AES-NIを用いたラージブロック暗号の効率的な構成

芝　廉太朗[1]　阪本　光星[1]　五十部　孝典[1,2]

概要：256 bits や 512 bits のブロック長を持つラージブロック暗号が，長期的なセキュリティの確保の観点で注目されている．既存のラージブロック暗号である Haraka-v2 や Pholkos では，高速なソフトウェア実装のため，AES-NI と SIMD 命令で効率的に実行可能な word 置換のみで構成されている．Haraka-v2 や Pholkos では，1 ステップで AES ラウンド関数の並列実行を 2 回行い，その出力を word 置換する構造 (2 ラウンド構成) を採用している．本研究では，このような構成のラージブロック暗号に対して，実行速度の観点で最適な構成について検討する．具体的には，効率良く SIMD 実装可能な word 置換のクラスの中から，より少ないラウンドで安全性を達成可能な最適な word 置換のクラスを，混合整数線形計画法を用いた探索により明らかにしたのち，CPU のアーキテクチャ毎にそれらの速度を計測する．結果として，既存研究とは異なり，256-bit の場合は，特定の CPU アーキテクチャでは 1 ステップで AES ラウンド関数を 1 回のみ実行し，その出力を word 置換する構造 (1 ラウンド構成) が最適であることを示す．また，512-bit の場合は，アーキテクチャによらず，2 ラウンド構成が最適であることを示す．さらに，同じサイクル数が必要な word 置換命令であってもアーキテクチャによっては明確な速度差がでることを示し，アーキテクチャ毎の最適な構成方法を明らかにする．

キーワード：AES-NI，ラージブロック暗号，置換，ラウンド関数

Efficient Constructions of Large-state Block Ciphers Using AES-NI

Rentaro Shiba[1]　Kosei Sakamoto[1]　Takanori Isobe[1,2]

Abstract: Large-state block ciphers with 256-bit or 512-bit block sizes receive much attention from the viewpoint of the long-term security. Existing large-state block ciphers, such as Haraka-v2 and Pholkos, consist of only AES-NI and a word shuffle that can be efficiently executed by SIMD instructions for fast software implementation. In Haraka-v2 and Pholkos, the AES round function is executed twice in parallel in each step and its outputs are shuffled (called two-round constructions). In this paper, we explore the optimal constructions based on AES-NI and efficient word shuffles for such a large-state block cipher in terms of execution speed. Specifically, after identifying an optimal class of word shuffles that can achieve the security in fewer rounds from among the classes of word shuffles that can be efficiently implemented in SIMD, and then measure their speed for each CPU architecture. As a result, in contrast to existing results, we reveal that the constructions such that only one-round of AES round function is executed in parallel in one step and its outputs are shuffled (called one-round constructions) is optimal in some architectures for each 256-bit block size. We also show that two-round constructions are always optimal for 512-bit block size, regardless of the architecture. Furthermore, we find that there is a clear difference in the speed of word shuffle instructions, even if they require the same number of cycles depending on the architecture, and clarify the optimal construction for each architecture.

Keywords: AES-NI, large-state block cipher, permutation, round function

[1] 兵庫県立大学大学院応用情報科学研究科，〒 650-0047 兵庫県神戸市中央区港島南町 7-1-28，University of Hyogo, 7-1-28, Minatojimaminami-cho, Chuo-ku, Kobe-shi, Hyogo, 650-0047, Japan.

[2] 国立研究開発法人 情報通信研究機構，〒 184-8795 東京都小金井市貫井北町 4-2-1, National Institute of Information and Communications Technology, 4-2-1, Nukui-Kitamachi, Koganei-shi, Tokyo, 650-0047, Japan.

1. はじめに

大きなブロック長を持つラージブロック暗号が，長期的なセキュリティ確保の観点で注目されている．Kölbl らは 2016 年に，AES-NI [1] と呼ばれる，AES ラウンド関数と鍵生成アルゴリズムをサポートする SIMD 命令を用いて，AES ラウンド関数を並列実行し，その出力の word 置換を数ステップ繰り返す構造を持つハッシュ関数 Haraka-v2 を提案した [2]．Bossert らは 2020 年に，Haraka-v2 を参考に，ブロック暗号 Pholkos を提案しており [3]，これは AES ラウンド関数の並列実行によって実質的なブロック長が 256 bits, 512 bits のラージブロック暗号となっている．どちらも，2 ラウンドの AES ラウンド関数と 32-bit を 1-word とした word 単位の置換を 1 ステップとして，複数ステップ繰り返す構造になっている．AES ラウンド関数の並列実行により，大きなブロック長を扱うことが可能なため，128-bit セキュリティ以上の安全性を担保できる．また，Haraka-v2 と Pholkos では置換の実現に用いられている命令は異なり，Haraka-v2 では punpckldq と punpckhdq，Pholkos では vpblendd がそれぞれ用いられている．Pholkos の設計者は vpblendd が punpckldq と punpckhdq より高速であることを選択理由にしている [3]．Pholkos と Haraka-v2 では，安全性評価を容易に行うため，1 ステップに 2 ラウンドの AES ラウンド関数を用いる 2 ラウンド構成となっているが，この手法が実行速度，安全性の観点から最適であることは示されていない．

