可変出力の長さを持つメッセージ認証コード:VMAC

許, 容碩
九州大学大学院システム情報科学府

櫻井, 幸一
九州大学大学院システム情報科学研究院

出版情報：2004年 暗号と情報セキュリティシンポジウム. 2, pp.1121-1126, 2004-01
バージョン：
権利関係：

可変出力の長さを持つメッセージ認証コード:VMAC
MAC Algorithm with Variable Output-Size: VMAC

許容碩 * 櫻井幸一†
Yong-Sork HER Kouichi SAKURAI

あらまし 本論文では，SHA-V を基盤にする可変出力長を持つメッセージ認証コード，VMAC を提案する． 2002 年に，我々は SHA-1 と HAS-V を基盤とする可変出力長のハッシュ関数，SHA-V を提案した．一般的に，MAC の目的はメッセージの認証とデータ完全性である．MAC はブロック暗号システム，対称鍵暗号システム，暗号学的ハッシュ関数を基盤とする．暗号学的ハッシュ関数を基盤とする MAC はハードウェアとソフト ウェアの実装での速度面では優秀であるが，出力メッセージ長が短いため選択平文攻撃に対して脆弱である．本論文で提案する VMAC は暗号学的ハッシュ関数の速い処理速度を維持しながら，選択平文攻撃に対して耐性をもつ．

キーワード メッセージ認証コード (MAC)，ハッシュ関数，SHA-1，SHA-V，安全性

1 はじめに

1.1 動機

メッセージ認証コード MAC はメッセージ認証とデータ完全性のサービスを提供する暗号技術である．コンピュータとネットワークの速度増加によって，速い処理速度を持つ MAC が 要求される．MAC は DES のようなブロック暗号システム，対称鍵暗号システム，暗号学的ハッシュ関数を基盤とする．今まで提案された MAC の問題点の一つとしては MAC の出力長である．MAC の出力長は本来の入力メッセージより短い． 短い出力長を持つ MAC は安全性と選択平文攻撃に対して問題がある [3]． また，ソフトウェアの実装で， 暗号学的ハッシュ関数は DES-CBC より高速であると知られている [9]．このような理由のから，以下のような MAC の要求が必要である (表 1 参照).

1. 高速な MAC が必要である． 最近の研究結果では [1，3]，公開鍵暗号システムと対称鍵暗号システムより暗号学的ハッシュ関数の方がさらに高速であることが知られている．
2. 出力長の観点で，出力長が可変である MAC が要求される．多様な入力長にともなう多様な出力の長さを持つ MAC が必要である．
3. MAC の安全性のために，出力長が長い MAC が必要である．

表 1: ブロック暗号とハッシュ関数基盤の MAC 比較

	ブロック暗号 基盤の MAC	ハッシュ関数 基盤の MAC
長所	ハードウェア の実装に適合	ソフトウェア の実装に適合
短所	ソフトウェア実装では ハッシュ関数より処理速度が遅い	選択平文攻撃 の弱点

暗号学的ハッシュ関数を利用した MAC は次のような条件が必要である [3，10].

1. 暗号鍵は始めの段階と最後の段階とハッシュ関数の繰り返しフレーズ組で，関連がなければならない.
2. 基盤となるハッシュ関数との差が最小化されなければならない.
3. 処理速度は，基盤となるハッシュ関数の処理速度とほとんど同一でなければならない.
4. 追加されるメモリ量は最小化されなければならない.
5. 設計方法は繰り返しフレーズ組を持つすべてのハッシュ関数に適用可能でなければならない.

* 九州大学大学院システム情報科学府, 〒 812-8581 福岡市東区箱崎 6-10-1, Graduate School of Information Science and Electrical Engineering, Kyushu University, 6-10-1 Hakozaki, Higashi-ku, Fukuoka, 812-8581 ysher@itslab.csce.kyushu-u.ac.jp
† 九州大学大学院システム情報科学研究院, 〒 812-8581 福岡市東区箱崎 6-10-1, Faculty of Information Science and Electrical Engineering, Kyushu University, 6-10-1 Hakozaki, Higashi-ku, Fukuoka, 812-8581 sakurai@csce.kyushu-u.ac.jp

