal{G} : \mathcal{K}_E \times \{0,1\}^{mn} \to \{0,1\}^\tau$ 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．Fig. 4, Fig. 5 にあるように動作する．Fig. 5 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 τ bits of $Y[m]$
return T

Fig. 4. CBC MAC のタグ生成アルゴリズム CBC-$\mathcal{G}_K(\cdot)$.

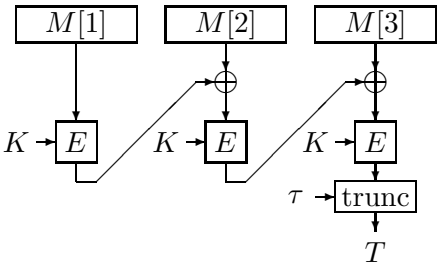

Fig. 5. $M = M[1]M[2]M[3]$ の場合の CBC-$\mathcal{G}_K(M)$ の動作.

- 確認アルゴリズム CBC-$\mathcal{V} : \mathcal{K}_E \times \{0,1\}^{mn} \times \{0,1\}^\tau \to$ 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，accept or reject = CBC-$\mathcal{V}_K(M, T)$ を出力する．Fig. 6 にあるように動作する．

安全性　Bellare, Kilian, Rogaway により，安全性が解析されている [7]. ブロック暗号 E が安全な擬似ランダム置換族であれば，$\mathrm{CBC}[E, \tau, m] = (\mathcal{K}_{\mathrm{CBC}}, \mathcal{G}_{\mathrm{CBC}}, \mathcal{V}_{\mathrm{CBC}})$ は，偽造不

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

Fig. 6. CBC MAC の確認アルゴリズム CBC-$\mathcal{V}_K(\cdot,\cdot)$.

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．) 以下の定理が示されている．

定理 5.1 $n, m \geq 1$ を整数，t, q を $qm \leq 2^{(n+1)/2}$ なる整数とする．$E : \mathcal{K}_E \times \{0,1\}^n \to \{0,1\}^n$ をブロック暗号とする．このとき，

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

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 5.1 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，

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

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，

$$\mathbf{Adv}^{\mathrm{prp}}_{E}(B) \geq \epsilon - \frac{2q^2m^2}{2^n} - \frac{1}{2^\tau}$$

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 ではなくなる．たとえば，図 7 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 $A^{\text{CBC-}\mathcal{G}_K(\cdot)}$
$M \leftarrow 0^n$
$T \leftarrow$ CBC-$\mathcal{G}_K(M)$
$M' \leftarrow (M, T)$
return (M', T)

Fig. 7. $A^{\text{CBC-}\mathcal{G}_K(\cdot)}$.

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．

5.2 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 $E : \mathcal{K}_E \times \{0,1\}^n \to \{0,1\}^n$ 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 EMAC$[E,\tau]$ と表記する．EMAC$[E,\tau]$ = (EMAC-$\mathcal{K}$, EMAC-$\mathcal{G}$, EMAC-$\mathcal{V}$) の鍵生成アルゴリズム EMAC-$\mathcal{K}$, タグ生成アルゴリズム EMAC-$\mathcal{G}$, 確認アルゴリズム 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 EMAC-$\mathcal{K}$ は確率的アルゴリズムであり，$K_1 \xleftarrow{R} \mathcal{K}_E$ と $K_2 \xleftarrow{R} \mathcal{K}_E$ を出力する．
- タグ生成アルゴリズム EMAC-$\mathcal{G} : (\mathcal{K}_E)^2 \times (\{0,1\}^n)^+ \to \{0,1\}^\tau$ 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 $T =$ EMAC-$\mathcal{G}_{K_1,K_2}(M) \in \{0,1\}^\tau$ を出力する．Fig. 8, Fig. 9 にあるように動作する．
- 確認アルゴリズム EMAC-$\mathcal{V} : (\mathcal{K}_E)^2 \times (\{0,1\}^n)^+ \times \{0,1\}^\tau \to$ 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，accept or reject = EMAC-$\mathcal{V}_{K_1,K_2}(M,T)$ を出力する．Fig. 10 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 [42]. ブロック暗号 E が安全な擬似ランダム置換族であれば，EMAC$[E,\tau]$ = (EMAC-$\mathcal{K}$, EMAC-$\mathcal{G}$, EMAC-$\mathcal{V}$) 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 τ bits of $E_{K_2}(Y[m])$
return T

Fig. 8. EMAC のタグ生成アルゴリズム EMAC-$\mathcal{G}_{K_1,K_2}(\cdot)$.

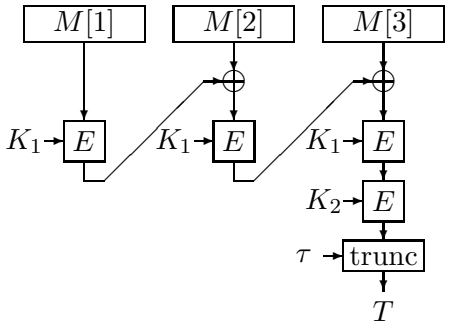

Fig. 9. $M = M[1]M[2]M[3]$ の場合の EMAC-$\mathcal{G}_{K_1,K_2}(M)$ の動作.

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

Fig. 10. EMAC の確認アルゴリズム EMAC-$\mathcal{V}_{K_1,K_2}(\cdot,\cdot)$.

定理 5.2 $n, \tau \geq 1$ を整数, $t, q, \sigma \geq 1$ を $\sigma^2 \leq 2^{(n+1)/2}$ なる整数とする．$E : \mathcal{K}_E \times \{0,1\}^n \to \{0,1\}^n$ をブロック暗号とする．このとき,

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

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 5.2 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,

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

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,

$$\mathbf{Adv}^{\mathrm{prp}}_{E}(B) \geq \frac{\epsilon}{2} - \frac{3\sigma^2}{2^{n+1}} - \frac{1}{2^{\tau}}$$

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 [25]．また，NESSIE の portfolio にも含まれている [40]．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れている [40].

5.3 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 [29, 30] と NIST が SP800-38B のドラフト版 [48] 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 [31] には 2 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．

$\mathrm{Perm}(n)$ 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，$\mathrm{Perm}(n)$ から一様ランダムに選んだ f_1 である．$f_1 \xleftarrow{R} \mathrm{Perm}(n)$ と表記する．置換族 F_2 を以下のように定義する．

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

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 $f_2^{(R)} \in \mathrm{Perm}(n)$ 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，$f_1 \xleftarrow{R} \mathrm{Perm}(n)$ と，各 $R \in \{0,1\}^r$ に対し，$f_2(R) \xleftarrow{R} \mathrm{Perm}(n)$ を出力する．

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

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

  となる．

- タグ生成アルゴリズム RMAC-JJV1-$\mathcal{G}$: $(\mathrm{Perm}(n) \times F_2) \times \{0,1\}^* \to (\{0,1\}^n \times \{0,1\}^r)$ は確率的アルゴリズムであり，鍵空間は $\mathrm{Perm}(n) \times F_2$, 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 $f_1 \in \mathrm{Perm}(n)$, $f_2^{(R)}$ $(R \in \{0,1\}^r)$ 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．Fig. 11, Fig. 12 にあるように動作する．

  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 RMAC-JJV1-$\mathcal{V}$: $(\mathrm{Perm}(n) \times F_2) \times \{0,1\}^* \times (\{0,1\}^n \times \{0,1\}^r) \to$ accept or reject は決定的アルゴリズムであり，鍵 $f_1 \in \mathrm{Perm}(n)$, $f_2^{(\cdot)} \in F_2$, 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，accept or reject $=$ $\text{RMAC-JJV1-}\mathcal{V}_{f_1, f_2^{(R)}}(M, T)$ を出力する．Fig. 13 にあるように動作する．

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 $f'^{(R)}_2$ を用いる．

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

Fig. 11. RMAC-JJV1 のタグ生成アルゴリズム RMAC-JJV1-$\mathcal{G}_{f_1,f_2^{(\cdot)}}(\cdot)$.