本稿では，このようなラージブロック暗号を，従来のように 1 ステップ中に AES ラウンド関数を 2 ラウンド実行せず，1 ラウンドの実行のみで実現する 1 ラウンド構成でも，同等の実行速度で，差分/線形攻撃に対して同等の安全性を達成できるような置換の探索を混合整数線形計画ソルバーを用いて行った．また，1 ステップ中の AES ラウンド関数の実行が 1 ラウンドの場合と 2 ラウンドの場合，置換に punpckldq と punpckhdq を用いた場合と vpblendd を用いた場合のそれぞれのラージブロック暗号を実装し，実行速度を計測することで，1 ラウンド構成と 2 ラウンド構成ではどちらが最適か，置換の命令として，どちらを用いるのがより最適かを比較した．結果として，次のことを明らかにした．

- 1 ラウンド構成でも，同等の実行速度で，差分/線形攻撃に対して同等の安全性を達成できるような置換が存在する．
- ブロック長が 256-bit の場合，特定のアーキテクチャにおいて，1 ラウンド構成の方が 2 ラウンド構成より高速であり，512-bit の場合は全てのアーキテクチャにおいて 2 ラウンド構成の方が高速である．
- 殆どのアーキテクチャにおいて，置換の命令として punpckldq と punpckhdq を用いた場合の方が vpblendd を用いた場合よりも高速である．これは，Pholkos の設計者の主張を覆す結果である．

これらの結果から，CPU のアーキテクチャ毎のラージブロック暗号の最適な構成方法を明らかにする．

本稿の構成を以下に示す．2 章で基礎事項として，AES-NI，既存構成である Haraka-v2，Pholkos の概要について説明し，Active S-box による差分/線形攻撃の安全性評価について説明する．その後で，1 ステップ中の AES ラウンド関数を 1 ラウンドで構成する動機について述べる．3 章では，1 ラウンド構成の提案を行う．4 章では，様々なクラスの置換を用いた場合の 1 ラウンド構成と 2 ラウンド構成に対して行った安全性評価の結果を示す．5 章では実装による速度評価の結果を示し，6 章でまとめを述べる．

2. 準備

本章では，本研究に関する予備知識として，AES-NI と先行研究において提案されている Haraka-v2，Pholkos について説明し，Active S-box による差分/線形攻撃の安全性評価について説明する．その後で，既存構成の問題点について指摘し，ステップ関数を 1 ラウンドの AES ラウンド関数で構成する動機について述べる．

2.1 AES-NI

AES-NI (AES New Instructions set) [1] はベクトルレジスタに複数のデータを保存し，複雑な計算を一度に行うことができる SIMD 命令群の一種であり，AES ラウンド関数とその逆関数を実行する命令とラウンド鍵生成を補助するための命令を持つ．例えば，aesenc は，AES の暗号化のラウンド関数 1 ラウンド分 (SubBytes，ShiftRows，MixColumns，AddRoundKey) を実行するための命令であり，aesenclast は AES の最終ラウンドのラウンド関数 (SubBytes, ShiftRows，AddRoundKey) を実行するための命令である．これらの命令の実行速度はレイテンシで評価できる．レイテンシは 1 つの命令を実行するのに必要なクロックサイクル数である．また，並列実行を考慮する場合，スループットも重要になる．スループットは，同じ命令を実行する際，待機にかかるクロックサイクル数である．図 1 に，aesenc のレイテンシが 4，スループットが 1 の場合の並列実行の様子を示す．レイテンシとスループットはプロセッサのアーキテクチャによって異なる．

2.2 Haraka-v2

Haraka-v2 [3] は 2 ラウンドの AES ラウンド関数と，8-word 置換から構成される，高速なハッシュ関数である．Haraka-v2 は入力が 256-bit である Haraka-v2-256 と，入力が 512-bit である Haraka-v2-512 があり，ともにステップ数が 5 で，256-bit の出力を持つ．AES ラウンド関数ではラウンド鍵として異なったラウンド定数を入力する．

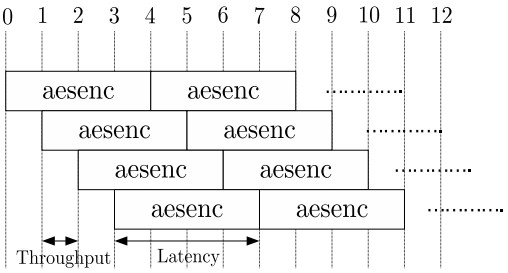

図 1 aesenc の並列実行

Haraka-v2-256 のステップ関数は 2 つの 2 ラウンドの AES ラウンド関数と 8-word 置換で構成されている．ここで Haraka-v2-256 の r ステップにおけるステップ関数の入力を $x_i^r \in \{0,1\}^{32}(i = 0, \cdots, 7)$ とする．haraka-v2-256 のステップ関数を図 2 に示す．

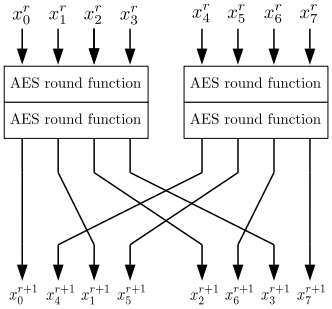

図 2 Haraka-v2-256 のステップ関数

Haraka-v2-512 のステップ関数は 4 つの 2 ラウンドの AES ラウンド関数と 16-word 置換で構成されている．ここで，Haraka-v2-512 の r ステップにおけるステップ関数の入力を $\hat{x}_j^r \in \{0,1\}^{32}, (j = 0, \cdots, 15)$ とする．Haraka-v2-512 のステップ関数を図 3 に示す．

