Vol.2015-DPS-162 No.28
Vol.2015-CSEC-68 No.28
2015/3/5

組込みシステムにおける認証付暗号手法のソフトウェア実装評価

小手川　誠[1,a)]　岩井　啓輔[1]　田中　秀磨[1]　黒川　恭一[1]

概要：認証付暗号の開発・評価として Competition for Authenticated Encryption: Security, Applicability, and Robustness(以下、CAESAR) が実施されている。ここでは、47 個のアルゴリズムが提案され、安全性及びソフトウェア・ハードウェア実装性能が評価されている。本稿では、CAESAR に提出された候補のうち、AES を利用した候補を Nonce Based 及び Nonce-misuse Resistance に分類し、各分類 2 候補の計 4 候補（SILC、AES-OTR、AES-COPA、POET）に注目した。さらに認証付暗号が、組込みシステムネットワークの安全性向上策の 1 つに挙げられていることに注目し、ARM プロセッサをベースとし、平文を 16Bytes から 2048Bytes までとしたときの処理性能の評価を実施した。また、使用する鍵の変更が与える影響も評価した。測定した項目を、プログラムサイズ、入力データ量、処理速度として、組込み用途に適したものを考察した。

キーワード：組込みシステム，認証付暗号，AES，CAESAR，CAN

1. はじめに

一般的な暗号通信では、暗号文の改ざんが検知できず、改ざんされた暗号文を復号し、セキュリティ上の問題が発生する可能性がある。この問題を解決する手段の一つに、認証付暗号がある。認証付暗号は、暗号化する平文の他に、Nonce や Associated Data などを入力し、暗号文と Tag と呼ばれるデータを出力する。Tag は平文又は暗号文に依存するように生成される。送信した暗号文を改ざんすると Tag の値と一致しないため、復号時の Tag との整合により、暗号文改ざんの検知が可能である。様々な機器がネットワークに接続されている今日、制御信号を改ざんする攻撃への対策の一つとして、認証付暗号は注目されている。

認証付暗号を更に発展させる取り組みとして、公的機関によるプロジェクトである CAESAR(Competition for Authenticated Encryption: Security, Applicability, and Robustness) がある [1]。CAESAR では認証付暗号を公募し、総数 47 個のアルゴリズムが提案されている。そのうち 17 アルゴリズムが、暗号プリミティブとして AES を採用している。更にいくつかのアルゴリズムが、AES の要素技術を応用した提案となっている。CAESAR の候補は、毎年夏に行われる DIAC（Direction in Authenticated Ciphers）での選考を経て、2017 年 12 月に最終選考により採用候補が決定される予定である。

現在、CAESAR に提案されている各アルゴリズムについて、処理速度や安全性などが評価されている.。A.Bogdanov らは、暗号プリミティブに AES を使用した認証付暗号アルゴリズムの処理速度を測定・評価した [2]。この評価は、インターネット上での通信を想定している。具体的には、1) インターネット上の通信で使用されるパケットのうち、81%が 40Bytes から 1500Bytes であることに着目し、平文長を 128Bytes から 2048Bytes に設定したこと、2)AES-NI[*1] を搭載し、AVX[*2][3] が使用可能な Haswell アーキテクチャ [*3] を採用した CPU を実験に使用したこと、が挙げられる。

一方、認証付暗号はインターネットだけでなく、組込みシステムでも注目されている。組込みシステムとは、ECU(Electronic Control Unit) で機器を制御するコンピュータシステムである。自動車や医療機器、船舶制御、家電等多くの製品で採用されおり、これからも利用用途の発展が予想されている [4]。組込みシステムは、内部ネッ

[1] 防衛大学校
National Defense Academy, Hashirimizu, Yokosuka, Kanagawa, 239-0811 Japan
[a)] em53037@nda.ac.jp

[*1] AES New Instruction:AES による高速な暗号化・復号を目的に、Intel 社 CPU に実装されている命令セット
[*2] Intel Advanced Vector eXtensions : Intel 社の開発した拡張命令セットであり、最大 256bit の演算幅で演算可能
[*3] 2014 年当時、Intel 社最新の CPU アーキテクチャ。

