自己評価書
PC-MAC-AES

日本電気株式会社

目次

1 概要

本稿は，PC-MAC-AES の安全性および実装性に関する自己評価の報告書である．

具体的な評価の内容の前に，PC-MAC-AES の設計指針および他の技術との違いについて述べる．AES[7] がブロック暗号として広く使われ，またその安全性に高い信頼が置かれている現在，AES の暗号モードでメッセージ認証コード（MAC）を実現することはごく自然なアプローチとなっている．このような暗号モードとしては古典的な CBC-MAC, EMAC[9], CMAC[8] などが挙げられる．これらのモードには，使用するブロック暗号の証明可能安全性に基づいた厳密な安全性証明（安全性帰着）があるという利点を有する．

しかしながら，これらのモードを AES を用いて実現した場合，基本的に AES をブラックボックスとして扱うため，1 メッセージブロックあたり少なくとも 1 回の AES 暗号化関数を適用する必要がある．したがって処理速度は AES のそれを上回ることはできないという限界がある．一方，代数的関数（有限体上の多項式演算など）を用いて構成されるユニバーサルハッシュ関数と AES を組み合わせることで，AES 単体の速度を上回る処理速度が実現可能であることが様々な研究（例えば [11][15]）で報告されている．しかしこちらのアプローチは，代数的関数を実装するためのコスト・性能を出すためのチューニングが必要となる．例えば，一般的に有限体上の多項式演算はソフトウェア上で大量のメモリを用いることで高速化が可能となるが，同じ高速化手法はハードウェアでは現実的ではない．このジレンマを打破するためには，AES の段関数をとりだし，これも部品として有効活用することが考えられる．このアプローチは最初に Daemen らにより alpha-MAC[13] として提案され，その後 Pelican[14] というバリエーションを生み出した．これらの提案は AES の実装のみで AES より高速な MAC を実現するものとして注目を集めたが，AES による安全性帰着がなくなるという大きな欠点を有していた．つまり，これらの MAC には AES 単体への攻撃が存在しなくとも有効な攻撃が存在する可能性がある．

本提案はこれに対して，AES の差分攻撃に対する理論的安全性証明の結果 [19][20] を用いて，CMAC[8]（OMAC[17]）などのモードを AES で実現したときと同等の安全性帰着を実現しつつ，これらより 1.4 から 2.5 倍ほど（推奨パラメータにおいては 1.4 から 2 倍ほど）の高速化を達成するものである．具体的な方式は峯松と角尾による文献 [23] に記載の PC-MAC をほぼそのままベースとしている．これは，CBC-MAC の処理における AES の暗号化関数コールの一部を周期的に AES の 4 段関数コールで置き換えるものである．alpha-MAC や Pelican などと異なり，証明可能安全性を保証するために，AES の 4 段関数へは AES の暗号化関数を用いた鍵スケジュールにより鍵を設定している．

2 安全性に対する評価概要

AES がランダム置換と選択平文攻撃（CPA）において計算量的に判別困難であれば，PC-MAC-AES が可変長入力ランダム関数と判別困難であることが文献 [23] で証明されている．この証明においては，Keliher ら [19][20] による，4 段 AES 関数の最大期待差分確率（Maximum Expected Differential Probability, MEDP）の解析結果が重要な役割を果たしている．可変長入力ランダム関数は MAC として理想的な関数であり，上記の証明と既存の結果 [10] を組み合わせることで，MAC としての安全性，すなわち（適応的）選択平文攻撃における偽造困難性，が証明できる．AES に対する仮定と達成される安全性については，CMAC など従来の証明可能安全性を有する MAC と同等である．以下で詳説する．なお，AES はすべて 128 ビット鍵の AES を指すものとする．PC-MAC-AES の仕様（例えばパラメータ π, d）については仕様書 [1] を参考のこと．