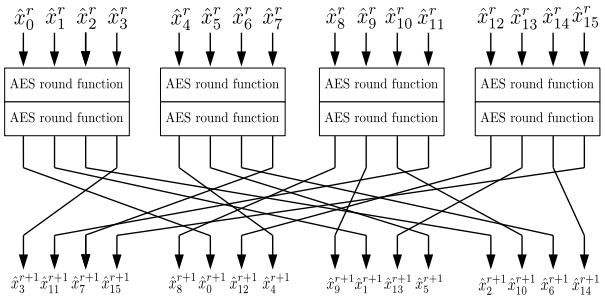

図 3 Haraka-v2-512 のステップ関数

2.3 Pholkos

Pholkos [2] は 2 ラウンドの AES ラウンド関数と，8-word 置換から構成されるラージ Tweakable ブロック暗号である．Pholkos は入出力が 256-bit の Pholkos-256 と，入出力が 512-bit の Pholkos-512 がある．いずれの場合も最終ステップでは置換を行わず，最後に用いる AES ラウンド関数は MixColumns を行わないものを用いる．また，鍵長や tweak のサイズによっても様々な種類がある．n を入出力長，k を鍵長，t を tweak 長としたとき，表 1 に各 Pholkos のパラメータを示す．Pholkos はデータ撹拌部であるステップ関数と Tweak と鍵を撹拌する Tweakey スケジュール関数から構成 [4] される．Pholkos-256 のステップ関数は 2 つの 2 ラウンドの AES ラウンド関数と 8-word 置換で構成されている．ここで Pholkos-256 の r ステップにおけるステップ関数の入力を $y_i^r \in \{0,1\}^{32}(i = 0, \cdots, 7)$ とする．Pholkos-256 のステップ関数を図 4 に示す．

表 1 Pholkos の種類と各パラメータ

Version	Sizes (bits)			Steps
	n	k	t	s
Pholkos-256-256	256	256	128	8
Pholkos-256-perm	256	-	128	12
Pholkos-512-256	512	256	128	10
Pholkos-512-512	512	512	128	10
Pholkos-512-perm	512	-	128	14

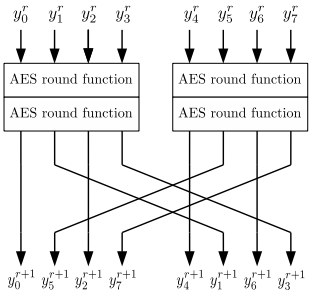

図 4 Pholkos-256 のステップ関数

Pholkos-512 のステップ関数は 4 つの 2 ラウンドの AES ラウンド関数と 16-word 置換で構成されている．ここで，Pholkos-512 の r ステップにおけるステップ関数の入力を $\hat{y}_j^r \in \{0,1\}^{32}, (j = 0, \cdots, 15)$ とする．Pholkos-512 のステップ関数を図 5 に示す．

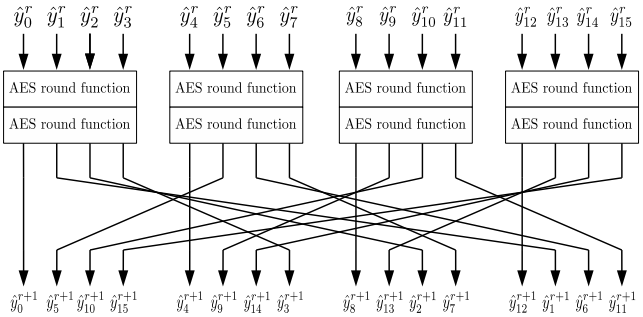

図 5 Pholkos-512 のステップ関数

2.4 Active S-box による差分/線形攻撃の安全性の評価

差分/線形攻撃はブロック暗号に対する最も基本的な攻撃法である．b-bit ブロック暗号 f についての差分/線形攻撃に対する安全性の評価を行う場合，以下の式で定義される差分確率 (DP_f) と線形確率 (LP_f) を導出し，それらの最大値である最大差分/線形確率を用いて評価する．なお，Δx と Δy は入力/出力差分，Γx と Γy は入力/出力マスクである．

$$DP_f(\Delta x, \Delta y) = \frac{\#\{x \in \{0,1\}^b | f(x) \oplus f(x \oplus \Delta x) = \Delta y\}}{2^b}$$

$$LP_f(\Gamma x, \Gamma y) = \left(2\frac{\#\{x \in \{0,1\}^b | x \bullet \Gamma x = f(x) \bullet \Gamma y\}}{2^b} - 1\right)^2$$