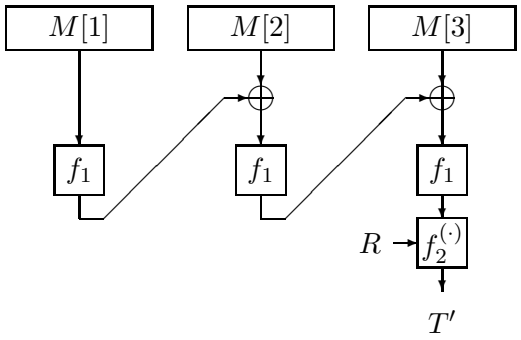

Fig. 12. $M = M[1]M[2]M[3]$ の場合の RMAC-JJV1-$\mathcal{G}_{f_1,f_2^{(R)}}(M)$ の動作.

Algorithm 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

Fig. 13. RMAC-JJV1 の確認アルゴリズム RMAC-JJV1-$\mathcal{V}_{f_1,f_2^{(\cdot)}}(\cdot,\cdot)$.

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，$\mathrm{AES}_{K_2 \oplus R}$ (K_2 が 128 ビットの場合)，もしくは $\mathrm{AES}_{K_2 \oplus (R \| 0^{128})}$ (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，$\mathrm{AES}_{K_2 \oplus R}$, $R = (R' \| 1 \| 0^{63})$ (K_2 が 192 ビットの場合) もしくは $\mathrm{AES}_{K_2 \oplus R}$, $R = (R' \| 1 \| 0^{127})$ (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，$f'^{(R)}_2$ として，$\mathrm{AES}_{K_2 \oplus R}$, $R = (R' \| 0 \| 0^{63})$ (K_2 が 192 ビットの場合) もしくは $\mathrm{AES}_{K_2 \oplus R}$, $R = (R' \| 0 \| 0^{127})$ (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 $E : \{0,1\}^k \times \{0,1\}n \to \{0,1\}^n$, 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，RMAC-NIST$[E, \tau, r]$ と表記する．

RMAC-NIST$[E, \tau, r]$ = (RMAC-NIST-$\mathcal{K}$, RMAC-NIST-$\mathcal{G}$, 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，$K_1, K_2 \xleftarrow{R} \{0,1\}^k$ を出力する．
- タグ生成アルゴリズム RMAC-NIST-$\mathcal{G}$: $(\{0,1\}^k)^2 \times \{0,1\}^r \times \{0,1\}^* \to \{0,1\}^r \times \{0,1\}^\tau$) 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 = RMAC-NIST-$\mathcal{G}_{K_1,K_2}(R, M) \in \{0,1\}^r \times \{0,1\}^\tau$ を出力する．Fig. 14, Fig. 15 にあるように動作する．

  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．

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 $K_3 \leftarrow K_2 \oplus (R \| 0^{k-r})$
$T' \leftarrow$ the left most τ bits of $E_{K_2}(Y[m])$
$T \leftarrow (R, T')$
return T

Fig. 14. RMAC-NIST のタグ生成アルゴリズム RMAC-NIST-$\mathcal{G}_{K_1,K_2}(\cdot,\cdot)$.

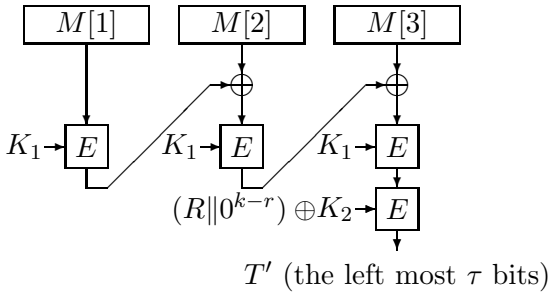

Fig. 15. $M = M[1]M[2]M[3]$ の場合の RMAC-NIST-$\mathcal{G}_{K_1,K_2}(R, M)$ の動作.

- 確認アルゴリズム RMAC-NIST-$\mathcal{V}$: $(\{0,1\}^k)^2 \times \{0,1\}^* \times (\{0,1\}^r \times \{0,1\}^\tau) \to$ 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．Fig. 16 にあるように動作する．

Algorithm 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 $K_3 \leftarrow K_2 \oplus (R \| 0^{k-r})$
$T'' \leftarrow$ the left most τ bits of $E_{K_2}(Y[m])$
if $T' = T''$ **then return** accept
    else return reject

Fig. 16. RMAC-NIST の確認アルゴリズム 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 $E : \{0,1\}^k \times \{0,1\}n \to \{0,1\}^n$, 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，Fig. 17 を提案している．

Parameter Set	$n = 128$		$n = 64$	
	r	n	r	n
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	

Fig. 17. RMAC-NIST のパラメータ.

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 [29, 30]．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ければならない点が異なる．

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 $M_1, \ldots, M_q$ を質問し，その答え $T_1, \ldots, T_q$ 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 $(M'_1, T'_1), \ldots, (M'_{q'}, T'_{q'})$ 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，$T'_i \notin \{T_1, \ldots, T_j\}$ であれば，A は偽造に成功した，という．$\{T_1, \ldots, T_j\}$ 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 $\mathbf{Adv}^{\text{rmac-uf}}(A)$ を以下のように定義する．

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

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

以下の定理が示されている [29, 30].

定理 5.3 $n \geq 2$ 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，

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

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 [29, 30] が，該当箇所には議論の不備が指摘されている [45]．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．[45] では帰着することは不可能である，と述べられている．

また，NESSIE でも考慮されたが，最終的には portfolio には含まれなかった [40].

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 [48] 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 [35]．Rogaway [45], Wagner [50], Black [11] 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た [41]．

5.4 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 $E : \mathcal{K}_E \times \{0,1\}^n \to \{0,1\}^n$ 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 XCBC$[E,\tau]$ と表記する．XCBC$[E,\tau]$ = (XCBC-$\mathcal{K}$, XCBC-$\mathcal{G}$, XCBC-$\mathcal{V}$) の鍵生成アルゴリズム XCBC-$\mathcal{K}$, タグ生成アルゴリズム XCBC-$\mathcal{G}$, 確認アルゴリズム 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 XCBC-$\mathcal{K}$ は確率的アルゴリズムであり，$K_1 \xleftarrow{R} \mathcal{K}_E$, $K_2 \xleftarrow{R} \{0,1\}^n$, $K_3 \xleftarrow{R} \{0,1\}^n$ を出力する．
- タグ生成アルゴリズム XCBC-$\mathcal{G} : (\mathcal{K}_E \times (\{0,1\}^n)^2) \times \{0,1\}^* \to \{0,1\}^\tau$ 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．Fig. 18, Fig. 19 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．Fig. 18 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．

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**
    $X[i] \leftarrow M[i] \oplus Y[i-1]$
    $Y[i] \leftarrow E_{K_1}(X[i])$
$X[m] \leftarrow \mathbf{pad}_n(M[m]) \oplus Y[m-1]$
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 τ bits of $E_{K_1}(Y[m])$
return T

Fig. 18. XCBC のタグ生成アルゴリズム XCBC-$\mathcal{G}_{K_1,K_2,K_3}(\cdot)$.