3 安全性評価

3.1 安全性指標と攻撃モデル

MAC に対する以下の安全性評価の枠組みは標準的なものである．詳細は例えば CRYPTREC 技術報告書 [6] を参照のこと．

MAC に対する攻撃者はタギングオラクルと検証オラクルの二つにアクセスが可能であるとする．ターゲットが決定的 MAC 関数 F のとき，攻撃者はメッセージ M をタギングオラクル O_T^F へ質問して $Y = F(M)$ を得るか，メッセージとタグの対 (M', T') を検証オラクル O_V^F へ質問して，もし $F(M) = T$ であれば 1 を，そうでない場合は 0 を得る．ただし M' は過去に O_T^F へ質問していないメッセージである．ここでは，O_V^F から 1 という回答を得た場合，あるメッセージに対する正当なタグを O_T^F に質問することなく求めたことになる．このイベントを偽造の成功と定義する．

攻撃者 $\mathcal{A}$ の計算量 τ，O_T^F への質問回数 q，O_V^F への質問回数 q_v，一つの質問でのメッセージの長さの最大値を ρ ブロック（ブロック長はあらかじめ定めておく：ここでは 128 ビット）のとき，$\mathcal{A}$ を (q, q_v, τ, ρ)-forger であるといい，$\mathcal{A}$ が O_V から少なくとも 1 回は回答 1 を得る確率を $\mathtt{FP}_F(\mathcal{A})$ とする．このとき，

$$\mathtt{FP}_F(q, q_v, \tau, \rho) = \max_{\mathcal{A}:(q,q_v,\tau,\rho)\text{-forger}} \mathtt{FP}_F(\mathcal{A})$$

を最大偽造成功確率と呼び，MAC に対する安全性評価の基準とする．

3.2 疑似ランダム性

ターゲットの MAC 関数 F がカウンターなどの初期値を必要としない，決定的 MAC 関数である場合，最大偽造成功確率の評価は F の疑似ランダム性の評価から直ちに導出が可能である．

これは，選択平文攻撃 (Chosen Plaintext Attack, CPA) による F とランダム関数 R の判別困難性を評価するものである．判別を目的とした攻撃は，二つのタギングオラクル O_T^F と O_T^R のどちらかが真という状況において，質問とその回答からどちらが真であるかを判別する（質問後に 0, 1 の 2 値判定を行う）ものである．ここで，質問は適応的に，過去の質問・回答結果を元に選択できる．

判別を目的とした攻撃者 $\mathcal{B}$ が q 回の質問（ただし各質問は高々 ρ ブロック）と計算量 τ を有するとき，$\mathcal{B}$ を (q, τ, ρ)-distinguisher と呼ぶ．また，$\mathcal{B}$ がオラクル O_T^F へ q 回の質問ののち 1 を出力する事象を $\mathcal{B}^F = 1$ と表記する．このとき，

$$\mathtt{Adv}_{F,R}^{\mathtt{cpa}}(\mathcal{B}) = |\Pr[\mathcal{B}^F = 1] - \Pr[\mathcal{B}^R = 1]|$$

とし，これを $\mathcal{B}$ の F と R の判別における advantage と呼ぶ．一般に，F の疑似ランダム性は F と同じ入出力幅を持つランダム関数 R との最大 advantage で評価する．すなわち

$$\mathtt{Adv}_F^{\mathtt{prf}}(q, \tau, \rho) = \max_{\mathcal{B}:(q,\tau,\rho)\text{-distinguisher}} \mathtt{Adv}_{F,R}^{\mathtt{cpa}}(\mathcal{B})$$

で評価する．同じ (q, τ, ρ) について上記が小さい関数ほど高い擬似ランダム性を有する．特に，F が n ビットの鍵付き置換（すなわちブロック暗号）である場合，P を n ビットランダム置換として，

$$\mathtt{Adv}_F^{\mathtt{prp}}(q, \tau) = \max_{\mathcal{B}:(q,\tau)\text{-distinguisher}} \mathtt{Adv}_{F,P}^{\mathtt{cpa}}(\mathcal{B})$$