1.2 関連研究

最初の MAC はブロック暗号の CBC(Cipher Block Chaining) と CFB(Cipher FeedBack) モードを基盤とした．メッセージ認証アルゴリズム MAA (Message Authentication Algorithm) は ISO 標準である． また，暗号学的ハッシュ関数を基盤とする MAC が提案された．MAC で暗号学的ハッシュ関数を利用する理由は，ソフトウェア実装の観点から速い処理速度が必要とされるからである [1].
Tsudik は暗号学的ハッシュ関数を基盤とする MAC を提案した [5]．また，彼はメッセージ M と鍵 K を利用して secret prefix method， secret suffix method， envelop method の 3 つモードを利用して，下記のように提案した [5].

1. Secret prefix Method :
   $MAC(M) = MD(K||M)$，K は 512 ビット秘密鍵.
2. Secret suffix Method :
   $MAC(M) = MD(M||K)$，K は 512 ビット秘密鍵.
3. Envelop Method :
   $MAC(M) = MD(M||K_2)$，$IV = K_1$ は 128 ビット秘密鍵，K_2 は 512 ビット秘密鍵.

Preneel と van Oorschot [3] は暗号学的ハッシュ関数 MDx を基盤とする MDx-MAC を提案し， 提案されたハッシュ関数基盤の MAC は小さい出力メッセージによって選択平文攻撃に対して弱点があることを指摘した．Bellare，Canetti 及び Krawczyk [6,9] は NMAC(Nested MAC) と HMAC(Hash based MAC) を提案した．NMAC ではハッシュ関数の初期値に秘密鍵を利用している．Boer, Rompay 及び Preneel と Vandewalle は暗号学的ハッシュ関数 RIPEMD を基盤とする Two-Track MAC(TTMAC) を提案した． TTMAC は並列構造で，64，96，128 または 160 ビットの可変出力を持つ.

1.3 提案方式の概要

本論文では， 可変出力長を持つ MAC(以下 VMAC と呼ぶ) を提案した． 多くの MAC アルゴリズムはメッセージの認証とデータの完全性のために提案されている．最近，MAC の入力メッセージ長より出力長が短い MAC は選択平文攻撃に脆弱であることが知られている [3]．最初の MAC はブロック暗号基盤の CBC-MAC と CFB であり，その後に，暗号学的ハッシュ関数を基盤にする MAC が提案された． 暗号学的ハッシュ関数を基盤にする MAC はブロック暗号を基盤とする MAC よりは実装において処理速度が高速である．しかし， 暗号学的ハッシュ関数を基盤とする MAC は出力メッセージ長が入力メッセージ長よりはるかに短い． 出力メッセージ長が短い場合には， 選択平文攻撃に脆弱である．本論文では，暗号学的ハッシュ関数の速い処理速度を活用して，選択平文攻撃を防ぐことができる VMAC を提案する． また，提案された VMAC は既存のハッシュ関数基盤の MAC の中で最も出力長が長い.

2 SHA-V の構造

VMAC は暗号学的ハッシュ関数 SHA-V[12] を基盤とする．我々は可変出力の長さを持つ暗号学ハッシュ関数 SHA-V を提案した．本章では SHA-V の構造に対して説明する．SHA-V の構造は右側-ライン，左側-ラインの二つの並列構造になっていて，各ラインは 80 段階で形成されている． 入力メッセージ長は 1024-ビットで，出力ハッシュ長は 128-ビットから 320-ビットまで各 32 ビットの単位で選択できる． SHA-V は SHA-1 を基盤としており，SHA-1 の長所を持つ．各段で計算されたメッセージ結果の要素は入力メッセージと単位演算の組で構成される． また，各段で計算されたメッセージ結果の要素は入力メッセージの偽造による衝突の大部分を防止できる.

2.1 初期値

表 2 で，SHA-V の初期値を示す．初期値は二つの並列ラインで区別される． 左側ラインの初期値は SHA-1[7] と同じである． 右側ラインの初期値は HAS-V[8] を基盤として，出力ハッシュ値の長さの増加のために 10 個の 32-ビットワードを持つ.