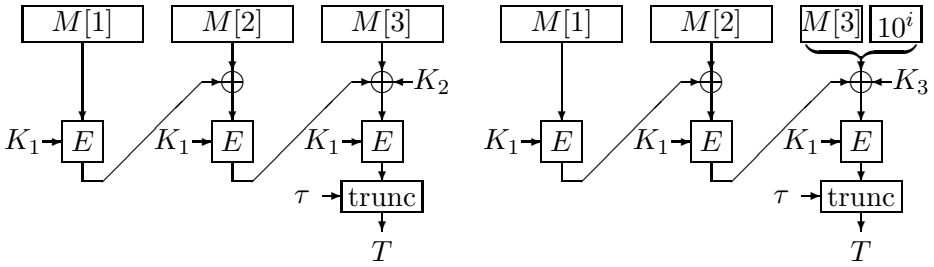

Fig. 19. $M = M[1]M[2]M[3]$ の場合の XCBC-$\mathcal{G}_{K_1,K_2,K_3}(M)$ の動作.

また，Fig. 18 の 7 行目の関数 $\mathbf{pad}_n : \{0,1\}^{\leq n} \to \{0,1\}^n$ 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)．ことのき，

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

- 確認アルゴリズム XCBC-$\mathcal{V} : (\mathcal{K}_E \times (\{0,1\}^n)^2) \times \{0,1\}^* \times \{0,1\}^\tau \to$ 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，accept or reject = XCBC-$\mathcal{V}_{K_1,K_2,K_3}(M,T)$ を出力する．Fig. 20 にあるように動作する．

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

Fig. 20. XCBC の確認アルゴリズム 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 [12]. ブロック暗号 E が安全な擬似ランダム置換族であれば，XCBC$[E,\tau]$ = (XCBC-$\mathcal{K}$, XCBC-$\mathcal{G}$, XCBC-$\mathcal{V}$) 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 [27].

定理 5.4 $n, \tau \geq 1$ を整数, $t, q, \sigma \geq 1$ を $\sigma^2 \leq 2^{(n+1)/2}$ なる整数とする．$E : \mathcal{K}_E \times \{0,1\}^n \to \{0,1\}^n$ をブロック暗号とする．このとき，

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

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 5.4 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，

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

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，

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

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 [41]．また，[19] や [23] で議論されている．

5.5　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 $E : \mathcal{K}_E \times \{0,1\}^n \to \{0,1\}^n$ 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 TMAC$[E, \tau]$ と表記する．TMAC$[E, \tau] =$ (TMAC-$\mathcal{K}$, TMAC-$\mathcal{G}$, 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，$K_1 \xleftarrow{R} \mathcal{K}_E$ と $K_2 \xleftarrow{R} \{0,1\}^n$ を出力する．
- タグ生成アルゴリズム TMAC-$\mathcal{G} : (\mathcal{K}_E \times \{0,1\}^n) \times \{0,1\}^* \to \{0,1\}^\tau$ 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 $T =$ TMAC-$\mathcal{G}_{K_1,K_2}(M) \in \{0,1\}^\tau$ を出力する．Fig. 21, Fig. 22 にあるように動作する．Fig. 21, Fig. 22 において，$K_2 \cdot \mathtt{u}$ は，GF(2^n) 上の乗算である．一般的に $a \in \{0,1\}^n$ に対し，

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

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，$a = a_{n-1}a_{n-2}\cdots a_1 a_0$ と a をビット表現した場合，$a \ll 1 = a_{n-2}a_{n-3}\cdots a_0 0$ 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，$\mathtt{msb}(a)$ は a の最上位ビットを表し，$\mathtt{Cst}_n$ は n ビットの定数である．たとえば，$\mathtt{Cst}_{128} = 0^{120}10000111$ であり，$\mathtt{Cst}_{64} = 0^{59}11011$ である．TMAC も XCBC と同様，M の長さが n の倍数でなくてもよい．Fig. 21 の 3 行目において，

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

Algorithm 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**
    $X[i] \leftarrow M[i] \oplus Y[i-1]$
    $Y[i] \leftarrow E_{K_1}(X[i])$
$X[m] \leftarrow \mathbf{pad}_n(M[m]) \oplus Y[m-1]$
if $|M[m]| = n$ **then** $X[m] \leftarrow X[m] \oplus K_2 \cdot \mathtt{u}$
    else $X[m] \leftarrow X[m] \oplus K_2$
$T \leftarrow$ the left most τ bits of $E_{K_1}(Y[m])$
return T

Fig. 21. TMAC のタグ生成アルゴリズム TMAC-$\mathcal{G}_{K_1,K_2}(\cdot)$.

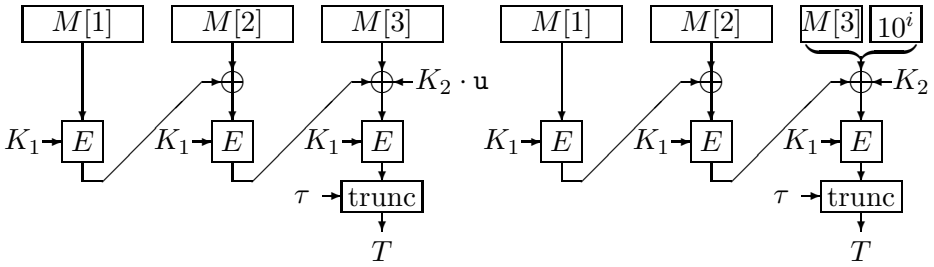

Fig. 22. $M = M[1]M[2]M[3]$ の場合の TMAC-$\mathcal{G}_{K_1,K_2}(M)$ の動作.

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

また，Fig. 21 の 7 行目の関数 $\mathbf{pad}_n : \{0,1\}^{\le n} \to \{0,1\}^n$ は (1) のように定義される．

- 確認アルゴリズム TMAC-$\mathcal{V}$: $(\mathcal{K}_E \times \{0,1\}^n) \times \{0,1\}^* \times \{0,1\}^\tau \to$ 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，accept or reject = TMAC-$\mathcal{V}_{K_1,K_2}(M,T)$ を出力する．Fig. 23 にあるように動作する．

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

Fig. 23. TMAC の確認アルゴリズム 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 [36]. ブロック暗号 E が安全な擬似ランダム置換族であれば，TMAC$[E,\tau]$ = (TMAC-$\mathcal{K}$, TMAC-$\mathcal{G}$, TMAC-$\mathcal{V}$) 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 [27].