トワークを介して、制御信号やセンサーの情報をやり取りしている。この制御信号等を改ざんされると、自動車のブレーキ誤作動や薬剤の過剰投与などの致命的な結果に至る可能性がある。そのため、組込みシステムでも認証付暗号は注目されている。本研究では、このような組込みシステムの通信環境と、PC 間の通信環境の差異に注目し、組込みシステムを想定した処理速度、プログラムサイズ及び入力データの総量を比較した。その結果から、組込みシステム用途に適する認証付暗号を考察した。

2. CAESAR

2.1 候補の分類

CAESAR の候補は、F.Abed[5] により、暗号プリミティブに AES を採用しているものと、そうでないものに分類される。暗号プリミティブに AES を採用した場合、共通鍵暗号の代表として幅広く使用されていることや、AES-NI を利用して高速に処理が可能なことから、実装面で有利となる。さらに、A.Bogdanov [2] は、Nonce Based と Nonce-misuse Resistance に分類した (表 1)。なお、表中 ($^{\neq}$) は Nonce Based を、($^{=}$) は Nonce-misuse Resistance を表す。それぞれに以下のような特徴がある。

Nonce Based(NB)

Nonce を必要とする。平文 1 ブロックの暗号化に必要な暗号プリミティブ実行回数が 1 回。

Nonce-misuse Resistance(NR)

Nonce を必要としない。平文 1 ブロックの暗号化に必要な実行回数は、2 回以上。

これらの特徴から、Nonce Based は、暗号化処理で実行する AES の回数が少ないため、速度面で Nonce-misuse Resistance に対し有利となる。Nonce-misuse Resistance は、Nonce を使用しないことから、実装時のメモリ削減に貢献し、Nonce に対するぜい弱性がないというセキュリティ上の有利な点がある。本研究も、これらの分類に従い、評価対象を選定した。

2.2 先行研究との関係

A.Bogdanov らは、インターネットによる PC 間の通信を想定した実験環境で、処理速度の測定を実施した [2]。この測定には、以下のような特徴がある。

- AES-NI を利用した高速実装。
- Nonce Based/Nonce-misuse Resistance に分けて比較評価。
- 扱う平文長は 128〜2048Bytes。

この実験の結果、AES-OTR (NB)[6][*4]、AES-COPA (NB)[7] が最速であるであると評価された。一方、本研究での処理速度測定では、組込みシステムにおける、ECU 間の通信を想定した実験環境を設定した。そのため、以下のような制約及び条件の変化が発生する。

- AES-NI や AVX 命令の使用不能。
- 動作周波数の低下（GHz 程度から MHz 程度へ）。
- 扱う平文長がインターネット用途よりも短い。

従って、AES-OTR や AES-COPA が、必ずしも実装有利なアルゴリズムではないと考えられる。

[*4] A.Bogdanov の行った実験では、平文長 128Bytes で AES-OTR、2048Bytes で OCB の処理速度が最も優れていた。本研究で想定している平文長は 16Bytes 程度であることから、16Bytes に近い 128Bytes で優れた処理性能を示した AES-OTR を選択した。

表 1　文献 [5] における CAESAR 候補の分類

	AES 又はその要素技術を暗号プリミティブに使用	AES 以外を暗号プリミティブに使用
Nonce Based	AES-CMCC, AES-CPFB, AES-AEGIS, **AES-OTR**, AVALANCHE, CBA, CLOC, Deoxys($^{\neq}$), Joltik($^{\neq}$), Julius-CTR, KIASU($^{\neq}$), OCB, Prøst-OTR, **SILC**, Silver, Tiaoxin, YAES	AMORUS, ACORN, Calico, Enchilada, iSCREAM, KetjeKeyak, LAC, NORX, PRIMATEs-GIBBON, PRIMATEs-HANUMAN, Raviyoyla, Sablier, SCREAM, STRIBOB, TriviA-ck, Wheesht, π-cipher,
Nonce-misuse Resistance	++AE, **AES-COPA**, AES-JAMBU, AEZ, Deoxys($^{=}$), ELmD, iFeed[AES], Joltik($^{=}$), Julius-ECB, KIASU($^{=}$), PAEQ, **POET**, Prøst-COPA, Prøst-APE, SHELL	HS1-SIV, Minalpher, Artemia, Ascon, ICEPOLE, PRIMATE-APE