表 2: SHA-V の初期値

Left-line				
$H_0^{(i)}$	$H_1^{(i)}$	$H_2^{(i)}$	$H_3^{(i)}$	$H_4^{(i)}$
67452301	$efcdab89$	$98badcfe$	10325476	$c3d2e1f0$
Right-line				
$H_5^{(i)}$	$H_6^{(i)}$	$H_7^{(i)}$	$H_8^{(i)}$	$H_9^{(i)}$
$8796a5b4$	$4b5a6978$	$0f1e2d3c$	$a0b1c2d3$	$68794e5f$

2.2 常数

SHA-V の常数は表 3 と同じである． 左側ラインの常数は $K[3]$ 以外は SHA-1 基盤とする．SHA-1 の $K[3]$ は $Oxca62c1d6(2^{30}\sqrt{10})$ であるが，SHA-V の $K[3]$ は $Ox953fd4e$ を利用する．右側ラインの常数は HAS-V[8] を基盤とする． しかし，常数の順序は HAS-V と異なる．また，HAS-V の常数は 5 ラウンドのために 5 個の 32-ビットで構成される．しかし，SHA-V は 4 ラウンドで，4 個の 32-ビットで構成される．そのため，SHA-V は HAS-V より計算量を削減できる.

2.3 Boolean 関数

各単位演算に使われる Boolean 関数は以下に示す．左側ラインの Boolean 関数は SHA-1 を基盤にして，右側

表 3: SHA-V の常数

Left-line			
$K[0]$	$K[1]$	$K[2]$	$K[3]$
$5a837999$ $(2^{30}\sqrt{2})$	$6ed9eba1$ $(2^{30}\sqrt{3})$	$8f1bbcdc$ $(2^{30}\sqrt{5})$	$a953fd4e$ $(2^{30}\sqrt{7})$
Right-line			
$K'[0]$	$K'[1]$	$K'[2]$	$K'[3]$
$7a6d76e9$ $(2^{30\ 3}\sqrt{7})$	$6d703ef3$ $(2^{30\ 3}\sqrt{5})$	$5c4dd124$ $(2^{30\ 3}\sqrt{3})$	$50a28be6$ $(2^{30\ 3}\sqrt{2})$

Boolean 関数は左側ラインの逆順で構成される．

- Left-line

$f_t(x, y, z) = (x \wedge y) \vee (\neg x \wedge z)$ $(0 \leq t \leq 19)$

$f_t(x, y, z) = x \oplus y \oplus z$ $(20 \leq t \leq 39)$

$f_t(x, y, z) = (x \wedge y) \vee (x \wedge z) \vee (y \wedge z)$ $(40 \leq t \leq 59)$

$f_t(x, y, z) = x \oplus y \oplus z$ $(60 \leq t \leq 79)$

- Right-line

$g_t(x, y, z) = x \oplus y \oplus z$ $(0 \leq t \leq 19)$

$g_t(x, y, z) = (x \wedge y) \vee (x \wedge z) \vee (y \wedge z)$ $(20 \leq t \leq 39)$

$g_t(x, y, z) = x \oplus y \oplus z$ $(40 \leq t \leq 59)$

$g_t(x, y, z) = (x \wedge y) \vee (x \wedge z)$ $(60 \leq t \leq 79)$

2.4 可変出力計算

可変出力長を生成するために，ハッシュ値は表 4 のように計算する．この計算方法は PMD-V を基盤にする．出力長は 128-ビットから 320-ビットまで，32 ビット単位で出力する．ただし，誕生日攻撃に対する安全性を考慮して 128 ビットは使用しない．SHA-V の特徴は各ラインで公正に計算される．計算速度の向上のために，シフトと加算を利用する．

3 VMAC の構造

本章では VMAC の構造について説明をする．VMAC の構造は SHA-V と共に左側-ラインと右側-ラインの並列構造を持つ．メッセージは 1024-ビットの倍数で構成されて，鍵の拡張の過程とメッセージの拡張の過程で構成される．

■ VMAC のアウトライン

1. SHA-V の初期値を VMAC の鍵として利用する．
2. 鍵は NMAC のような opad と ipad を利用して，512 ビットに拡張する．
3. 拡張した鍵は，SHA-V を利用して新しい 160 ビットの鍵にする．
4. SHA-V の常数は VMAC の常数を利用する．
5. VMAC で最初のメッセージ $M^L{}_1$,$M^R{}_1$ に対する SHA-V の出力値を，2 番目のラウンドの鍵として利用する．
6. 最終的に VMAC の出力値は，表 4 ような計算になり，可変な出力長を持つ．