比較的ブロック長 b が小さい場合，最大差分/線形確率を求めることは容易であるが，現在提案されている多くのブロック暗号が持つブロック長 64/128-bit においては現実的な時間で最大差分/線形確率を導出することはできない．そこで，実際の評価の際は最大差分/線形確率の近似値として最大差分/線形特性確率 (DCP_{fmax} / LCP_{fmax}) が用いられる．これらは各ラウンドの差分/線形特性確率 (DP_f / LP_f) の積で定義される．

$$DCP_f = \prod_{R=1}^{r} DP_f(\Delta x_R, \Delta x_{R+1})$$

$$LCP_f = \prod_{R=1}^{r} LP_f(\Gamma x_R, \Gamma x_{R+1})$$

$$DCP_{fmax} = \max_{\substack{\Delta x_1 \neq 0 \\ \Delta x_2, \ldots, \Delta x_{r+1}}} DCP_f$$

$$LCP_{fmax} = \max_{\substack{\Gamma x_1 \neq 0 \\ \Gamma x_2, \ldots, \Gamma x_{r+1}}} LCP_f$$

一般的に差分/線形攻撃に対する安全性の評価を行う際，Active S-box による安全性の評価が行われる．S-box への入力差分/マスクが非 0 であるとき，その S-box を Active S-box と呼ぶ．差分/線形特性確率は系全体の Active S-box の最大差分/線形確率の積で抑えられる．遷移する可能性のあるすべての差分/線形マスクのパスを考慮し，Active S-box 数の下界を評価することで，差分特性確率の上界を評価することができる．一般的に，ブロック暗号に含まれる Active S-box の数を保証する方法には 2 種類ある．1 つは証明などで示された Active S-box 数の下界を用いる方法，もう 1 つは探索アルゴリズムにより，Active S-box 数の下界を評価する方法である．本稿では 2 つ目の探索アルゴリズムにより，Active S-box 数の下界を評価する方法を用いる．

この方法での Active S-box 数の下界の評価は，混合整数線形計画法 (Mixed Integer Linear Programming，MILP) を用いて効率的に行うことができる．MILP は，ある変数に対して線形式で与えられる制約式の下，線形式で与えられる目的関数を最適化 (最大化もしくは最小化) する変数の値を探索する．Mouha らが提案した手法 [5] では，まず暗号内部の各演算を線形式で表現し，制約式として与える．そして目的関数として Active S-box の合計数を与え，最小化することにより Active S-box の最小数を得る．本稿では MILP ソルバーとして Gurobi Optimizer [6] を用い，様々なクラスの置換を用いたラージブロック暗号に対して，Mouha らと同様の手法を用いて，安全性の評価を行う．

2.5 未解決問題と本論文の目的

2.2，2.3 で述べたように，Haraka-v2，Pholkos では，1 ステップ中に 2 ラウンドの AES ラウンド関数を用いる 2 ラウンド構成となっている．これは，2 ラウンドの AES ラウンド関数に差分/線形マスクが入力された場合，5 つの Active S-box を保証することができ，容易に安全性評価を行うことができるためである．

しかし，安全性や実行速度の観点から，2 ラウンドの AES ラウンド関数を用いることが最適であることは Pholkos と Haraka-v2 の提案論文では検討されていない．

本稿では 1 ラウンド構成の 256-bit ブロック暗号と 512-bit ブロック暗号の検討を行い，AES-NI を用いた SIMD 実装による速度評価と，MILP による Active S-box 評価の観点から 1 ラウンド構成のブロック暗号と 2 ラウンド構成のブロック暗号の比較を行う．また，実行速度については，CPU のアーキテクチャに依存するため，複数の異なるアーキテクチャを持つ CPU 上で実行速度を測定し，各構成の比較を行う．

3. 対象構成

本章では，本稿で検討する AES-NI と置換からなる 256-bit ブロック暗号と 512-bit ブロック暗号の構成を示す．ここで，8-word 置換を π_8，16-word 置換を π_{16}，1 ラウンドの AES ラウンド関数を R_{AES} とする．

3.1 2 ラウンド構成

2 ラウンド構成では Pholkos，Haraka-v2 と同様に 2 ラウンドの AES ラウンド関数を適用した後，置換を適用する．2 ラウンド構成における 256-bit ブロック暗号と 512-bit ブロック暗号のステップ関数をそれぞれ $S_{(2,256)}$，$S_{(2,512)}$ とする．2 ラウンド構成のステップ関数は以下の式で表される．

$$S_{(2,w)} = \pi_{w/32} \circ R_{AES} \circ R_{AES}, \quad w \in \{256, 512\}$$

3.2 1 ラウンド構成

1 ラウンド構成では，1 ラウンドの AES ラウンド関数を適用した後，置換を適用する．1 ラウンド構成における 256-bit ブロック暗号と 512-bit ブロック暗号のステップ関数をそれぞれ $S_{(1,256)}$，$S_{(1,512)}$ とする．1 ラウンド構成のステップ関数は以下の式で表される．

$$S_{(1,w)} = \pi_{w/32} \circ R_{AES}, \quad w \in \{256, 512\}$$

4. Active S-box 評価における最適な置換の検討

本章では，3 章で示した構成について Active S-box 評価における最適な置換の検討を行う．まず，Pholkos と Haraka-v2 で用いられている置換を実現する SIMD 命令について説明し，同様の SIMD 命令で実現できる置換のクラスを示す．次に，本稿で検討する置換のクラスを示し，各構成について Active S-box 評価において最適な置換の探索を行う．