とする．このときは攻撃者の質問は常に n ビットとなるため，ρ の記述は不要となっている．

さらに F が可変長入力の場合，R^* を可変長入力のランダム関数とし，

$$\mathtt{Adv}^{\mathtt{vilprf}}_{F}(q,\tau,\rho) = \max_{\mathcal{B}:(q,\tau,\rho)\text{-distinguisher}} \mathtt{Adv}^{\mathtt{cpa}}_{F,R^*}(\mathcal{B})$$

で疑似ランダム性を評価する．

3.3 PC-MAC-AES の安全性

PC-MAC-AES の安全性は，まず PC-MAC-AES の疑似ランダム性を証明したのち，決定的 MAC 関数が疑似ランダム性を有する場合の一般的な安全性証明と組み合わせることで得られる．PC-MAC-AES の疑似ランダム性を証明するには，内部で用いる 4 段 AES 関数 G_U（図 1）の差分確率評価が必要となる．

定義 3.1 F_K が n ビットの鍵付き置換であるとき，F_K の最大期待差分確率（MEDP）と最大期待自己差分確率（Maximum expected self-differential probability, MESDP）を，

$$\mathrm{MEDP}(F_K) = \max_{a\neq 0,b} \Pr(F_K(X)\oplus F_K(X\oplus a) = b),$$
$$\mathrm{MESDP}(F_K) = \max_{a\in\{0,1\}^n} \Pr[X\oplus F_K(X) = a],$$

とする．ただし X は n ビット乱数である．ここで，確率はそれぞれ F の鍵 K の分布と X の分布から定まるものである．

ここで，4 段 AES 関数 G_U について，384 ビットの U が一様分布に従う場合，Keliher ら [19][20] により $\mathrm{MEDP}(G_U) \le 2^{-113}$ が知られている．また AES の段鍵加算の構造より，$G_U(X)$ は入力 X と独立であるため $\mathrm{MESDP}(G_U) = 2^{-128}$ は自明である．

なお，文献 [23] の AES ベースの PC-MAC 実装においては，4 段 AES 関数の最後の ShiftRows と MixColumns を省略して用いている．ここで提案する PC-MAC-AES はこの省略をせずに 4 段 AES 関数を用いるものである．しかしこの違いは証明可能安全性に影響を与えない．これは，ある n ビット (鍵付き) 置換 F と線形な置換 g について，

$$\mathrm{MEDP}(F) = \mathrm{MEDP}(g\circ F)$$

が成立し，Keliher ら [19][20] が導出した MEDP の上界が 4 段 AES 関数の最後の ShiftRows と MixColumns の存在の有無に関わらず成立するためである．

文献 [23] の Theorem 2 において，以下の結果が得られている．

定理 3.1（文献 [23], Theorem 2）$c = \lceil d|\mathcal{U}|/n\rceil + d$ とすると

$$\mathtt{Adv}^{\mathtt{vilprf}}_{\text{PC-MAC}_d[E_K,L|G_U]}(q,\tau,\rho) \le \mathtt{Adv}^{\mathtt{prp}}_{E_K}(\rho q + c,\tau') + \frac{2.5(\rho q+c)^2}{2^n} + (d\epsilon_{\mathrm{dp}} + \epsilon_{\mathrm{sdp}})\frac{q^2}{2},$$

が成り立つ．ただし $t' = t + O(\rho q)$, $\epsilon_{\mathrm{dp}} = \mathrm{MEDP}(G_U)$, $\epsilon_{\mathrm{sdp}} = \mathrm{MESDP}(G_U)$ である．

ここで，$\text{PC-MAC}_d[E_K, L|G_U]$ は，n ビットの鍵 K と L を用い，パラメータ d を持ち，n ビットブロック暗号 E_K と n ビット鍵付き置換 G_U（鍵 U は集合 $\mathcal{U}$ の要素）を部品とした，文献 [23] の定義による PC-MAC である．文献 [23] の定義は，仕様書 [1] に記載の $\text{PC-MAC-AES}_{(128,d)}$ の，AES とその 4 段関数に限定しない