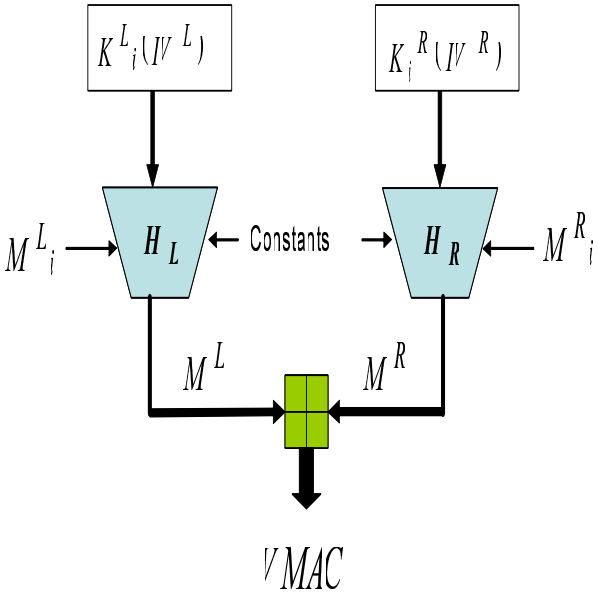

図 1: 一つのブロックでの VMAC

■ 鍵の生成

VMAC は NMAC[6,9] ように暗号学的ハッシュ関数の初期値に秘密鍵を利用する． すなわち，SHA-V の初期値 IV_L と IV_R は VMAC の秘密鍵 $K^L{}_{IV}$ と $K^R{}_{IV}$ に対応する．$K^L{}_{IV}$ と $K^R{}_{IV}$ は NMAC のように下記のように拡張される．

$$opad = Ox36 \text{ 44 回反復}$$
$$ipad = Ox5c \text{ 44 回反復}$$
$$K^L{}_0 = (K^L{}_0 || opad) = 512 \text{ ビット}$$
$$K^R{}_0 = (K^R{}_0 || ipad) = 512 \text{ ビット}$$
$$K^L{}_{IV} \text{ と } K^R{}_{IV} \text{ は各 512 ビットである．}$$
$$K^L{}_i = h(K^L{}_0),\ K^R{}_i = h(K^R{}_0)$$

h は暗号学的ハッシュ関数 SHA-V である．

■ VMAC の生成順序

メッセージ長は 1024-ビットの倍数で，各メッセージは 1024-ビットである．繰り返しメッセージが 1024-ビットより小さいならば，SHA-V のように入力メッセージをパディングする． パディング後，各メッセージは M^L と M^R の 2 つに分けられる．左側ラインのメッセージは M^L で， M^L の長さは 512-ビットの倍数である． また，M^L は $M^L{}_1 M^L{}_2, \ldots, M^L{}_n$ で構成される． また，$M^L{}_i$

表 4: SHA-V と VMAC での可変出力長の計算

	128-bit	160-bit	192-bit	224-bit	256-bit	288-bit	320-bit
A	$A + V^{<<2}$	$A + F$	$A + J^{<<2}$	$A + J^{<<2}$	$A + J^{<<2}$	$A + J^{<<2}$	A
B	$B + V^{<<3}$	$B + G$	$B + J^{<<3}$	$B + J^{<<3}$	$B + J^{<<3}$	$B + J^{<<3}$	B
C		$C + H$	$C + J^{<<5}$	$C + J^{<<5}$	$C + J^{<<5}$	$C + J^{<<5}$	C
D		$D + I$		$D + J^{<<7}$	$D + J^{<<7}$	$D + J^{<<7}$	D
E		$E + J$		$E + J^{<<11}$		$E + J^{<<11}$	E
F	$F + W^{<<2}$		$F + J^{<<2}$	$F + J^{<<13}$	$F + E^{<<2}$	$F + J^{<<13}$	F
G	$G + W^{<<3}$		$G + J^{<<3}$	$G + J^{<<17}$	$G + E^{<<3}$	$G + J^{<<17}$	G
H			$H + J^{<<5}$		$H + E^{<<5}$	$H + J^{<<19}$	H
I					$I + E^{<<7}$	$I + J^{<<23}$	I
J							J