4.1 Pholkos 及び Haraka-v2 で使用されている置換

Pholkos 及び Haraka-v2 の置換は，AVX2 の SIMD 命令における punpckldq (_mm_unpacklo_epi32)，punpckhdq (_mm_unpackhi_epi32)，vpblendd (_mm_blend_epi32) により実現されている．これらは全て，128-bit レジスタを操作する命令を用いている．Algorithm1，2，3 に各命令のアルゴリズム [7] を示す．これらの命令のレイテンシは本稿で扱う CPU のアーキテクチャ上では共通して 1 である．入力ブロック長が 512-bit である Pholkos-512 と Haraka-v2-512 の置換については，本節で示した 3 つの命令のうち 1 つを 2 回使用することで実現している．表 2 に Pholkos と Haraka-v2 で使用されている置換の SIMD 命令と，各置換のレイテンシを示す [8]．ここで，π_8^u，π_8^b，π_{16}^u，π_{16}^b は同様の SIMD 命令で実現可能な置換のクラスを示す．

Algorithm 1 punpckldq (_mm_unpacklo_epi32)

```
Input: dst[128], src1[128], src2[128]
Store the result in: dst[128]
  dst[31:0]←src1[31:0]
  dst[63:32]←src2[31:0]
  dst[95:64]←src1[63:32]
  dst[127:96]←src2[63:32]
```

Algorithm 2 punpckhdq (_mm_unpackhi_epi32)

```
Input: dst[128], src1[128], src2[128]
Store the result in: dst[128]
  dst[31:0]←src1[95:64]
  dst[63:32]←src2[95:64]
  dst[95:64]←src1[127:96]
  dst[127:96]←src2[127:96]
```

4.2 検討する置換のクラス

256-bit ブロック暗号の構成については，8-word 置換全ての候補である 8!($\approx 2^{15.30}$) 通りについて MILP による Active S-box 評価を行う．ここでは，AES の S-box の最大差分/線形特性確率が 2^{-6} であるため，より少ないステップ数で Active S-box の最少数が 43 以上 ($2^{-6\times 43} < 2^{-256}$) となる構成の探索を行う．

512-bit ブロック暗号の構成については，16-word 置換全ての候補を探索することは計算量的に困難なため，π_{16}^u の一部である全 7962624($\approx 2^{22.92}$) 通り中 120000($\approx 2^{16.87}$) 通りと，π_{16}^b のクラスの全ての置換 331776($\approx 2^{18.34}$) 通りについて MILP による Active S-box 評価を行う．256-bit ブロック暗号の場合と同様に，AES の S-box の最大差分/線形特性確率が 2^{-6} であるため，より少ないステップ数で Active S-box の最少数が 86 以上 ($2^{-6\times 86} < 2^{-512}$) となる構成の探索を行う．

Algorithm 3 vpblendd (_mm_blend_epi32)

```
Input: dst[128], a[128], b[128], imm8
Store the result in: dst[128]
  for j = 0 to 3 do
    i←j*32
    if imm8[j] then
      dst[i+31:i]←b[i+31:i]
    else
      dst[i+31:i]←a[i+31:i]
    end if
  end for
  dst[MAX:128]←0
```

表 2 Haraka-v2，Pholkos で使用している各置換の命令とレイテンシ

置換	使用	命令	レイテンシ
π_8^u	Haraka-v2-256	punpckldq, punpckhdq	1
π_8^b	Pholkos-256	vpblendd	1
π_{16}^u	Haraka-v2-512	punpckldq, punpckhdq	2
π_{16}^b	Pholkos-512	vpblendd	2

4.3 置換の探索結果

256-bit ブロック暗号については，2 ラウンド構成において 3.5 ステップ (3 ステップ+R_{AES}) で Active S-box の最少数が 43 以上の構成を 20736($\approx 2^{14.34}$) 通り発見した．1 ラウンド構成においては 6 ステップで条件を満たす構成を 20736 通り発見した．いずれの構成においても，条件を満たした置換の中に，表 2 で示した π_8^u, π_8^b のクラスに含まれる置換が存在した．

512-bit ブロック暗号については，π_{16}^b の置換を用いた場合は，2 ラウンド構成においては 4.5 ステップ (4 ステップ+R_{AES}) で条件を満たす構成を 5464($\approx 2^{12.16}$) 通り発見した．1 ラウンド構成においては 8 ステップで条件を満たす構成を 432($\approx 2^{8.75}$) 通り発見した．π_{16}^u の置換を用いた場合は，探索した全ての構成 120000($\approx 2^{16.87}$) 通りが 2 ラウンド構成において，4.5 ステップで条件を満たし，1 ラウンド構成においては，探索した全ての構成 120000($\approx 2^{16.87}$) 通りが，10 ステップで条件を満たした．

4.4 AES ラウンド関数の削減

探索の結果から，1 ラウンド構成と 2 ラウンド構成それぞれにおいて，各クラスの置換を用いた場合の AES ラウンド関数と置換の実行回数を表 3 に示す．表より，$n = 256$, $n = 512$ のいずれの場合でも，1 ラウンド構成にすることで，AES ラウンド関数 R_{AES} の実行回数を削減可能な置換が存在することを発見した．