一般化に相当する．実際に AES 暗号化関数を E_K，G_U を 4 段 AES 関数としたとき，PC-MAC$_d[E_K, L|G_U]$ は仕様書 [1] に記載の PC-MAC-AES$_{(128,d)}$ と完全に一致する（詳細は文献 [23] を参照のこと）．

上記の定理に対して $n = 128$，$|U| = 384$，$\mathrm{MEDP}(G_U) \le 2^{113}$，$\mathrm{MESDP}(G_U) = 2^{-128}$ を代入することで，提案の PC-MAC-AES$_{(\pi,d)}$ に対する疑似ランダム性が以下で証明される．

系 3.1 任意の $d \ge 1$ と $1 \le \pi \le 128$ について，

$$\mathtt{Adv}^{\mathtt{vilprf}}_{\text{PC-MAC-AES}_{(\pi,d)}}(q,\tau,\rho) \le \mathtt{Adv}^{\mathtt{prp}}_{E_K}(\rho q + 4d, \tau') + \frac{2.5(\rho q + 4d)^2}{2^{128}} + \left(\frac{d}{2^{114}} + \frac{1}{2^{129}}\right) q^2,$$

ただし $\tau' = \tau + O(\rho q)$ である．

なお，任意の $1 \le \pi \le 128$ で上記の上界が成立するのは，タグを短くすることは攻撃者にとって不利であり，同じ出力長のランダム関数との判別成功確率を上げないためである．

最後に，この結果をもとに PC-MAC-AES$_{(\pi,d)}$ に対する偽造成功確率を評価する．可変長入力，π ビット出力の疑似ランダム関数 F を決定的 MAC 関数として用いた場合，その偽造成功確率は

$$\mathtt{FP}_F(q, q_v, \tau, \rho) \le \mathtt{Adv}^{\mathtt{vilprf}}_F(q + q_v, \tau', \rho) + \frac{q_v}{2^\pi}, \tag{1}$$

ただし $\tau' = \tau + O((q + q_v)\rho)$，として上界されることが証明されている（文献 [10], Proposition 7.3）．

したがって，系 3.1 と式 (1) を組み合わせて以下が得られる．

$$\begin{aligned}
\mathtt{FP}_{\text{PC-MAC-AES}_{(\pi,d)}}(q, q_v, \tau, \rho) &\le \mathtt{Adv}^{\mathtt{vilprf}}_{\text{PC-MAC-AES}_{(\pi,d)}}(q + q_v, \tau', \rho) + \frac{q_v}{2^\pi}, && (2)\\
&\le \mathtt{Adv}^{\mathtt{prp}}_{E_K}(\rho(q + q_v) + 4d, \tau') && (3)\\
&\quad + \frac{2.5(\rho(q + q_v) + 4d)^2}{2^{128}} + \left(\frac{d}{2^{114}} + \frac{1}{2^{129}}\right)(q + q_v)^2 + \frac{q_v}{2^\pi} && (4)
\end{aligned}$$

となる．式 (4) の右辺第 1 項は AES の疑似ランダム性（ランダム置換との計算量的判別困難性）を表す項なので十分小さいとされる．従って，PC-MAC-AES$_{(\pi,d)}$ に対する偽造成功確率は AES の安全性に帰着されることが証明できる．

前節の偽造成功確率の上界において一般に支配的なのは $\frac{d}{2^{114}}(q + q_v)^2$ である．仕様書 [1] に従い $d \le 5, \pi \ge 64$ とすれば，q と q_v が十分に 2^{56} より小さいならば，PC-MAC-AES$_{(\pi,d)}$ に対する偽造成功確率は十分小さく抑えられる．この質問回数の許容される上限は CBC-MAC ベースの MAC のそれと比べると若干小さい（EMAC, CMAC など CBC-MAC ベースではおおむね 2^{64} 程度）が，現実的な問題とはならないと考えられる．