定理 5.5 $n, \tau \ge 1$ を整数, $t, q, \sigma \ge 1$ を $\sigma^2 \le 2^{(n+1)/2}$ なる整数とする．$E : \mathcal{K}_E \times \{0,1\}^n \to \{0,1\}^n$ をブロック暗号とする．このとき,

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

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 5.5 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,

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

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,

$$\mathbf{Adv}^{\mathrm{prp}}_{E}(B) \ge \epsilon - \frac{3\sigma^2}{2^n} - \frac{1}{2^\tau}$$

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 [41].

5.6　OMAC

OMAC は 2 つの方式 OMAC1 と OMAC2 の総称である．

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 $E : \mathcal{K}_E \times \{0,1\}^n \to \{0,1\}^n$ 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 OMAC1$[E, \tau]$ と表記する．OMAC1$[E, \tau] =$ (OMAC1-$\mathcal{K}$, OMAC1-$\mathcal{G}$, 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 OMAC1-$\mathcal{K}$ 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 OMAC1-$\mathcal{G} : \mathcal{K}_E \times \{0,1\}^* \to \{0,1\}^\tau$ 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 $T =$ OMAC1-$\mathcal{G}_K(M) \in \{0,1\}^\tau$ を出力する．Fig. 24, Fig. 25 にあるように動作する．Fig. 24, Fig. 25 において，$L \cdot \mathrm{u}$ は (2) によって得られ，$L \cdot \mathrm{u}^2$ は $(L \cdot \mathrm{u}) \cdot \mathrm{u}$ 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．Fig. 24 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．$M = \varepsilon$ のときは例外でり，この場合，$|M[m]| = 0$ となる．

  また，Fig. 24 の 7 行目の関数 $\mathbf{pad}_n : \{0,1\}^{\leq n} \to \{0,1\}^n$ は (1) のように定義される．

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**
    $X[i] \leftarrow M[i] \oplus Y[i-1]$
    $Y[i] \leftarrow E_K(X[i])$
$X[m] \leftarrow \mathbf{pad}_n(M[m]) \oplus Y[m-1]$
if $|M[m]| = n$ **then** $X[m] \leftarrow X[m] \oplus L \cdot \mathtt{u}$
else $X[m] \leftarrow X[m] \oplus L \cdot \mathtt{u}^2$
$T \leftarrow$ the left most τ bits of $E_K(Y[m])$
return T

Fig. 24. OMAC1 のタグ生成アルゴリズム OMAC1-$\mathcal{G}_K(\cdot)$.

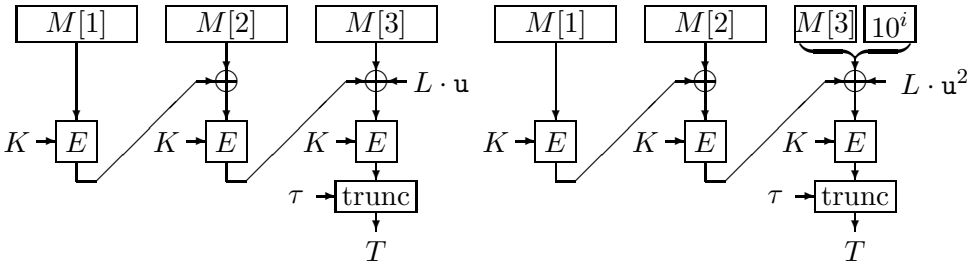

Fig. 25. $M = M[1]M[2]M[3]$ の場合の OMAC1-$\mathcal{G}_K(M)$ の動作.

- 確認アルゴリズム OMAC1-$\mathcal{V}$: $\mathcal{K}_E \times \{0,1\}^* \times \{0,1\}^\tau \to$ 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，accept or reject = OMAC1-$\mathcal{V}_K(M,T)$ を出力する．Fig. 26 にあるように動作する．

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

Fig. 26. OMAC1 の確認アルゴリズム OMAC1-$\mathcal{V}_K(\cdot,\cdot)$.

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

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

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，$a \gg 1 = 0a_{n-1}a_{n-2}\cdots a_2a_1$ 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，$\mathtt{lsb}(a)$ は a の最下位ビットを表し，$\mathtt{Cst}'_n$ は n ビットの定数である．たとえば，$\mathtt{Cst}'_{128} = 10^{120}1000011$ である．

安全性 Iwata, Kurosawa により，安全性が解析されている [26]. ブロック暗号 E が安全な擬似ランダム置換族であれば，OMAC1$[E,\tau]$ = (OMAC1-$\mathcal{K}$, OMAC1-$\mathcal{G}$, OMAC1-$\mathcal{V}$) と OMAC2$[E,\tau]$ = (OMAC2-$\mathcal{K}$, OMAC2-$\mathcal{G}$, OMAC2-$\mathcal{V}$) 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 [27].

定理 5.6 $n, \tau \geq 1$ を整数, $t, q, \sigma \geq 1$ を $\sigma^2 \leq 2^{(n+1)/2}$ なる整数とする．$E : \mathcal{K}_E \times \{0,1\}^n \to \{0,1\}^n$ をブロック暗号とする．このとき，

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

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 5.6 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，

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

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，

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

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 に提案されている [41]．NIST は 2003 年 6 月，RMAC の決定を見直し，OMAC を提案すると発表した [41]．2003 年 11 月現在，OMAC1 を提案予定である，としている．

6　並列計算可能な MAC

本章では，並列計算可能な MAC として，XOR MAC, XECB MAC, PMAC について述べる．CBC MAC とその変形では，ブロック暗号の並列計算ができないのに対し，本章で挙げる MAC では，ブロック暗号の並列計算が可能である．

6.1　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 $F : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^{n'}$ 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，

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

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

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

ただし，$M = M[1] \cdots M[m]$ はパディングされていて，各 $M[1], \ldots, M[m]$ 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，$\mathbf{tag}(K, M, r)$ を $\mathbf{tag}_K(M, r)$ と表記する．Fig. 27 参照.

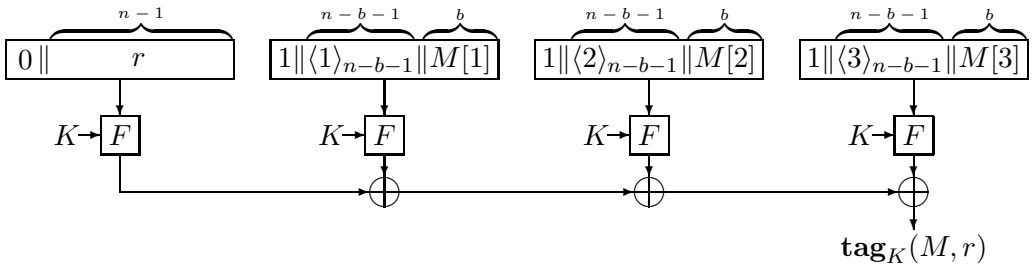

Fig. 27. $M = M[1]M[2]M[3]$ の場合の $\mathbf{tag}_K(M, r)$ の動作.