$V = C \oplus D \oplus E$, $W = H \oplus I \oplus J$($\oplus$は exclusive $-$ OR)

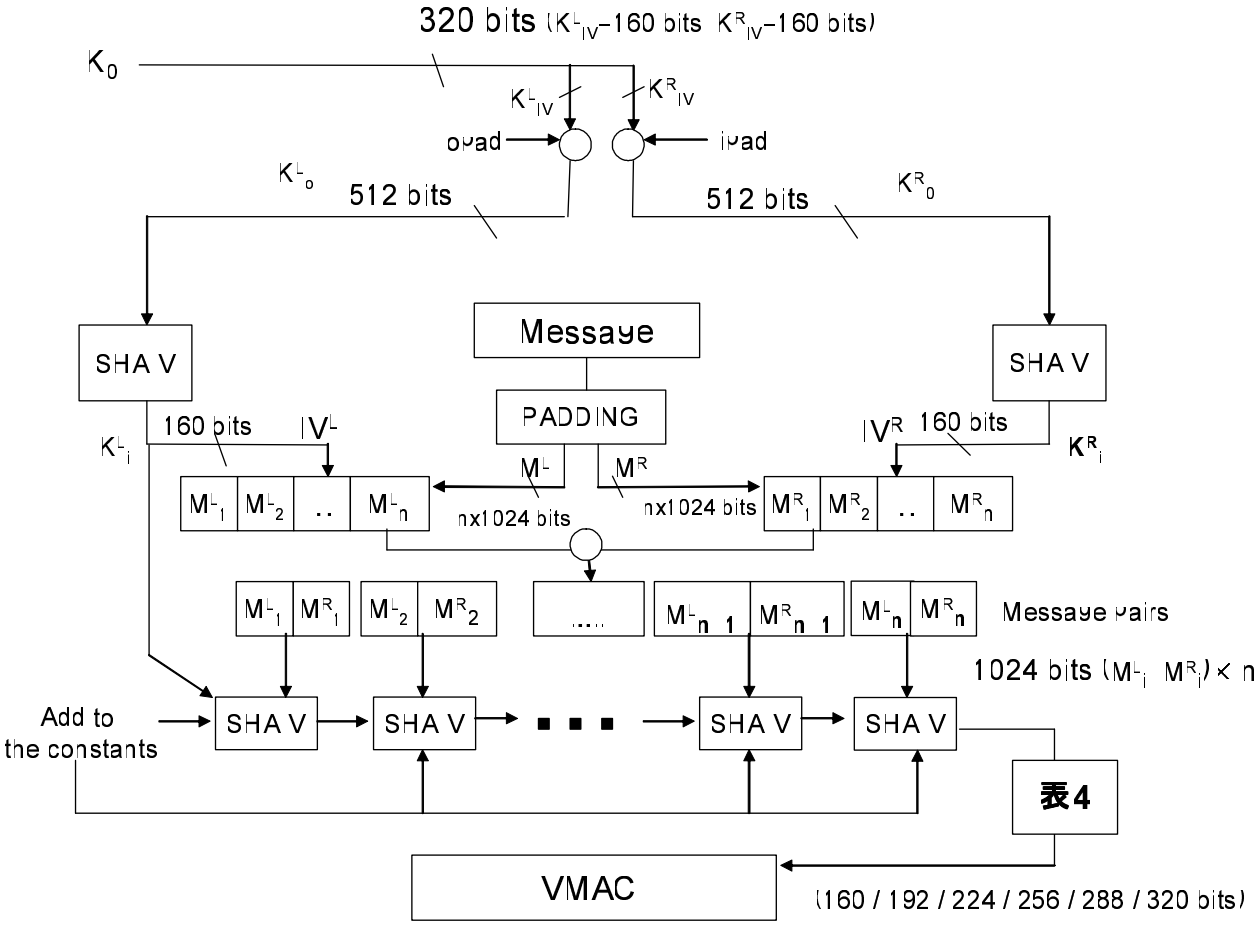

図 2: VMAC の構造

は A_i, B_i, C_i, D_i, E_i の 5 個の 32-ビットで構成される．VMAC は下記のように計算する (図 2 参考).