2.3 評価対象の選択

本研究では、上述したようにインターネット環境での評価が、組込みシステム環境でも活用できるかを考察する。従って、本研究での評価対象として、まず AES-OTR

と AES-COPA を挙げる。これらは AES を利用しているので、本研究でも AES を利用している候補を対象とする。SILC(NB)[8] は、提案者によると、利用する関数の種類を減らし、かつ単純にすることにより、組込み環境を意識した設計がなされている。従って本研究の評価対象とした。POET(NR)[9] は、暗号化処理で使用する Universal Hash Function に Full-Round AES または 4-Round AES を使用する。A.Bogdanov の評価では、Full-Round AES を採用している。従って、4-Round AES を使用した場合、COPA に比べて処理速度で優れることが見込まれる。一方、初期化の際に、AES を 5 回実行する特徴がある。このため、短いメッセージを、毎回異なる鍵で暗号化する場合は、処理速度の低下が予想される。この 2 点から、本研究の評価対象とした。以下、本研究で評価対象とした各アルゴリズムの概要を述べる。

AES-OTR(NB)[6]

Feistel 構造を応用し、Nonce に依存する変数 $\delta = E_k(Nonce)$ 及び $L = 4\delta$ を利用する。暗号化及び復号では 2 ブロックごとに並列化処理が可能である。Tag 生成時の処理は、AES の実行回数が平文長に依存せず、1 回の実行で生成する (図 1(a))。以下 OTR と略記する。

SILC（NB）[8]

CFB モードを応用し、暗号化及び Tag 生成は並列処理が不可能であるが、復号時は可能である。SILC 独自の定数として、Associated Data 及び暗号文のバイト長 Len(A),Len(C) を使用する。Tag 生成部の構造上、AES 実行回数が、平文のブロック数に比例して増大する。そのため、数ブロックに及ぶデータの処理では AES 実行回数が多くなるため、同じく Nonce Based に属する AES-OTR よりも遅い [*5]。一方、組込み環境を意識して、関数が単純化され、変数の種類が他の候補と比べて削減されている (図 1(b))。その結果、プログラムサイズの評価で有利と予想される。

AES-COPA(NR)[7]

平文 1 ブロックの暗号化処理に対し、2 回の AES の実行を必要とする。暗号化及び復号処理は並列化可能で、使用される変数 L の初期値は、$L = E_k(0)$ となる。Tag 生成時の AES 実行回数は、平文長に依存せず 2 回である (図 1(c))。以下 COPA と略記する。

POET（NR）[9]

平文 1 ブロックの暗号化処理に対し、1 回の AES の実行と、2 種類の Universal Hash Function E_{k2}, E_{k3}(4-Round または Full-Round AES) で生成された値を使用する。1 ブロック目の暗号化処理の際、2 つの Universal Hash Function に入力する初期値は、鍵付ハッシュ値 V を使用する。Tag 生成時は 1 回の AES 処理と、2 種類の Universal Hash Function による処理を各 1 回ずつ行う。Universal Hash Function に 4-Round AES を使用した場合、同じ NR に属する COPA に比べて、AES の必要処理ラウンド数を減らすことができる。従って、処理速度の評価で有利と予想される。一方、鍵の変更による初期化の際に、AES を 5 回実行する必要がある。そのため、頻繁な鍵の更新が伴う場合や、処理する平文長が短い場合には、処理速度の低下が予想される (図 1(d))。

[*5] アルゴリズム全体での、AES 実行回数は各仕様書で、以下のように定義されている。
|A|: Associated Data の長さ [Byte], |M|: 平文の長さ [Byte]
n: ブロック長 [Byte]
AES-OTR: $\lceil |A|/n \rceil + \lceil |M|/n \rceil + 2$
SILC : $\lceil |A|/n \rceil + 2\lceil |M|/n \rceil + 2 + \lceil |N|/n \rceil$

3. 組込みシステム

組込みシステムは、複数の ECU とそれらをつなぐネットワークで構成される。ECU は、入力された命令やセンサーから得た情報をもとに、搭載されている機器装置を、制御するためのものである。汎用性よりもコストの削減を重視し、製品が必要とする性能に抑えられている (表 2)。また