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 $F : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^{n'}$ 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 XMACR$[F, b]$ と表記する．XMACR$[F, b] =$ (XMACR-$\mathcal{K}$, XMACR-$\mathcal{G}$, 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，$K \xleftarrow{R} \{0,1\}^k$ を出力する．
- タグ生成アルゴリズム XMACR-$\mathcal{G} : \{0,1\}^k \times \{0,1\}^* \to (\{0,1\}^{n-1} \times \{0,1\}^{n'})$ 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 $T =$ XMACR-$\mathcal{G}_K(M) \in (\{0,1\}^{n-1} \times \{0,1\}^{n'})$ を出力する．Fig. 28 にあるように動作する．

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

Fig. 28. 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 XMACR-$\mathcal{V}$: $\{0,1\}^k \times \{0,1\}^* \times (\{0,1\}^{n-1} \times \{0,1\}^{n'}) \to$ 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，accept or reject = XMACR-$\mathcal{V}_K(M,T)$ を出力する．Fig. 29 にあるように動作する．

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

Fig. 29. 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 $F : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^{n'}$ 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．XMACC$[F,b]$ = (XMACC-$\mathcal{K}$, XMACC-$\mathcal{G}$, 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，$K \xleftarrow{R} \{0,1\}^k$ を出力する．
- タグ生成アルゴリズム XMACC-$\mathcal{G} : \{0,1\}^k \times \{0,1\}^* \times \mathsf{Count} \to (\mathsf{Count} \times \{0,1\}^{n'})$ 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 $(\mathsf{Count} \times \{0,1\}^{n'})$ である．さらに入力として，カウンタ $C \in \mathsf{Count}$ をとる．Count はカウンタの空間であり，$\mathsf{Count} = \{0,1,\ldots,2^{n-1}-1\}$ 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 $C \in \mathsf{Count}$ を入力とし，タグ $T =$ XMACC-$\mathcal{G}_K(M) \in (\mathsf{Count} \times \{0,1\}^{n'})$ を出力する．Fig. 30 にあるように動作する．

  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-$\mathcal{V} : \{0,1\}^k \times \{0,1\}^* \times (\mathsf{Count} \times \{0,1\}^{n'}) \to$ 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 $T \in (\mathsf{Count} \times \{0,1\}^{n'})$ を入力とし，accept or reject = XMACC-$\mathcal{V}_K(M,T)$ を出力する．Fig. 31 にあるように動作する．

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

Fig. 30. XMACC のタグ生成アルゴリズム XMACC-$\mathcal{G}_K(\cdot)$.

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

Fig. 31. 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 [6]. 関数族 F が安全な擬似ランダム関数族であれば，XMACR$[F, b]$ = (XMACR-$\mathcal{K}$, XMACR-$\mathcal{G}$, XMACR-$\mathcal{V}$) と XMACC$[F, b]$ = (XMACC-$\mathcal{K}$, XMACC-$\mathcal{G}$, XMACC-$\mathcal{V}$) 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 $F : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^{n'}$ 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 $\{F_K(\cdot) \in \mathrm{Rand}(n, n') \mid K \in \{0,1\}^k\}$ と $\{0,1\}^n$ から $\{0,1\}^{n'}$ へのすべての関数の集合 $\mathrm{Rand}(n, n')$ 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 $F : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^{n'}$ 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 $\mathbf{Adv}^{\mathrm{prf}}_F(A)$ によって評価される．ここで，

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

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．

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

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

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

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 [6].

定理 6.1 $n, n' \geq 1$ 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．$F : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^{n'}$ を関数族とする．このとき，

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

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 6.1 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，

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

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，

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

なる敵 B が存在する．

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

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

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 [6].

定理 6.2 $n, n' \geq 1$ 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．$F : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^{n'}$ を関数族とする．このとき，

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

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 6.2 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，

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

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，

$$\mathbf{Adv}^{\text{prp}}_{E}(B) \geq \epsilon - \frac{2q_v^2}{2^{n'}}$$

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．

6.2　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 $E : \mathcal{K}_E \times \{0,1\}^n \to \{0,1\}^n$ 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 を XECB$-MAC[E] と表記する．XECB$-MAC[$E$] $=$ (XECB$-$\mathcal{K}$, XECB$-$\mathcal{G}$, 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，$K \xleftarrow{R} \mathcal{K}_E$ を出力する．
- タグ生成アルゴリズム XECB$-$\mathcal{G}$: $\mathcal{K}_E \times \{0,1\}^* \to (\{0,1\}^n \times \{0,1\}^n)$ 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 $T =$ XECB$-$\mathcal{G}_K(M) \in (\{0,1\}^n \times \{0,1\}^n)$ を出力する．Fig. 32 にあるように動作する．

Algorithm 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** $Z \leftarrow \overline{z_0}$
else $Z \leftarrow z_0$
$M[m] \leftarrow \mathbf{pad}_n(M[m])$
$M[m+1] \leftarrow Z$
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

Fig. 32. XECB$-MAC のタグ生成アルゴリズム XECB$-$\mathcal{G}_K(\cdot)$.

ただし，$+$ や $\times$ の演算は $\bmod 2^n$ 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 $\overline{z_0}$ は z_0 のビットごとの反転である．8 行目の $\mathbf{pad}_n(\cdot)$ は (1) で定義されている．

- 確認アルゴリズム XECB$-$\mathcal{V}$: $\mathcal{K}_E \times \{0,1\}^* \times (\{0,1\}^n \times \{0,1\}^n) \to$ 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，accept or reject $=$ XECB$-$\mathcal{V}_K(M,T)$ を出力する．Fig. 33 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 accpet 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** $Z \leftarrow \overline{z_0}$
else $Z \leftarrow z_0$
$M[m] \leftarrow \mathbf{pad}_n(M[m])$
$M[m+1] \leftarrow Z$
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

Fig. 33. 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．XECBC-MAC$[E]$ = (XECBC-$\mathcal{K}$, XECBC-$\mathcal{G}$, 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，$K \xleftarrow{R} \mathcal{K}_E$ を出力する．
- タグ生成アルゴリズム XECBC-$\mathcal{G} : \mathcal{K}_E \times \{0,1\}^* \times \mathsf{Count} \to (\mathsf{Count} \times \{0,1\}^n)$ 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 $(\mathsf{Count} \times \{0,1\}^n)$ である．さらに入力として，カウンタ $C \in \mathsf{Count}$ をとる．Count はカウンタの空間であり，$\mathsf{Count} = \{1, 2, \ldots\}$ 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 $C \in \mathsf{Count}$ を入力とし，タグ $T = \text{XECBC-}\mathcal{G}_K(M) \in (\mathsf{Count} \times \{0,1\}^n)$ を出力する．Fig. 34 にあるように動作する．

  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，$\bmod 2^n$ 上で行われる．
- 確認アルゴリズム XECBC-$\mathcal{V} : \mathcal{K}_E \times \{0,1\}^* \times (\mathsf{Count} \times \{0,1\}^n) \to$ 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 $T \in (\mathsf{Count} \times \{0,1\}^n)$ を入力とし，accept or reject = XMACC-$\mathcal{V}_K(M, T)$ を出力する．Fig. 35 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** $Z \leftarrow \overline{z_0}$
else $Z \leftarrow z_0$
$M[m] \leftarrow \mathbf{pad}_n(M[m])$
$M[m+1] \leftarrow Z$
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

Fig. 34. XECBC-MAC のタグ生成アルゴリズム XECBC-$\mathcal{G}_K(\cdot)$.