5. 実装による速度評価

本章では，Pholkos-256, 512, Haraka-v2-256, 512 と 4.3 節で発見した置換を用いた 256-bit ブロック暗号と 512-bit ブロック暗号について，SIMD 実装を行い，実行速度を計測し，各構成の実行速度について考察する．まず，各構成における暗号化処理の理論上の速度についての考察を述べる．次に，速度計測を行う際の計測方法・環境について説明し，各構成における実行速度の計測結果を示す．最後に，各構成の実測値について比較検討を行う．

5.1 各構成の実行速度の理論値

本節では，4 章で探索した Active S-box 評価において最適な構成についての SIMD 実装による実行速度についての検討を行う．表 3 に 4 章で発見した Active S-box 評価において最適な構成の AES ラウンド関数の実行回数および，置換の実行回数を示す．以下に，表 3 を参考に，256-bit ブロック暗号と 512-bit ブロック暗号の 1 ラウンド構成と 2 ラウンド構成の実行速度についてそれぞれ考察する．なお，ここでは，比較的長い平文を暗号化した際に要する，暗号化処理の平均のクロックサイクル数についての考察を行う．

256-bit ブロック暗号，2 ラウンド構成の場合

表 3 より，$n = 256$ のとき，2 ラウンド構成では $S_{(2,256)}$ が 3 回と，R_{AES} が 1 回用いられている．すなわち，aesenc が 2 回と π_8^u，π_8^b の置換の実行が 3 ステップと更に 1 回の R_{AES} の実行が必要になる．実際には最後の R_{AES} には，最終ラウンド用の AES ラウンド関数の実行命令 aesenclast が用いられる．ここで，aesenc のレイテンシを L_{aesenc} とし，π_8^u，π_8^b の置換のレイテンシを L_{π_8} とする．また，$L_{\pi_8} = 1$ であり，aesenclast のレイテンシは aesenc のレイテンシと同じであるため，2 ラウンド構成の場合，暗号化処理でかかるサイクル数は，

$$3 \times (L_{\text{aesenc}} \times 2 + L_{\pi_8}) + L_{\text{aesenc}} \qquad (1)$$

と表せる．

256-bit ブロック暗号，1 ラウンド構成の場合

2 ラウンド構成の場合と同様に考えると，1 ラウンド構成の場合は，$S_{(1,256)}$ が 5 回と，R_{AES} が 1 回用いられているため，aesenc1 回と π_8^u，π_8^b の置換の実行が 5 ステップと 1 回の aesenclast の実行が必要になる．このとき暗号化処理全体でかかるサイクル数は，

$$5 \times (L_{\text{aesenc}} + L_{\pi_8}) + L_{\text{aesenc}} \qquad (2)$$

となる．

L_{aesenc} の取りうる値は各アーキテクチャによって異なり，表 4 のようになる [8]．*well アーキテクチャであれば，$L_{\text{aesenc}} = 7$ のため，2 ラウンド構成でかかるサイクル数は (1) より，52 になる．対して，1 ラウンド構成であれば，(2) より 47 になる．よって理論的には，*well アーキテクチャにおいて 1 ブロックの暗号化処理であれば 1 ラウンド構成の方が高速であると予測できる．

Skylake や Kaby Lake などのアーキテクチャでは，$L_{\text{aesenc}} = 4$ のため，2 ラウンド構成ではかかるサイクル数は 31，1 ラウンド構成では 29 となり，この場合においても，1 ラウンド構成の方が速いと予測できる．

512-bit ブロック暗号，2 ラウンド構成の場合

表 3 より，$n = 512$ のとき，2 ラウンド構成では $S_{(2,512)}$ が 4 回と，R_{AES} が 1 回用いられている．すなわち，aesenc が 2 回と π_{16}^u，π_{16}^b の置換の実行が 4 ステップと 1 回の aesenclast の実行が必要になる．また，π_{16} の置換の場合，用いる命令は 256-bit ずつしか操作できず，その出力同士がお互いの出力結果を必要とするため，置換 1 回にかかるサイクル数は 2 となる．よって π_{16} のレイテンシを $L_{\pi_{16}}$ とすると $L_{\pi_{16}} = 2$ である．256-bit ブロック暗号の場合と同様に考えると，暗号化処理でかかるサイクル数は

$$4 \times (L_{\text{aesenc}} \times 2 + L_{\pi_{16}} \times 2) + L_{\text{aesenc}} \qquad (3)$$

となる．

512-bit ブロック暗号，1 ラウンド構成の場合

1 ラウンド構成で，π_{16}^b の置換を用いる場合は $S_{(1,512)}$ が 7 回と R_{AES} が 1 回用いられており，aesenc が 1 回と π_{16}^b の置換の実行が 7 ステップと 1 回の aesenclast の実行が必要になる．このとき暗号化処理全体でかかるサイクル数は，

$$7 \times (L_{\text{aesenc}} + L_{\pi_{16}} \times 2) + L_{\text{aesenc}} \qquad (4)$$

となる．

$L_{\text{aesenc}} = 7$ である*well アーキテクチャで考えると，2 ラウンド構成の場合，(3) より全体でのサイクル数は 44, 1 ラウンド構成の場合では (4) より 42 となり，これらのアーキテクチャにおいて，理論的には 1 ラウンド構成に速いと予測できる．同様に考えると，$L_{\text{aesenc}} \leq 4$ となるアーキテクチャにおいては，2 ラウンド構成の方が高速であると予測できる．