$$M = (M^L{}_i, M^R{}_i)$$
$$M^L{}_i = M^L{}_1, M^L{}_2, \ldots, M^L{}_n$$
$$M^R{}_i = M^R{}_1, M^R{}_2, \ldots, M^R{}_n$$
$$M^L{}_i || M^R{}_i$$
$$= (M^L{}_1 || M^R{}_1), (M^L{}_2 || M^R{}_2), \ldots, (M^L{}_n || M^R{}_n)$$

各メッセージの対は次のように計算される．

$$M'_i = h(M^L{}_i, M^R{}_i)$$
$$= (A_i || B_i || C_i || D_i || E_i || F_i || G_i || H_i || I_i || J_i)$$

$A_i||B_i||C_i||D_i||E_i||F_i||G_i||H_i||I_i||J_i$ は表 4 ように計算される．$VMAC = M'_i \oplus M'_i \oplus \cdots \oplus M'_n$
すなわち
$VMAC = h(M^L, h(K^L{}_{IV}||ipad), Con) \odot$
$h(M^R, h(K^R{}_{IV}||opad), Con)$
Con は常数で，$\odot$ は表 4 の可変出力のための計算方法である．

4 VMAC の安全性

VMAC の安全性は暗号学的ハッシュ関数 SHA-V を基盤とする．SHA-V の並列ライン，右側ラインと左側ラインは各 80 段階の演算をして，各段階では AND， OR, シフトをする．特に，可変出力のため，最終計算で，実装での処理速度の向上のためにシフトと加算をする．シフト演算でのシフトの数は互いに異なるように行う．これにより SHA-V の内部衝突を防止できる．また， SHA-V の最終出力長は 320 ビットである．320 ビットより短い出力長が要求される時は， 最終計算でもう一度シフトと加算を行うことにより圧縮できる．また， 右側ラインと左側ラインの Boolean 関数，常数，初期値が互いに異なるように構成されているため，ライン間での関連性がない．VMAC では SHA-V を鍵とメッセージの圧縮に利用する (図 2 参考)．そして，鍵長も SHA-V の出力長のように，出力長を可変にすることができ，メッセージの出力長も可変にできる．図 2 では，左側と右側ラインの鍵長が 160 ビットこと時の VMAC を説明する．各ラインの最大鍵長は 320 ビットにできる．すなわち，VMAC は SHA-V を利用しているため，鍵長，メッセージの出力長を可変にでき，現在までに提案された他の MAC の出力長より最大 2 倍の出力長を持つ．既存の暗号学的ハッシュ関数と MAC に攻撃に対する安全性は 5 章で説明する．

5 攻撃に対する耐性

5.1 TTMAC vs．VMAC

本章では，TTMAC と VMAC を比較する．TTMAC は RIPEMD-160 を基盤とする MAC である．TTMAC は Boer，Rompay，Preneel，Vandewalle により提案され，NESSIE プロジェクトに提出された [13]．TTMAC の特徴は短いメッセージで早い処理速度を持つ．表 5 で，可変出力長がを持つ TTMAC と VMAC を比較する．

表 5: TTMAC vs．VMAC

	TTMAC	VMAC
入力メッセージ長さ	512-ビットの倍数	1024-ビットの倍数
MAC の出力長	64, 96, 128 ビット	160, 192, 224,256, 288, 320 ビット

5.2 誕生日攻撃

Preneel らは反復関数を基盤にして鍵を持つ MAC に誕生日攻撃の可能性を指摘した [9]．誕生日攻撃はハッシュ値の長さに起因する．ハッシュ値の長さを l とする．衝突発生の可能性は $2^{l/2}$ である．一般的に，誕生日攻撃を避けるための最小限のハッシュ長は 160-ビットとして知られている．VMAC の場合には，SHA-V の可変出力値段で 128-ビットを除外した 160-ビットから 320 ビットまでの 32 ビット単位で生成できる．