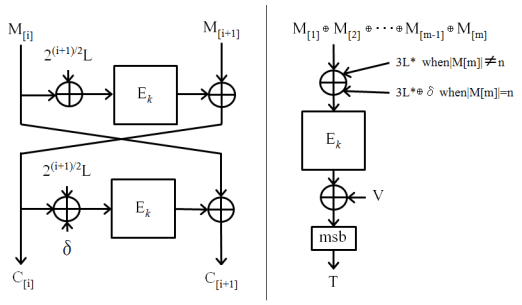
表 2 汎用な CPU と、ARM Coretex-A9 の性能比較

	Intel Core i5-4300U	ARM Cortex-A9
動作周波数	1.9GHz	667MHz
命令セット	64bit	32bit
AES 高速化	AES-NI	なし

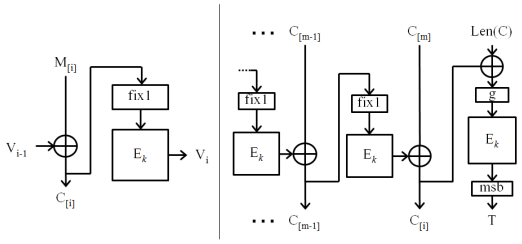
ECU は、それぞれを相互に接続することによって、1 つのシステムを構成している。ECU 同士の接続は、配線の複雑化やコストの増大、低い柔軟性といった欠点を解決するため、そのほとんどがネットワーク方式を採用している。組込みシステムに広く採用されているネットワーク方式の代表として、CAN(Controller Area Network) がある (表 3)。CAN におけるデータのやり取りは、データフレームを使用して行われる。1 データフレーム当たりのペイロードは 8Bytes であり、それ以上の情報は複数のデータフレームに分割する。ネットワークトポロジは、バス型（シリアルバス）で、ID によって優先順位を決定している。通信速度は最大 1Mbps となっている [*6]。CAN が開発された 1990 年代の時点では、CAN を外部ネットワークに接続することは想定されていなかった。しかしながら、現在ではカーナビゲーションやスマートフォンのように、インターネットに接続された製品が CAN とも接続されている。このように、組込みシステムが外部ネットワークとの接続することが一般的になっている一方で、外部からの攻撃を想定して設計されていない。従って、認証付暗号を含む様々な対策が検討されている。

[*6] Society of Automotive Engineers が定める規格

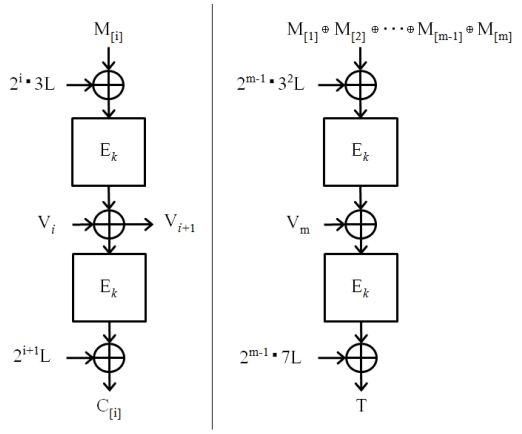
Vol.2015-DPS-162 No.28
Vol.2015-CSEC-68 No.28
2015/3/5



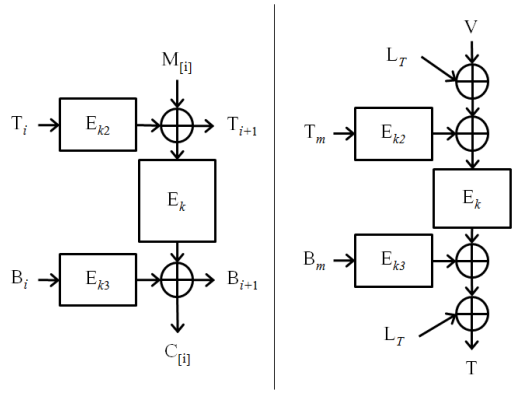
(a) OTR



(b) SILC



(c) COPA



(d) POET

M:平文, C:暗号文, T:Tag, Ek:AES, V:鍵付ハッシュ値

図 1 評価対象のアルゴリズム概略

表 3 CAN の特徴 [10]

ネットワークトポロジー	バス型 (シリアルバス)
データペイロード	1 フレーム当たり 8Bytes (8Bytes 以上のデータは分割送信)
通信速度	125Kbps〜1Mbps
接続可能機器数	理論上はなし (バスの遅延時間等による制限)
通信調停	ID の優先順位による調停