3.4 鍵 L の効果について

PC-MAC-AES の鍵は AES の鍵 K と，これと独立な 128 ビットの鍵 L の二つからなる．しかし後者は MAC としての計算量的安全性を 256 ビットに高めるためのものではなく，Tweakable ブロック暗号 [22] というブロック暗号の拡張を実現し，結果的に高い実行効率をシンプルな方法で実現するためのものである．もし L が存在しない場合，鍵スケジュール KeySch（仕様書参照）とタグ生成 TagGen における AES への入力を攻撃者が容易に衝突させられるため，安全ではない．また最終メッセージブロックの処理においても同様の問題が生じる．さらに，L を AES 暗号化関数 E_K へ定数を入れて暗号化した結果の暗号文とすることも安全ではない．ゆえに鍵 L の導入は適切と考えられる．

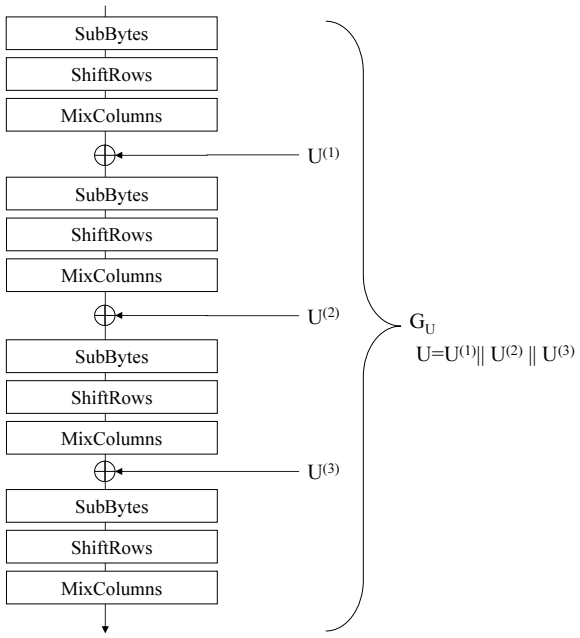

図 1　4 段 AES 関数 G_U

3.5　CRYPTO 2009 で報告された攻撃について

CRYPTO 2009 において Yuan ら [31] により PC-MAC-AES$_{(128,d)}$（ただし文献 [23] の定義に従ったバージョンのため 4 段 AES 関数の最終 ShiftRows と MixColumns が省略されている）への鍵回復攻撃が報告されているが，$q = 2^{85.5}$ と 2^{128} の計算量を要するため非現実的である．またそもそもこの質問回数は安全性が保証される範囲を大きく超えているため，本稿で示した安全性証明と矛盾するものではない．Yuan らの結果の新規性は，（偽造成功だけでなく）鍵回復を目的とした攻撃の計算量を導出した点にある．Yuan らの指摘にあるように 256 ビットの鍵を持つにも関わらず 2^{128} の計算量で鍵が求まるが，3.4 節で述べたように，鍵 L の役割が攻撃への耐性を高めるためではなく Tweakable ブロック暗号を実現して効率を高めるためのものであることから自然な結果といえる．

いずれにせよ，前述の結果から，現実の使用においては q，q_v ともに 2^{56} に達する前に鍵を更新することが強く推奨される．なお，$q = 2^{64}$ 程度あれば 128 ビットの内部変数の衝突により CMAC などの多くの一般的な MAC において偽造困難性が破れることが知られている（例えば文献 [24]）．このような攻撃はバースデー攻撃と呼ばれる．PC-MAC-AES も 128 ビットのデータパスを用いる以上同種の偽造困難性を破る攻撃が存在することが報告されている [18]．しかしながらこれは，128 ビットのデータパスを用いる一般的な MAC に共通する限界である（例えば文献 [18] や文献 [16] の指摘を参照のこと）．