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** $Z \leftarrow \overline{z_0}$
else $Z \leftarrow z_0$
$M[m] \leftarrow \mathbf{pad}_n(M[m])$
$M[m+1] \leftarrow Z$
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

Fig. 35. 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，$K \xleftarrow{R} \mathcal{K}_E$ を出力する．
- タグ生成アルゴリズム XECBS-$\mathcal{G} : \mathcal{K}_E \times \{0,1\}^* \times \mathsf{Count} \to (\mathsf{Count} \times \{0,1\}^n)$ 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 $(\mathsf{Count} \times \{0,1\}^n)$ である．さらに入力として，カウンタ $C \in \mathsf{Count}$ をとる．Count はカウンタの空間であり，$\mathsf{Count} = \{1, 2, \ldots\}$ 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 $C \in \mathsf{Count}$ を入力とし，タグ $T = \text{XECBS-}\mathcal{G}_K(M) \in (\mathsf{Count} \times \{0,1\}^n)$ 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 [20]．Fig. 36 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** $Z \leftarrow \overline{R}$
else $Z \leftarrow R$
$M[m] \leftarrow \mathbf{pad}_n(M[m])$
$M[m+1] \leftarrow Z$
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

Fig. 36. 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，$\bmod 2^n$ 上で行われる．

- 確認アルゴリズム XECBS-$\mathcal{V} : \mathcal{K}_E \times \{0,1\}^* \times (\mathsf{Count} \times \{0,1\}^n) \to$ 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 $T \in (\mathsf{Count} \times \{0,1\}^n)$ を入力とし，accept or reject = 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．Fig. 37 にあるように動作する．

Algorithm XMACC-$\mathcal{V}_K(M,(C,T'))$
Partition M into $M[1]\cdots M[m]$
if $|M[m]|=n$ **then** $Z \leftarrow \overline{R}$
else $Z \leftarrow R$
$M[m] \leftarrow \mathbf{pad}_n(M[m])$
$M[m+1] \leftarrow Z$
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

Fig. 37. XECBS の確認アルゴリズム XECBS-$\mathcal{V}_K(\cdot,\cdot)$.

安全性　Glogor, Donescu により，安全性が解析されている [20]. ブロック暗号 E が安全な擬似ランダム置換族であれば，XECB$-MAC$[E]$ = (XECB$-$\mathcal{K}$, XECB$-$\mathcal{G}$, XECB$-$\mathcal{V}$), XECBC-MAC$[E]$ = (XECBC-$\mathcal{K}$, XECBC-$\mathcal{G}$, XECBC-$\mathcal{V}$), および，XECBS-MAC$[E]$ = (XECBS-$\mathcal{K}$, XECBS-$\mathcal{G}$, XECBS-$\mathcal{V}$) 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．

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

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

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．

XECB$-MAC については，以下の定理が示されている [20].

定理 6.3 $n \geq 1$ を整数，$t,q_g,\mu_g,q_v,\mu_g \geq 1$ を整数とする．$E : \mathcal{K}_E \times \{0,1\}^n \to \{0,1\}^{n'}$ をブロック暗号とする．このとき，

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

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 $\mathbf{Adv}^{\text{s-uf}}_{\text{XECBS-MAC}[E]}(t, q_g, \mu_g, q_v, \mu_v)$ も XECB$-MAC[$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 [20].

定理 6.4 $n \geq 1$ を整数，$t, q_g, \mu_g, q_v, \mu_g \geq 1$ を整数とする．$E : \mathcal{K}_E \times \{0,1\}^n \to \{0,1\}^{n'}$ をブロック暗号とする．このとき，

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

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

XECBC-MAC の安全性 $\mathbf{Adv}^{\text{s-uf}}_{\text{XECBC-MAC}[E]}(t, q_g, \mu_g, q_v, \mu_v)$ 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 [20].

定理 6.5 $n \geq 1$ を整数，$t, q_g, \mu_g, q_v, \mu_g \geq 1$ を整数とする．$E : \mathcal{K}_E \times \{0,1\}^n \to \{0,1\}^{n'}$ をブロック暗号とする．このとき，

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

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：XECB$-MAC と XECBC-MAC では，$\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．
- 事前計算するべき 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 [41].

6.3 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 $E : \mathcal{K}_E \times \{0,1\}^n \to \{0,1\}^n$ 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 PMAC$[E,\tau]$ と表記する．PMAC$[E,\tau]$ = (PMAC-$\mathcal{K}$, PMAC-$\mathcal{G}$, PMAC-$\mathcal{V}$) の鍵生成アルゴリズム PMAC-$\mathcal{K}$, タグ生成アルゴリズム PMAC-$\mathcal{G}$, 確認アルゴリズム 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 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 PMAC-$\mathcal{G} : \mathcal{K}_E \times \{0,1\}^* \to \{0,1\}^\tau$ 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 $T =$ PMAC-$\mathcal{G}_K(M) \in \{0,1\}^\tau$ を出力する．Fig. 38, Fig. 39 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．Fig. 38 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．

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

ここで，$\mathbf{ntz}(i)$ 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，$\mathbf{ntz}(i)$ は，2^z が i を割り切る最大の z である．たとえば，$\mathbf{ntz}(7) = 0$, $\mathbf{ntz}(8) = 3$ 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，Fig. 38 の 7 行目の関数 $\mathbf{pad}_n : \{0,1\}^{\leq n} \to \{0,1\}^n$ は (1) と同様である．8 行目の $L \cdot \mathtt{u}^{-1}$ は，(3) のように計算される．

- 確認アルゴリズム PMAC-$\mathcal{V} : \mathcal{K}_E \times \{0,1\}^* \times \{0,1\}^\tau \to$ 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，accept or reject = PMAC-$\mathcal{V}_K(M,T)$ を出力する．Fig. 40 にあるように動作する．

安全性　Black, Rogaway により，安全性が解析されている [14]. ブロック暗号 E が安全な擬似ランダム置換族であれば，PMAC$[E,\tau]$ = (PMAC-$\mathcal{K}$, PMAC-$\mathcal{G}$, PMAC-$\mathcal{V}$) 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 [14].

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** $X[m] \leftarrow \Sigma \oplus L \cdot \mathtt{u}^{-1}$
    else $X[m] \leftarrow \Sigma$
$T \leftarrow$ the left most τ bits of $E_K(X[m])$
return T

Fig. 38. PMAC のタグ生成アルゴリズム PMAC-$\mathcal{G}_K(\cdot)$.