4. 実験環境

本研究では、組込みシステム用途での評価を目的とするため、第 3 節に示した組込みシステムの環境より、表 4 に示す実験環境を想定した。本実験で使用した CPU は、ARM 社製 Cortex-A9 シリーズ 32bit である。この CPU は、主にスマートフォンや自動車などの組込みシステムに採用されている。A.Bogdanov が処理速度測定で使用した CPU と比較すると、動作周波数や AES 処理速度の面で劣っている。本実験では、制御信号を対象とした認証付暗号の処理を想定しているので、並列処理は適さない。そのため、ARM は 2 コアであるが、これを用いた並列処理実装は行わない。また、本研究では組込みシステム環境を想定しているので、CAN のネットワーク仕様から、8Bytes 単位の通信を考える。ただし、認証付暗号で用いる暗号プリミティブとして、128bit ブロック暗号である AES を使用しているので、平文長を最小 16Bytes に設定した。さらに、A.Bogdanov の実験結果と比較するため、128Bytes〜2048Bytes の平文長に対する評価を行った。結果として、16Bytes〜2048Bytes までの平文処理に関して測定を行った。

表 4 実験環境

CPU	ARM Cortex-A9
動作周波数	667MHz
CPUbit 幅	32bit
言語等	C 言語（コンパイルに-O3 オプションを使用)
最小平文長	16Bytes

5. 実験方法

本研究では、組込みシステムにおける各候補の評価のため、以下で述べる方法でプログラムサイズ、入力データの総量及び処理速度を測定する。

5.1 プログラムサイズ

実装性を評価する手法の 1 つとして、作成した各候補の実行ファイルのプログラムサイズを測定した。Tag や暗号文生成が平文サイズに依存するので、その差を明確にするために、平文サイズごとにプログラムを作成し、そのサイズを測定した。なお、本研究で使用している AES は共通であり、プログラムサイズは、71.4KBytes である。速度は

ECB モードで 54.1[Mbps] である。

5.2 入力データ量

入力データは平文と鍵以外の、Associated Data、Nonce などを指す。実装する際、入力データ量が小さい方がメモリ量、プログラムサイズが小さくなることが予想される。また、入力データが小さいとき、送受信間で事前に共有すべきデータ量も小さくなるため、通信効率も高く攻撃への対策も容易になると考えられる。従って、なるべく入力データが小さい方がよい。なお、OTR と SILC については、その仕様書で Nonce の長さが複数推奨されている。本実験では、両候補の第 1 推奨パラメータである、96bit を採用した。

5.3 処理速度

処理速度は、組込みシステムの最大通信速度が 1Mbps であることから、これとの比較のために [bit/sec] を単位として評価を行う。ただし、秘密鍵の更新に関して以下の 2 通りの測定を行う。

[測定 1]

秘密鍵を固定する。そのため、拡大鍵生成や中間変数の生成に関わる初期化処理は 1 回だけ行う。

[測定 2]

秘密鍵を定期的に変更する。16〜2048Bytes の平文を、10000 回繰り返して処理し、1 回の処理が終わるごとに、秘密鍵を更新する。

測定 1 は、A.Bogdanov と同一の測定条件であり、実装環境の変化が実装性能に与える影響を考察する。測定 2 は、組込みシステムにおいて、最も安全な設定による運用を考えた時の、実装性能を測定する。

6. 実験結果

6.1 プログラムサイズ

各候補のプログラムサイズの測定結果を、表 5 に示す。プログラムサイズは、SILC が最も小さくなった。提案者らの主張のとおり、使用されている関数が単純かつ少ないため、プログラムサイズが小さくなったと考えられる。プログラムサイズが最も大きくなった POET との差は、約 9KBytes であった。利用環境が標準的な PC であれば、これは無視できる差である。しかし、組込みシステム環境では、実装できるメモリ資源の制約が用途によりさまざまである。そのため、KByte 単位の差が無視できない場合は、SILC が有利である。

表 5 評価対象のプログラムサイズ

	プログラムサイズ [KBytes]			
平文長	SILC	OTR	COPA	POET
16Bytes	86.1	91.8	92.6	97.2
32〜2048Bytes	90.9	93.0	93.1	97.5