3.6　サイドチャネル攻撃への安全性

同じく 4 段 AES 関数を利用している alpha-MAC については内部変数の衝突を利用したサイドチャネル攻撃が報告されている [12]．PC-MAC-AES については既知の報告はないものと思われる．しかし一般的に

AES に対するサイドチャネル攻撃を応用することは可能と推測されるため，PC-MAC-AES が証明可能安全性を持つとは言え，サイドチャネル攻撃に対する対策は別途考慮する必要があると考えられる．

なお，AES へのサイドチャネル攻撃に対する汎用的な対策，例えばハードウェアでは [25][26][28][30][29] などは適用可能と考えられるが，この対策のみで MAC 関数全体としての安全性が保証されるとは限らず，検証が必要である．

4 実装性

4.1 一般的性質

AES とその 4 段関数を部品として用いるため，その実行効率は基本的にプラットフォームに依存せず概算することが可能である．文献 [23] の Table 1. に従い，表 1 が得られる．この表から，CMAC と比べて $d = 1$ でおよそ 1.4 倍の高速化，$d = 3$ でおよそ 1.8 倍の高速化が図れることが分かる．d を増やすことにより漸近的には 2.5 倍の高速化が可能であるが，事前処理とその結果を保存するのに必要なメモリが d について線形に増加するため，d を極端に大きくすることは現実的ではない．

表 1 CMAC-AES との効率の比較．Rounds は 1 メッセージブロックを処理するのに必要な平均の AES 段数，Preproc はタグ生成をはじめる前に必要な AES 暗号化回数．

MAC	Rounds	Preproc	Key size
PC-MAC-AES$_{(\pi,d)}$	$4 + \frac{6}{d+1}$	$4d - 1$	256
CMAC-AES	10	1	128

以下，ソフトウェアおよびハードウェアでの実装とその性能について述べる．

4.2 ソフトウェア実装性

AES のソースコードがあれば基本的に実装は容易である．また，ソースコードでなくとも，段関数（すなわち，SubBytes, ShihtRows, MixColumns, AddRoundKey の合成関数）へのアクセスが可能なライブラリか専用命令があれば，これを用いて実現することも容易である．このような環境としては，例えば Intel の高級 CPU に搭載予定の AES-NI [4] が挙げられる．AES-NI では段関数単位の命令が利用可能である [27].

ソフトウェア実装は C 言語で行った．AES 自体のコードはパブリックドメインのソース [5] を利用している．開発環境やソースコードに関する詳細は参照ソースコード仕様書 [2] を参照のこと．標準的な PC の上で速度などを計測した結果を以下に記す．比較のために CMAC-AES も実装している．開発と実験環境は以下である．

- CPU：Intel Core Duo 1.66GHz
- オペレーティングシステム：Microsoft Windows XP Professional
- 記述言語：ANSI C
- コンパイラ：Microsoft Visual C++ 2008 Express Edition
- 最適化：速度最適化（/O2 オプション指定）

上記の環境のもと，32 ブロックメッセージ（4096 ビット）について 1024 回鍵スケジュールとタグ生成を

実行し，平均サイクル数を記録した．タグ長 $\pi = 128$ とし，オーダ d について $d = 1, \ldots, 5$ について評価している．なお，ここでの鍵スケジュールには AES 単体の鍵スケジュール時間も含めてある（CMAC においては $E_K(0^{128})$ の計算を含める）．

表 2　参照ソースコードの実行速度（サイクル数）

MAC	鍵スケジュール	タグ生成
PC-MAC-AES$_{(1)}$	1532	10158
PC-MAC-AES$_{(2)}$	3101	9204
PC-MAC-AES$_{(3)}$	4601	8803
PC-MAC-AES$_{(4)}$	6327	8614
PC-MAC-AES$_{(5)}$	7711	8396
CMAC-AES	759	12991

なお，オブジェクトファイルサイズは PC-MAC-AES が 60,715 バイト，CMAC-AES で 58,925 バイトとなった．共に，サイズのうち 46,167 バイトは AES ソースの部分である．