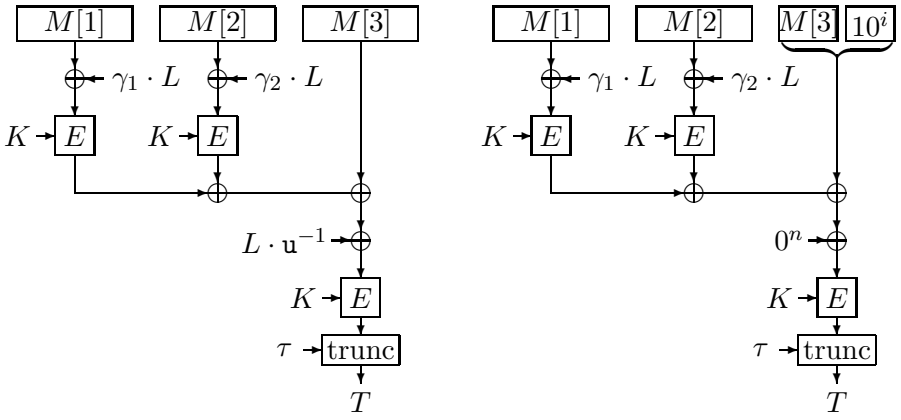

Fig. 39. $M = M[1]M[2]M[3]$ の場合の PMAC-$\mathcal{G}_K(M)$ の動作.

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

Fig. 40. PMAC の確認アルゴリズム PMAC-$\mathcal{V}_K(\cdot, \cdot)$.

定理 6.6 $n, \tau \geq 1$ を整数，$t, q, \sigma \geq 1$ を整数とする．$E : \mathcal{K}_E \times \{0,1\}^n \to \{0,1\}^n$ をブロック暗号とする．このとき，

$$\mathbf{Adv}^{\mathrm{mac}}_{\mathrm{PMAC}[E,\tau]}(t, q, \sigma) \leq \mathbf{Adv}^{\mathrm{prp}}_{E}(t', q') + \frac{1.5\sigma'^2}{2^n} + \frac{1}{2^\tau}$$

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 6.6 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，

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

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，

$$\mathbf{Adv}^{\mathrm{prp}}_{E}(B) \geq \epsilon - \frac{1.5\sigma'^2}{2^n} - \frac{1}{2^\tau}$$

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 [41].

7　その他の MAC

本章では，その他の MAC として，$f9$ について述べる．

7.1 $f9$

方式 $f9$ はブロック暗号 KASUMI : $\{0,1\}^{128} \times \{0,1\}^{64} \to \{0,1\}^{64}$ を用いる．鍵生成アルゴリズム $f9$-$\mathcal{K}$, タグ生成アルゴリズム $f9$-$\mathcal{G}$, 確認アルゴリズム $f9$-$\mathcal{V}$ はそれぞれ以下のように動作する．

- 鍵生成アルゴリズム $f9$-$\mathcal{K}$ は確率的アルゴリズムであり，$K \xleftarrow{R} \{0,1\}^{128}$ を出力する．
- タグ生成アルゴリズム $f9$-$\mathcal{G} : \mathcal{K}_E \times \{0,1\}^* \to \{0,1\}^\tau$ 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，送受信者の間で共有されている．Fig. 41, Fig. 42 にあるように動作する．

Algorithm $f9$-$\mathcal{G}_K(\mathrm{COUNT}, \mathrm{FRESH}, \mathrm{DIRECTION}, M)$
$M \leftarrow \mathtt{pad}_{64}(\mathrm{COUNT}, \mathrm{FRESH}, \mathrm{DIRECTION}, M)$
Break M into 64-bit blocks $M[1] \| \cdots \| M[m]$
$Y[0] \leftarrow 0^{64}$
For $i = 1$ to m do:
　　$X[i] \leftarrow M[i] \oplus Y[i-1]$
　　$Y[i] \leftarrow \mathrm{KASUMI}_K(X[i])$
$T \leftarrow \mathrm{KASUMI}_{K \oplus \mathtt{KM}}(Y[1] \oplus \cdots \oplus Y[m])$
$T \leftarrow$ the leftmost 32 bits of T
Return T

Fig. 41. $f9$ のタグ生成アルゴリズム $f9$-$\mathcal{G}_K(\cdot)$.

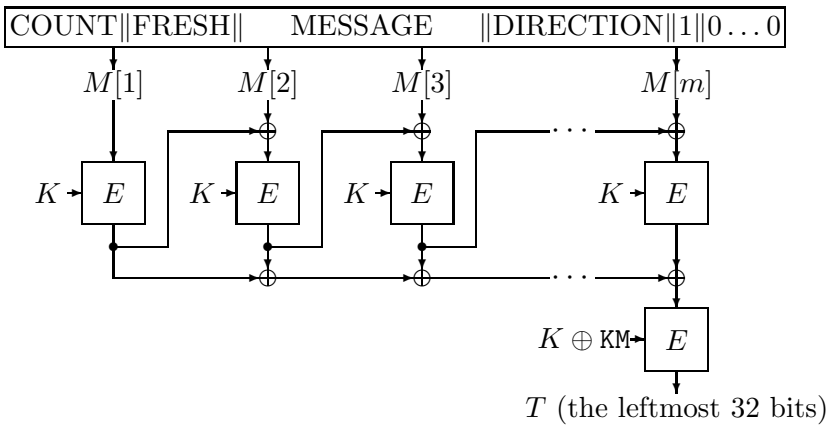

Fig. 42. $f9$-$\mathcal{G}_K(M)$ の動作．E は KASUMI である．

2 行目の $\mathtt{pad}_{64}(\mathrm{COUNT}, \mathrm{FRESH}, \mathrm{DIRECTION}, M)$ 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，

$$\begin{aligned}&\mathtt{pad}_{64}(\mathrm{COUNT}, \mathrm{FRESH}, \mathrm{DIRECTION}, M)\\ &\qquad = \mathrm{COUNT}\|\mathrm{FRESH}\|M\|\mathrm{DIRECTION}\|1\|0^{63-(|M|+1 \bmod 64)}\ .\end{aligned}$$

とする．KM は，128 ビットの定数であり，KM = 0xAA...AA である．

- 確認アルゴリズム $f9\text{-}\mathcal{V} : \{0,1\}^{64} \times \{0,1\}^* \times \{0,1\}^{32} \to$ 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，accept or reject $= f9\text{-}\mathcal{V}_K(M,T)$ を出力する．さらに，送受信者の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．

  Fig. 43 にあるように動作する．

Algorithm $f9\text{-}\mathcal{V}_K(\mathrm{COUNT}, \mathrm{FRESH}, \mathrm{DIRECTION}, M, T)$
$T' \leftarrow f9\text{-}\mathcal{G}_K(\mathrm{COUNT}, \mathrm{FRESH}, \mathrm{DIRECTION}, M)$
if $T = T'$ **then return** accept
else return reject

Fig. 43. $f9$ 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 [24] が，証明の不備が指摘されている [28].

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 $K \oplus \mathtt{KM}$ 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 [1, 2].

8 まとめ

本報告書では，メッセージ認証に関するブロック暗号利用モードについて，主に安全性の観点からその特徴をまとめるとともに，効率についても言及した．本報告書では，証明可能安全性に焦点をあてたが，実装に関する攻撃などはこの範疇ではない．そういった環境も考慮した新たな安全性の概念も次々に提案されている．また，新たなブロック暗号利用モードも数多く提案されている．NIST も含めた標準化作業も活動中であり，その動向を注視していく必要がある．また，ブロック暗号利用モードの選択に当たっては，安全性，効率，そして本報告書では触れなかった知的所有権の問題等を勘案し，十分な注意が必要である．

参考文献

[1] 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`.