6.2 入力データ量

入力データ量は NB よりも NR の方が小さい傾向がある (表 6)。このため、送受信間で、事前に共有すべきデータ量を削減する場合は、NR が有利である。さらに、NR は Nonce 生成のための乱数アルゴリズムの実装が不要なことや、Nonce 利用時特有の脆弱性がないという利点がある。一方、NB は Nonce に加え、その他の入力データを必要とする場合がある。例えば SILC の場合、Associated Data と暗号文の長さ Len(A) 及び Len(C) というそれぞれ 128bit のデータ量が追加的に発生する。ただし、Associated Data や暗号文長が一定である場合は、Len(A) 及び Len(C) は定数であり、入力データ量として扱う必要はない。しかし、暗号文長が可変の利用環境では、暗号化処理のたびに Len(A) 及び Len(C) の設定が必要になり、値を更新する必要がある。NR と比較すると、SILC は 360bit 大きい入力データ量を必要とする。仮に、暗号文 128bit、Tag128bit 出力を要求する利用環境の場合、事前に必要な入力データ量が出力よりも大きいという効率低下を引き起こす。従って、NB に関しては平文長が固定か可変かで SILC と OTR で評価が異なる。例えば、利用用途が固定平文長であれば、SILC と OTR は同じ結果となる。

表 6 評価対象の入力データ量

候補名	Associated Data	Nonce	その他	合 計
OTR	128bit	96bit	な し	224bit
SILC			Len(A), Len(C)	488bit
COPA		な し		128bit
POET				128bit

6.3 処理速度

処理速度の測定結果を、図 2,3 及び表 7,8 に示す。実験全体では、AES 実行回数の少ない OTR がいずれの平文長においても優れた結果となった。OTR と同じ NB に属する SILC は、Tag 生成部の AES 実行回数がブロック数に比例する。これにより NB が NR に対し、暗号化に必要な AES 実行回数が少なくなるという長所が相殺されるため、OTR よりも NR である COPA や POET に近い処理速度となった。また、各候補とも平文長が短くなるほど処理速度が低下し、その差が小さくなることが確認できた。

ただし、現在代表的な組込みシステムネットワーク方式である、CAN の最大通信速度は 1Mbps である。これを考慮すると、平文長 16Bytes で最も処理速度の遅かった POET でも約 5Mbps(測定 2) であり、必要な処理速度は満たしている。

次に、NR に属する COPA と POET に注目する。測定 1 では、暗号化及び Tag 生成で実行する AES の総ラウン

表 7　測定 1

	処理速度 [Mbps]			
平文長	SILC	OTR	COPA	POET
16Byte	9.1	10.8	8.9	9.9
32Byte	15.5	17.8	13.3	14.9
64Byte	19.7	26.7	17.8	20.3
128Byte	23.0	35.7	21.4	24.4
256Byte	25.0	42.6	23.8	27.4
512Byte	26.2	47.3	25.3	29.1
1024Byte	26.7	50.0	25.9	30.0
2048Byte	27.1	51.7	26.3	30.5

表 8　測定 2

	処理速度 [Mbps]			
平文長	SILC	OTR	COPA	POET
16Byte	8.6	10.2	8.5	4.9
32Byte	14.8	17.0	12.8	8.4
64Byte	19.1	25.8	17.4	13.2
128Byte	22.5	35.0	21.2	18.5
256Byte	24.6	41.9	23.6	23.1
512Byte	25.9	46.8	25.1	26.5
1024Byte	26.7	49.6	25.9	28.5
2048Byte	27.1	51.3	26.3	29.7