なお，今回の実装は可読性と移植性を高めた参照ソースコードであり，その性能は，CMAC-AES との比でいって文献 [23] の実装より劣るものである．例えば，データの処理単位（1 バイト単位か 4 バイト単位か）や関数構成（パラメータの数や関数化の単位）などの点で冗長な部分を含んでいる．可読性を求めないコードの実装環境に応じた最適化により，速度の向上と実行ファイルサイズの縮小が可能であると考えられる．

4.3 ハードウェア実装性

ハードウェアにおける実装も，AES のハードウェア記述言語（HDL）を利用して効率よく実装することが可能である．ここでは，Verilog-HDL で PC-MAC-AES$_{(1)}$ を FPGA 上に実装した結果を記す．詳細は参照ハードウェア設計記述仕様書 [3] を参照のこと．

開発環境と，合成結果は以下のようになった．

■開発環境

- ターゲットデバイス：Virtex5 xc5vlx50-3
- 設計ツール：Xilinx ISE ver9.2i
- 合成オプション: 速度優先，effort high 指定，そのほかはデフォルト

■合成結果

- スライス数 4356
- 最高動作周波数 128MHz

なお，参照ハードウェア設計記述仕様書にも記述のように，使用した AES の HDL は S-box 実装がややASIC に向いたものになっており，FPGA に特化することで動作速度とサイズを改善することが可能と予測される．また，同じ開発環境における ECB モード AES との速度比は約 1.37 倍，スライス数の比は約 1.4 倍と

なった．これらの数値も今後の実装手法の改良やターゲットデバイスへの特化により改善する可能性があると思われる．さらに，3.6 節で述べたように，AES へのサイドチャネル攻撃への汎用的対策を組み込むことも可能である．

参考文献

[1] 暗号技術仕様書 PC-MAC-AES, 日本電気株式会社, 2010.

[2] 参照ソースコード仕様書 PC-MAC-AES, 日本電気株式会社, 2010.

[3] 参照ハードウェア設計記述仕様書 PC-MAC-AES, 日本電気株式会社, 2010.

[4] Intel Advanced Encryption Standard (AES) Instructions Set - Rev 3, http://software.intel.com/en-us/articles/intel-advanced-encryption-standard-aes-instructions-set/

[5] V. Rijmen, A. Bosselaers, and P. Barreto. Optimised ANSI C code for the Rijndael cipher ver.3.0.

[6] 岩田 哲. “ブロック暗号利用モードの安全性に関する調査.” CRYPTREC 技術報告書, 2003. http://www.ipa.go.jp/security/enc/CRYPTREC/fy15/documents/rep_ID0204.pdf

[7] NIST FIPS-197. http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[8] NIST Special Publication 800-38B, Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication. http://csrc.nist.gov/publications/nistpubs/800-38B/SP_800-38B.pdf

[9] 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, *RIPE Integrity Primitives*, final report of RACE Integrity Primitives Evaluation. 1995.

[10] M. Bellare, O. Goldreich, and A. Mityagin. “The Power of Verification Queries in Message Authentication and Authenticated Encryption.” *Cryptology ePrint Archive*, 2004/309.

[11] D. J. Bernstein. “The Poly1305-AES Message-Authentication Code.” *Fast Software Encryption*, FSE’05, LNCS 3557, pp. 32-49, 2005.

[12] A. Biryukov, A. Bogdanov, D. Khovratovich, and T. Kasper. “Collision Attacks on AES-Based MAC: Alpha-MAC.” *Cryptographic Hardware and Embedded Systems*- CHES ’07, LNCS 4727, pp.166-180, 2007.

[13] J Daemen and V. Rijmen. “A New MAC Construction ALRED and a Specific Instance ALPHA-MAC.” *Fast Software Encryption*, FSE’05, LNCS 3557, pp. 1-17, 2005.

[14] J Daemen and V. Rijmen. “The Pelican MAC Function.” *IACR ePrint Archive*, 2005/088.