5.3 偽造攻撃

MAC に対する他の攻撃は内部衝突を基盤とする偽造攻撃である． 攻撃者は平文を収集できる． 文献 [1] で，Boer らは一般的な攻撃に対する TTMAC の耐性を示した．我々は Boer らの方法を利用して，MAC の一般的な攻撃に対する耐性を示す． 鍵長を k，出力長を m と仮定する．最初に，攻撃者は鍵の全数探索をする．即ち，2^k 個の鍵を探索をする．次に，既知の平文と MAC の対は k/m 個である．また，選択平文に対応する MAC が見つかる確率は $1/2^m$ である．偽造攻撃に対する VMAC の耐性の結果は $1/2^{160}$ から $1/2^{320}$ である．

6 まとめ

MAC は暗号学と応用分野でメッセージ認証とデータ完全性の重要な役割をする． MAC はブロック暗号を基盤としたものが提案されて，その後，対称鍵暗号と暗号学的ハッシュ関数を利用した MAC も提案された． 処理速度では暗号学的ハッシュ関数が優秀である． しかし，出力メッセージ長が短いために選択平文攻撃に脆弱

である．2002 年に，我々は可変出力長を持つハッシュ関数 SHA-V を提案した． 本論文では，SHA-V を利用した可変出力長を持つ VMAC を提案した． VMAC はハッシュ関数の誕生日攻撃と偽造の攻撃に対して安全である．そして，既存の MAC の出力長より長く，可変出力であるため，様々な応用が可能である．

謝辞

本研究の一部は，文部科学省科学研究費補助金学術創成研究課題番号 14GS0218『社会基盤を構築するためのシステム LSI 設計手法の研究 (研究代表安浦寛人九州大学システム LSI 研究センター長) と 21 世紀 COE プログラム「システム情報科学での社会基盤システム形成」の支援を受けている．

参考文献

[1] B.den Boer, B.V Rompay, B.Preneel and J.Vandewalle *"New (Two-Track-)MAC Based on the Two Trails of RIPEMD "* SAC2001, LNCS 2259, Spirnger-Verlag, pp. 314-324, 2001.

[2] Black, J., Halevi, S., Krawczyk, H., Krovetz, T., and Rogaway, P. " *UMAC:Fast and secure message authentication* " CRYPTO '99, LNCS 1666, Springer-Verlag, pp.216-233, 1999.

[3] B.Preneel and P.C.van Oorshot " *MDx-MAC and Building Fast MACs from Hash Functions* " CRYPTO '95, LNCS 963, Springer-Verlag, pp.1-14, 1995.

[4] D.Davies " *A Message Authenticator Algorithm Suitable for a Mainframe Computer* " CRYPTO '84, LNCS 196, Springer-Verlag, pp.393-400, 1984.

[5] G.Tsudik" *Message Authentication with One-Way Hash Functions* "Proc. of Infocom 92, 1992.

[6] M.Bellare, R.Canetti and H.Krawczyk " *Keying Hash Functions for Message Authentication* " CRYPTO 96, LNCS 1109, Springer-Verlag, pp. 1-15, 1996.

[7] NIST " *Secure Hash Standard* "FIPS PUB180-1, May, 1993

[8] N.K.Park, J.H.Hwang, P.J.Lee " *HAS-V : A New Hash Function with Variable Output Length* " SAC2000, LNCS2012, 2001.

[9] M.Bellare, R.Canetti and H.Krawczyk " *Message Authentication using Hash Functions ? The HMAC Construction* "Appears in RSA Laboratories ' CryptoBytes, Vol.2, No.1, Spring 1996.

[10] M.Semanko " *L-collision attacks against randomized MACs* " CRYPTO '2000, LNCS 1880, Springer-Verlag, pp. 216-228, 2000.

[11] S.U.Shin, K.H.Rhee, D.H.Rye and S.J.Lee " *A New Hash Function Based on MDx-Family and Its Application to MAC* " PKC98, LNCS 1431, Springer-Verlag, pp.234-246, 1998.

[12] Y.S.Her and K.Sakurai " *Design and Analysis of Cryptographic Hash Function for the Next Generation* "International Workshop on Informations & Electrical Engineering 2002, pp168-pp173, 2002.

[13] NESSIE Project *New European Schemes for Signature, Integrity and Encryption* http://cryptonessie.org