[2] 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`.

[3] ANSI X3.106, American National Standard for Information Systems — Data Encryption Algorithm — Modes of Operation, American National Standard Institute, 1983.

[4] ANSI X3.92, American National Standard for Information Systems — Data Encryption Algorithm, American National Standard Institute, 1981.

[5] M. Bellare, A. Desai, E. Jokipii, 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.

[6] M. Bellare, R. Guérin, and P. Rogaway. XOR MACs: New methods for message authentication using finite pseudorandom functions. *Advances in Cryptology — CRYPTO '95, LNCS 963,* pp. 15–28, Springer-Verlag, 1995.

[7] 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.

[8] 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.

[9] M. Bellare, P. Rogaway, and D. Wagner. A conventional authenticated-encryption mode. NIST submission, 2003. Available at `http://csrc.nist.gov/CryotoToolKit/modes/proposedmodes/`.

[10] A. Berendschot, B. den Boer, J. P. Boly, A. Bosselaers, J. Brandt, D. Chaum, I. Damgård, M. Dichtl, W. Fumy, M. van der Ham, C. J. A. Jansen, P. Landrock, B. Preneel, G. Roelofsen, P. de Rooij, and J. Vandewalle. Final Report of RACE Integrity Primitives. *LNCS 1007,* Springer-Verlag, 1995.

[11] J. Black. Comments on the RMAC algorithm. Comments to NIST. Available at `http://csrc.nist.gov/CryptoToolkit/modes/comments/`.

[12] J. Black and P. Rogaway. CBC MACs for arbitrary-length messages: The three key constructions. *Advances in Cryptology — CRYPTO 2000, LNCS 1880,* pp. 197–215, Springer-Verlag, 2000.

[13] 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/`.

[14] J. Black and P. Rogaway. A block-cipher mode of operation for parallelizable message authentication. *Advances in Cryptology — EUROCRYPT 2002, LNCS 2332,* pp. 384–397, Springer-Verlag, 2002.

[15] National Institute of Standards and Technology, Federal Information Processing Standards Publication 46-3, Data Encryption Standard (DES).

[16] National Institute of Standards and Technology, Federal Information Processing Standards Publication 81, DES modes of operation (DES), 1980.

[17] National Institute of Standards and Technology, Federal Information Processing Standards Publication 113, Computer data authentication, 1994.

[18] National Institute of Standards and Technology, Federal Information Processing Standards Publication 197, Advanced Encryption Standard (AES), 2001.

[19] S. Frankel, and H. Herbert. The AES-XCBC-MAC-96 algorithm and its use with IPsec. Available at `http://www.ieft.org/`.

[20] V. G. Gligor, and P. Donescu. Fast encryption and authentication: XCBC encryption and XECB authentication modes. *Fast Software Encryption, FSE 2001, LNCS 2355,* pp. 92–108, Springer-Verlag, 2002.

[21] O. Goldreigh, S. Goldwasser and S. Micali. How to construct random functions. *J. ACM,* Vol. 33, No. 4, pp. 792–807, October 1986.

[22] S. Halevi, and P. Rogaway. A tweakable enciphering mode. *Advances in Cryptology — CRYPTO 2003, LNCS 2729,* pp. 482–499, Springer-Verlag, 2003.

[23] P. Hoffman. The AES-XCBC-PRF-128 algorithm for IKE. Available at `http://www.ieft.org/`.

[24] D. Hong, J-S. Kang, B. Preneel, and H. Ryu. A concrete security anallysis for 3GPP-MAC. *Fast Software Encryption, FSE 2003, LNCS 2887,* pp. 154–169, Springer-Verlag, 2003.

[25] 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.

[26] T. Iwata and K. Kurosawa. OMAC: One-Key CBC MAC. *Fast Software Encryption, FSE 2003, LNCS 2887,* pp. 129–153, Springer-Verlag, 2003.

[27] T. Iwata and K. Kurosawa. Stronger security bounds for OMAC, TMAC and XCBC. *Progress in Cryptology — INDOCRYPT 2003, LNCS 2904,* pp. 402–415, 2003.

[28] 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.

[29] 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.

[30] 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/`.

[31] 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`.

[32] J. Jonsson. On the security of CTR + CBC-MAC. *Selected Areas in Cryptography, SAC 2002, LNCS 2595,* pp. 76–93, Springer-Verlag, 2002.

[33] C.S. Jutla. Encryption modes with almost free message integrity. *Advances in Cryptology — EUROCRYPT 2001, LNCS 2045,* pp. 529–544, Springer-Verlag, 2001.

[34] T. Kohno, J. Viega, and D. Whiting. The CWC authenticated encryption (associated data) mode. NIST submission, 2003. Available at `http://csrc.nist.gov/CryotoToolKit/modes/proposedmodes/`.

[35] L.R. Knudsen. Analysis of RMAC. Comments to NIST. Available at `http://csrc.nist.gov/CryptoToolkit/modes/comments/`.

[36] K. Kurosawa and T. Iwata. TMAC: Two-Key CBC MAC. *Topics in Cryptology — CT-RSA 2003, LNCS 2612,* pp. 33–49, Springer-Verlag, 2003.

[37] R. Lidl and H. Niederreiter. Introduction to finite fields and their applications, revised edition. Cambridge University Press, 1994.

[38] M. Luby and C. Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM J. Comput.,* Vol. 17, No. 2, pp. 373–386, April 1988.

[39] 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.

[40] `http://www.cosic.esat.kuleuven.ac.be/nessie/`

[41] `http://www.nist.gov/modes`

[42] E. Petrank and C. Rackoff. CBC MAC for real-time data sources. *J.Cryptology,* Vol. 13, No. 3, pp. 315–338, Springer-Verlag, 2000.

[43] 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.

[44] P. Rogaway. Bucket hashing and its application to fast message authentication. *Advances in Cryptology — CRYPTO '95, LNCS 963,* pp. 29–42, Springer-Verlag, 1995.

[45] 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/`.

[46] P. Rogaway, M. Bellare, J. Black, and T. Krovetz. OCB: a block-cipher mode of operation for efficient authenticated encryption. *Proceedings of ACM Conference on Computer and Communications Security, ACM CCS 2001,* ACM, 2001.

[47] M. Dworkin, NIST Special Publication 800-38A. Recommendation for block cipher modes of operation. Available at `http://csrc.nist.gov/CryptoToolkit/modes`.

[48] M. Dworkin, NIST Special Publication 800-38B. Draft recommendation for block cipher modes of operation: the RMAC authentication mode. Available at `http://csrc.nist.gov/CryptoToolkit/modes`.

[49] M. Dworkin, NIST Special Publication 800-38C. Draft recommendation for block cipher modes of operation: the CCM mode for authentication and confidentiality. Available at http://csrc.nist.gov/CryptoToolkit/modes.

[50] D. Wagner. Comments on RMAC. Comments to NIST. Available at http://csrc.nist.gov/CryptoToolkit/modes/comments/.

[51] D. Whiting, R. Housley, and F. Ferguson, Counter with CBC-MAC (CCM). NIST submission, 2002. Available at http://csrc.nist.gov/CryotoToolKit/modes/proposedmodes/.