表 3 各クラスの置換を用いた場合の AES ラウンド関数 R_{AES} と置換の実行回数

構成手法	ブロック長 (-bit)	置換	R_{AES} の実行回数	置換の実行回数
1 ラウンド構成	256	π_8^u	6	5
		π_8^b	6	5
	512	π_{16}^u	8	7
		π_{16}^b	10	9
2 ラウンド構成	256	π_8^u	7	3
		π_8^b	7	3
	512	π_{16}^u	9	4
		π_{16}^b	9	4

表 4 aesenc のレイテンシとスループット

プロセッサ	レイテンシ	スループット
Haswell	7	1
Broadwell	7	1
Skylake	4	1
Kaby Lake	4	1
Coffee Lake	4	1
Cannon Lake	4	0.5
Comet Lake	unknown	unknown
Ice Lake	3	0.5

5.2 測定手法/環境

実際にこれらのラージブロック暗号で平文を暗号化する際，平文は比較的大きいサイズで，様々な暗号利用モードで用いられる．本稿における速度計測では，100000-byte の平文を CBC モード，CTR モードでそれぞれ暗号化した場合についての 1-byte あたりの処理に必要なクロックサイクル数 (cbp) を計測する．また，aesenc のレイテンシは CPU のアーキテクチャ依存であることから，複数の異なった CPU のアーキテクチャ上で速度を計測する． ここで，実装に用いた置換を表 5 に示す．$\hat{\pi}_8^u$ と $\hat{\pi}_8^b$ は 4 章で発見した 256-bit ブロック暗号の置換であり，2 ラウンド構成で 3.5 ステップ，1 ラウンド構成で 6 ステップで Active S-box の最少数が 43 以上となる構成の置換の一つである．$\hat{\pi}_{16}^u$ は 4 章で発見した 512-bit ブロック暗号の置換であり，2 ラウンド構成で 4.5 ステップ，1 ラウンド構成で 10 ステップで Active S-box の最少数が 86 以上となる構成の置換の一つである．$\hat{\pi}_{16}^b$ も同様に 4 章で発見した 512–bit ブロック暗号の置換であり，2 ラウンド構成で 4.5 ステップ，1 ラウンド構成で 8 ステップで Active S-box の最少数が 86 以上となる構成の置換の一つである．また，本実験では暗号化処理のみを考慮しているため，鍵スケジューリングは考慮せず，ラウンド鍵として適当な定数を与える．

5.3 測定結果および比較評価

表 6, 7 に各測定結果を示す．左から順に，Intel の第 4 世代 (Haswell) から第 10 世代 (Comet Lake) のプロセッサで計測した結果を示している．結果より，以下の 3 つのことを明らかにした．

- CBC モードでは，256-bit ブロック暗号かつ*well アーキテクチャの場合において，1 ラウンド構成が 2 ラウンド構成よりも高速である．それ以外の場合では 2 ラウンド構成の方が高速または 1 ラウンド構成の場合と同等である．
- CTR モードでは，1 ラウンド構成と 2 ラウンド構成が同じ実行速度であった 256-bit ブロック暗号の Broadwell 以外の条件下で，2 ラウンド構成が 1 ラウンド構成よりも高速である．
- *well アーキテクチャ以外のプロセッサと，256-bit ブロック暗号の 1 ラウンド構成の CTR モード (Kaby Lake，第 8 世代の Coffee Lake) 以外において，置換を AVX2 の 128-bit のレジスタ操作を行う命令で実現する場合，punpckldq と punpckhdq を用いた場合の方が vpblendd を用いた場合よりも高速である．

1 ブロックの暗号化において，5.1 で示した理論値では 256-bit ブロック暗号の場合，1 ラウンド構成の方が高速で，512-bit ブロック暗号の場合*well アーキテクチャであれば，1 ラウンド構成の方が高速であったが，実測値では異なった結果になった．これは，置換の命令が，128-bit ずつの aesenc の出力結果を待つため，aesenc との並列実行ができなかったことが原因と考えられる．また，暗号利用モードの実装に用いた命令のサイクル数も含まれていることも原因の一部であると考えられる．

6. まとめ

本稿では，AES-NI の命令と置換から構成される 256-bit および 512-bit ブロック暗号について，1 ステップ中に AES ラウンド関数を 1 ラウンド用いた構成と，2 ラウンド用いた構成について，MILP による Active S-box 評価の観点と，SIMD 実装による実行速度の観点で比較を行った．結果として，1 ラウンド構成において，2 ラウンド構成よりも少ない AES ラウンド関数の実行回数で Active S-box 評価に

表 5 実装に用いた置換

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$\hat{\pi}_8^u(x)$	0	4	1	5	2	6	3	7	-	-	-	-	-	-	-	-
$\hat{\pi}_{16}^u(x)$	3	11	7	15	8	0	12	4	9	1	13	5	2	10	6	14
$\hat{\pi}_8^b(x)$	0	5	2	7	4	1	6	3	-	-	-	-	-	-	-	-
$\hat{\pi}_{16}^b(x)$	0	5	10	15	4	9	14	3	8	13	2	7	12	1	6	11