[15] S. Halevi and H. Krawczyk. “MMH:Software Message Authentication in the Gbit/second rates.” *Fast Software Encryption*, FSE’97, LNCS 1267, pp. 172-189, 1997.

[16] T. Iwata. “Comments on “ On the security of XCBC, TMAC and OMAC ” by Mitchell.” csrc.nist.gov/groups/ST/toolkit/BCM/documents/comments/.../Iwata3.pdf

[17] T. Iwata and K. Kurosawa. “OMAC: One-Key CBC MAC.” *Fast Software Encryption*- FSE’03, LNCS 2887, pp. 129-153, 2003.

[18] K. Jia, X. Wang, Z. Yuan, and G. Xu. “Distinguishing Attack and Second-Preimage Attack on the CBC-like MACs.” *Cryptology ePrint Archive*, Report 2008/542.

[19] L. Keliher and J. Sui. “Exact Maximum Expected Differential and Linear Probability for 2-Round Advanced Encryption Standard (AES).” *IACR ePrint Archive*, 2005/321.

[20] L. Keliher and J. Sui. “Exact maximum expected differential and linear cryptanalysis for two-round Advanced Encryption Standard.” *IET Information Security*, Vol. 1, No. 2, pp. 53-57, June. 2007.

[21] K. Kurosawa and T. Iwata. “TMAC: Two-Key CBC MAC.” *Topics in Cryptology*- CT-RSA 2003, LNCS 2612, pp. 33-49, 2003.

[22] M. Liskov, R. Rivest, and D. Wagner. “Tweakable Block Ciphers.” *Advances in Cryptology*-CRYPTO’02, LNCS 2442, pp. 31-46, 2002.

[23] K. Minematsu and Y. Tsunoo. “Provably Secure MACs From Differentially-uniform Permutations and AES-based Implementations.” *Fast Software Encryption*, FSE ’06, LNCS 4047, pp. 226-241, 2006.

[24] C.J. Mitchell. “On the security of XCBC, TMAC and OMAC.” *Technical Report*, RHUL-MA-2003-4, 19, 2003. `http://www.rhul.ac.uk/mathematics/techreports`

[25] S. Nikova and C. Rechberger, and V. Rijmen, “ Threshold Implementations Against Side-Channel Attacks and Glitches, ” The 8th International Conference on Information and Communications Security (ICICS 2006), LNCS4307, pp. 529-545, Springer-Verlag, Dec. 2006.

[26] T. Pop and S. Mangard, “ Masked Dual-Rail Pre-charge Logic: DPA-Resistance without Routing Constrain, ” Workshop on Cryptographic Hardwar and Embedded Systems (CHES2005), LNCS 3659, pp. 172-186, Springer-Verlag, Aug. 2005.

[27] S. Gueron. “Intel’s New AES Instructions for Enhanced Performance and Security.”, *Fast Software Encryption*, FSE ’09, LNCS 5665, pp. 51-66, 2006.

[28] D. Suzuki, M. Saeki, and T. Ichikawa, “ Random Switching Logic: A New Countermeasure against DPA and Second-Order DPA at the Logic Level, ” IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol. E90-A, no. 1, pp. 160-168, Jan. 2007.

[29] E. Trichina, “ Combinational Logic Design for AES SubByte Transformation On masked Data, ” Cryptology ePrint Archive, 2003/236, 2003.

[30] K. Tiri and I. Verbauwhede, “ A Logic Level Design Methodology for a Secure DPA Resistant ASIC or FPGA Implementation, ” Proc. 2004 Design, Automation and Test in Europe Conference and Exposition (DATE 2004), pp. 246-251, Feb. 2004.

[31] Z. Yuan, W. Wang, K. Jia, G. Xu, X. Wang: “New Birthday Attacks on Some MACs Based on Block Ciphers.” *Advances in Cryptology*, CRYPTO ’09, LNCS 5677, pp. 209-230.