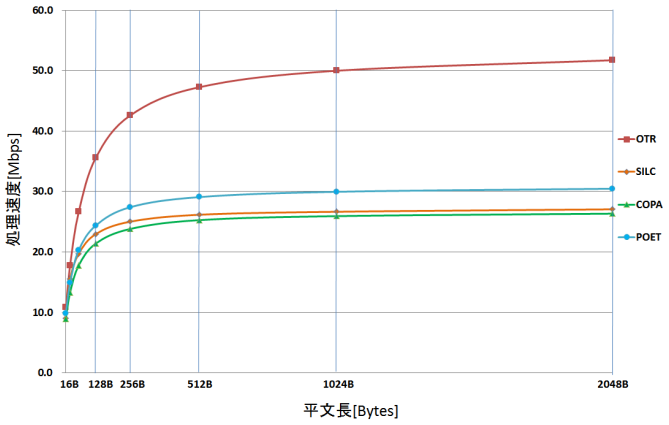


図 2　測定 1

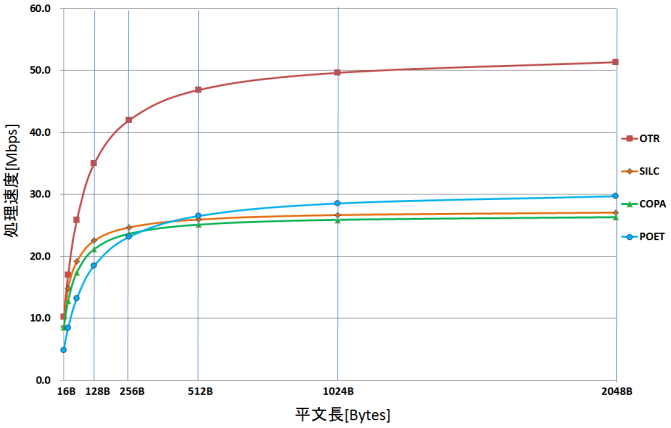


図 3　測定 2

ド数が少ない POET が、COPA を上回っている。しかし、測定 2 では平文長が短くなるにつれ、POET の処理速度が COPA よりも低下する。これは POET の鍵に依存する初期化が、COPA に比べて 4 回多く AES を実行するためである。また、測定 2 で POET が 256Bytes 以下の平文を処理する際に、急激に処理速度が低下する。これは、初期化で実行される AES の回数と、暗号化及び Tag 生成で実行される AES の回数の比率が関係している (表 9)。平文が短いときは、初期化における AES 実行回数が、暗号化及び Tag 生成に対して多くなるため、処理速度が低下する。逆に、初期化で実行される AES の回数が、暗号化及び Tag 生成で実行される AES の回数よりも十分小さい場合は、鍵変更による初期化処理が全体の処理速度に対してほとんど影響を与えない。そのため、鍵変更の影響を受けにくい平文長 256〜2048Bytes の範囲では、POET が COPA に処理速度で勝る。ただし、組込みシステムをよりセキュアにするには、測定 2 の条件の方が適している。特に、組込みシステムでの制御系信号のデータ長は比較的短いことから、NR では COPA が適している。

7. 考察

まず、処理速度に関する本研究の実験結果を考察する。測定 1 で、得られた処理速度と A.Bogdanov らの測定結果を比較する。処理速度の特性は、平文長が短くなるほど、低下するという点で A.Bogdanov の結果 [2] と同じであった。一方 COPA、POET 及び SILC 間の優劣関係が、A.Bogdanov の研究結果と変化した。A.Bogdanov の実験では処理速度が早い順に、OTR、SILC、COPA、POET であった。本研究では、OTR、POET、SILC、COPA の順であった。POET に関しては、A.Bogdanov は Universal Hush Function に Full-round AES を採用しているのに対し、本研究では提案者らが推奨する 4-round AES を採用した。このため、本研究での POET の処理速度は A.Bogdanov と比較して、高速な結果となった。一方 POET 以外は、A.Bogdanov と同様の仕様による実装であり、採用した関数に変化はない。このことから、POET が有利な結果となっているものの、PC 環境と組込みシステム環境で、鍵変更を行わない実装の場合は　優劣関係に大きな影響を与えないと予想できる。

表 9　16Bytes 平文処理時の、初期化と暗号化及び Tag 生成で実行される AES の回数

	AES 実行回数	
	初期化	暗号化及び Tag 生成
COPA	1 回	4　回
POET	5 回	3.6 回

測定 2 では、組込みシステムにおいて安全な運用を考え、定期的な鍵変更を行った。処理速度の特性は測定 1 と変わ

らなかったものの、POET が、平文長 16〜256Bytes の範囲で急激に処理速度が低下した。これは、第 6.3 節で示したように、鍵に依存する初期化で実行される AES の回数が、他の候補よりも多かったためである。このことから、平文長が短く、鍵変更を伴う通信環境では、初期化で実行される AES の回数が、処理速度に大きな影響を与えることが分かった。

次に、本研究における実験結果から、プログラムサイズ、初期設定データ量及び処理速度に対する、各候補の評価から、組込みシステムに適した候補の選択を行う。各候補の実装の特徴を表 10 に示す。