表 6 256-bit ブロック暗号の各アーキテクチャにおける暗号化処理の実行速度 (cpb)

モード	R (per step)	置換	Haswell	Broadwell	Skylake	Kaby Lake	Coffee Lake (8)	Coffee Lake (9)	Commet Lake
CBC	1	$\hat{\pi}_8^b$	**2.08**	**1.99**	1.71	1.47	0.97	1.60	1.39
	(total: 6)	$\hat{\pi}_8^u$	**2.23**	**2.14**	**1.58**	**1.17**	**0.87**	**1.38**	1.32
	2	$\hat{\pi}_8^b$	2.16	2.06	**1.67**	**1.39**	**0.93**	**1.55**	**1.31**
	(total: 7)	$\hat{\pi}_8^u$	2.25	2.15	**1.58**	**1.17**	**0.87**	**1.38**	**1.31**
CTR	1	$\hat{\pi}_8^b$	0.66	0.59	0.53	0.41	0.33	0.49	0.43
	(total: 6)	$\hat{\pi}_8^u$	0.81	**0.76**	0.53	0.42	0.34	0.49	0.38
	2	$\hat{\pi}_8^b$	**0.64**	**0.58**	**0.48**	**0.38**	**0.31**	**0.44**	**0.39**
	(total: 7)	$\hat{\pi}_8^u$	**0.75**	**0.76**	**0.47**	**0.37**	**0.31**	**0.44**	**0.37**

表 7 512-bit ブロック暗号の各アーキテクチャにおける暗号化処理の実行速度 (cpb)

モード	R (per step)	置換	Haswell	Broadwell	Skylake	Kaby Lake	Coffee Lake (8)	Coffee Lake (9)	Commet Lake
CBC	1 (total: 8)	$\hat{\pi}_{16}^b$	1.64	1.59	1.39	0.96	0.66	1.10	1.07
	1 (total:10)	$\hat{\pi}_{16}^u$	1.99	1.93	1.06	0.95	0.63	1.10	1.06
	2	$\hat{\pi}_{16}^b$	**1.52**	**1.47**	**1.12**	**0.84**	**0.48**	**1.01**	**0.93**
	(total: 9)	$\hat{\pi}_{16}^u$	**1.58**	**1.53**	**1.04**	**0.76**	**0.62**	**0.89**	**0.86**
CTR	1 (total: 8)	$\hat{\pi}_{16}^b$	0.99	0.93	0.77	0.59	0.45	0.69	0.62
	1 (total: 10)	$\hat{\pi}_{16}^u$	1.35	1.25	0.71	0.55	0.43	0.64	0.59
	2	$\hat{\pi}_{16}^b$	**0.87**	**0.81**	**0.64**	**0.50**	**0.38**	**0.61**	**0.54**
	(total: 9)	$\hat{\pi}_{16}^u$	**0.92**	**0.87**	**0.56**	**0.44**	**0.34**	**0.52**	**0.46**

よる差分/線形攻撃に対して安全な構成を発見した．また，実装評価の結果から，1 ラウンド構成と比べ，2 ラウンド構成が，殆どのアーキテクチャにおいて実行速度が速いことを発見した．さらに，置換を実現する SIMD 命令として，punpckldq と punpckhdq を用いた場合の方が vpblendd を用いた場合よりも*well アーキテクチャ以降であれば高速であることを発見した．

謝辞

本研究は科研費 19H02141 と中島記念国際交流財団 研究助成金の助成を受けたものです．

参考文献

[1] Shay Gueron. Intel advanced encryption standard (aes) new instructions set, 2010.

[2] Stefan Kölbl, Martin M. Lauridsen, Florian Mendel, and Christian Rechberger. Haraka v2 - efficient short-input hashing for post-quantum applications. *IACR Trans. Symmetric Cryptol.*, Vol. 2016, No. 2, pp. 1–29, 2016.

[3] Jannis Bossert, Eik List, Stefan Lucks, and Sebastian Schmitz. Pholkos - efficient large-state tweakable block ciphers from the AES round function. *IACR Cryptol. ePrint Arch.*, Vol. 2020, p. 275, 2020.

[4] Jérémy Jean, Ivica Nikolic, and Thomas Peyrin. Tweaks and keys for block ciphers: The TWEAKEY framework. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014, Proceedings, Part II*, Vol. 8874 of *Lecture Notes in Computer Science*, pp. 274–288. Springer, 2014.

[5] Nicky Mouha, Qingju Wang, Dawu Gu, and Bart Preneel. Differential and linear cryptanalysis using mixed-integer linear programming. In Chuankun Wu, Moti Yung, and Dongdai Lin, editors, *Information Security and Cryptology - 7th International Conference, Inscrypt 2011, Beijing, China, November 30 - December 3, 2011. Revised Selected Papers*, Vol. 7537 of *Lecture Notes in Computer Science*, pp. 57–76. Springer, 2011.

[6] Gurobi Optimization Inc. Gurobi optimizer 6.5. Official webpage, `http://www.gurobi.com/`, 2015.

[7] Intel Corporation. Intel intrinsics guide. Official webpage, `https://software.intel.com/sites/landingpage/IntrinsicsGuide/`.

[8] Real-Time and Embedded Sys Lab. uops.info. Official webpage, `https://www.uops.info/`.