表 10 各評価対象の実装の特徴

候補名	実装の特徴
OTR	• AES 実行回数が少なく、処理速度が速い。
SILC	• 関数が単純かつ少ないため、プログラムサイズが小さい。 • 同じ NB に属する他の候補に比べて処理速度が遅い。
COPA	• Nonce を使用する必要がない。
POET	• Nonce を使用する必要がない。 • 平文長が短い場合、鍵の変更による処理速度低下への影響が大きい。

数 KBytes 程度のメモリ資源制約が無視できない場合は、プログラムサイズを優先して SILC を選択する。ただし、Nonce を動作させるための、乱数生成アルゴリズムのプログラムサイズとの関係に注意しなければならない。逆に、乱数生成アルゴリズムの実装を避け、Nonce を使用せずに認証付暗号を使用する場合は、NR である COPA 及び POET が有利となる。ただし POET は、平文長が短く、鍵の変更がある処理については、処理速度が著しく低下することに注意する必要がある。従って、鍵更新の頻度と平文長の関係に注意しなければならない。OTR は、他の候補に比べ処理速度の早さが目立つが、現行の CAN における通信条件では、処理速度が他の候補に対して有利とならない。本実験環境で、処理速度の速さが利点となるのは、CAN の最大通信速度が 30Mbps を超えるときである。従って、CAN の通信速度が、現行の 30 倍以上高速になる場合、他の候補に対して有利となる。

8. おわりに

組込みシステムはそれぞれ、用途、規模、コストの設定が様々であり、認証付暗号についても、そのシステム毎に最適なものを選択する必要がある。今後も CAESAR の選考委員により、候補が絞られていくことになる。その選考においては、処理速度、安全性などを総合的に評価して選考するだけでなく、利用条件ごとケースを分類し、それぞれのケースで求められている特徴は何かという、視点を持つことも重要である。

また、本研究については、FPGA などハードウェア実装により、消費電力や実装面積について検討し、選択を行うためのケースについて考察する予定である。

参考文献

[1] *CAESAR call for submissions, final*, CAESAR committee, available from ⟨http://competitions.cr.yp.to/caesar-call.html⟩ (2014.01.27).
[2] Andrey Bogdanov, Martin M. Lauridse, and Elmar Tischhauser.: *AES-Based Authenticated Encryption Modes in Parallel High-Performance Software*,Cryptology ePrint Archive:Report 2014/186, available from ⟨http://eprint.iacr.org/2014/186⟩(2014).
[3] *Intel 64 and IA-32 Architectures Software Developer's Manual*, available from ⟨http://www.intel.co.jp /content/www/jp/ja/processors/architectures-software-developer-manuals.html⟩(2014.6).
[4] 株式会社　矢野経済研究所 : 車載ネットワーク用デバイス世界市場に関する調査結果 2013 〜ECU の搭載個数増加、ISO26262 の影響により市場規模は拡大〜(online), 入手先 ⟨http://www.yano.co.jp/press/pdf/1201.pdf⟩(2014.1.21).
[5] Farzaneh Abed, Christian Forler and Stefan Lucks.: *Classification of the CAESAR Candidates*, Cryptology ePrint Archive:Report 2014/792. available from ⟨http://eprint.iacr.org/2014/792⟩(2014).
[6] Kazuhiko Minematsu.:*AES-OTR*,available from ⟨http://competitions.cr.yp.to/caesar-submissions.html⟩ (2014).
[7] Elena Andreeva, Andrey Bogdanov, Atul Luykx, Bart Mennink, Elmar Tischhauser, and Kan Yasuda.:*AES-COPA*,available from ⟨http://competitions.cr.yp.to/caesar-submissions.html⟩ (2014).
[8] Tetsu Iwata, Kazuhiko Minematsu, Jian Guo, Sumio Morioka, and Eita Kobayashi.: *SILC: SImple Lightweight CFB*,available from ⟨http://competitions.cr.yp.to /caesar-submissions.html⟩ (2014).
[9] Farzaneh Abed, Scott Fluhrer, John Foley, Christian Forler, Eik List, Stefan Lucks, David McGrew, and Jakob Wenzel.: *The POET Family of On-Line Authenticated Encryption Schemes*, available from ⟨http://competitions.cr.yp.to/caesar-submissions.html⟩ (2014).
[10] Robert Bosch GmbH.: *CAN Specification Version 2.0*, pp.42-55 (1991.4.5).