

AES-NIを用いた Feistel 型ラージブロック置換

中橋 元輝^{1,a)} 芝 廉太郎¹ 五十部 孝典^{1,2,3}

概要：256 bit や 512 bit のブロック長を持つラージブロック置換は、ハッシュ関数や置換ベースのスポンジ関数での利用の観点で注目されている。既存のラージブロック置換としては、Feistel 構造で AES-NI 命令のみで実装可能な Simpira v2 や、SPN 構造で AES-NI 命令と SIMD 命令によるワード置換命令を用いて実装可能な Haraka v2 や Pholkos がある。本研究では、より高速な 256-bit/512-bit ラージブロック置換の設計を目的として、AES-NI 命令のみで実装可能でかつ、より多くの AES-NI 命令が並列実行可能な新しい Feistel 型のラージブロック置換を提案する。具体的には、対象とする全ての構成に対して混合整数線形計画法を用いた差分/線形攻撃の安全性を評価し、より少ない AES-NI 命令数で実装可能な構成を明らかにする。これらを複数の異なるアーキテクチャの CPU 上で速度評価を行い、全てのアーキテクチャにおいて 1 ブロックの暗号化が既存のラージブロック置換より高速であることを示す。特に最新のアーキテクチャである Ice Lake では、1 ブロックの暗号化は 30%程度高速であり、並列実行においても 10%以上の高速化を達成した。

キーワード：AES-NI, Feistel 構造, ラージブロック置換

Feistel-type Permutations with Large-state based on AES-NI

MOTOKI NAKAHASHI^{1,a)} RENTARO SHIBA¹ TAKANORI ISOBE^{1,2,3}

Abstract: Large-state permutations with 256-bit or 512-bit block sizes receive much attention from the viewpoint of their use in hash functions and permutation-based authenticated encryption. While existing large-state permutations like Haraka-v2 and Pholkos are designed to be implemented by instructions of AES-NI and some SIMD instructions for their word-shuffle layer, Simpira v2 which has the Feistel structure as the underlying construction are designed to be implemented by only instructions of AES-NI. In this paper, we propose new constructions of large-state permutations that can be implemented by only instructions of AES-NI like Simpira v2. Specifically, we evaluate the security of all the targeted constructions against differential and linear attacks using a mixed-integer linear programming solver. Based on the result of that evaluation, we identify constructions that can satisfy sufficient security against linear and differential attacks using the execution of the same or fewer number of AES-NI instructions than Simpira v2. Finally, we implement such constructions and evaluate their performance on some CPUs of Intel. As a result, we show that the encryption with 256-bit and 512-bit permutations we identified are faster than that with the some of existing large-state permutations. In particular, the encryption speed of one 512-bit block is 20 ~ 50% faster than the existing large-state permutation.

Keywords: AES-NI, Feistel structure, large-state permutation

1. はじめに

256 bit や 512 bit のブロック長を持つラージブロック置換は、ハッシュ関数や置換ベースの認証暗号での利用の観点で注目されている。Gueron らは 2016 年に、Feistel 型

¹ 兵庫県立大学, University of Hyogo, Japan.

² 国立研究開発法人情報通信研究機構, National Institute of Information and Communications Technology, Japan.

³ 国立研究開発法人科学技術振興機構, PRESTO, Japan Science and Technology Agency, Japan.

a) ad21p005@ai.u-hyogo.ac.jp

ラージブロック置換 *Simpira v2* を提案した [1]. *Simpira v2* は高速なソフトウェア実装と実装の簡易化の観点から構成に AES-NI と呼ばれる AES ラウンド関数とラウンド鍵生成をサポートする SIMD 命令のみを実装しており、128 の倍数のブロック長をサポートしている。また、Kölbl らは、2016 年に AES-NI を用いて AES ラウンド関数を並列実行し、その出力のワード置換を適用する処理を数回繰り返す構造を持つ SPN 型ハッシュ関数 *Haraka v2* を提案した [3]. Bossert らは 2020 年に、*Haraka v2* を参考に SPN 型ブロック暗号 *Pholkos* を提案した [4]. *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。

Haraka v2 と *Pholkos* が、その実装に AES-NI 命令と他のワード置換を行う SIMD 命令を必要とするのに対し、*Simpira v2* は、AES-NI 命令のみで実装できるように設計されている。AES-NI 命令のみを用いて構成できる置換の利点としては、ECB モードや CTR モード等で暗号化を並列処理で行う場合、AES-NI 命令に加えて他の命令セットを用いた場合より多くの命令をパイプライン実行できるという点が挙げられる。また、CPU の世代が新しくなる毎に AES-NI 命令のレイテンシが小さくなっている為、AES-NI 命令と他の SIMD 命令を同時に利用する場合は、AES-NI 命令のみを用いる場合と比較してオーバーヘッドが大きくなる。よって、AES-NI 命令のみを用いた置換の構成を考えることは、今後も進化していくと考えられる CPU 上でのラージブロック置換の実装を検討する上で重要である。

本稿では、*Simpira v2* のように AES-NI 命令のみで実装可能かつ、高速な暗号化処理が可能なラージブロック置換の構成について検討する。具体的には、*Simpira v2* より多くの AES-NI 命令を並列実行でき、より高速な暗号化処理が行えるような新しい構成を提案する。まず、本研究で対象とする全ての構成に対して、混合整数線型計画法ソルバーを用いた差分/線形攻撃に対する安全性評価を行う。その後、探索の結果得られた差分/線形攻撃に対して安全な構成のうち、既存構成である *Simpira v2* よりも少ない、または同じ数の AES-NI 命令で暗号化処理を実装可能な構成を実装し、異なるアーキテクチャを持つ複数の CPU 上での暗号化処理の速度を計測する。最後に、本稿で提案する新たな構成と 3 つの既存構成: *Simpira v2*, *Haraka v2*, *Pholkos* との比較検討を行う。結果として、次のことを明らかにした。

1 ブロック暗号化における実行速度

- 提案する構成は、256-bit 置換の Ice Lake 環境を除く 1 ブロックを暗号化する実行速度において最も高速であり、既存のラージブロック置換より 2%以上高速である。
- 提案する構成は、256-bit 置換の Ice Lake 環境での 1

ブロックを暗号化する実行速度において最も高速であり、既存のラージブロック置換より 27%以上高速である。

- 提案する構成は、512-bit 置換の 1 ブロックを暗号化する実行速度において最も高速であり、既存のラージブロック暗号より 17%以上高速である。

並列暗号化における実行速度

- *Haraka v2* と *Pholkos* は、256-bit 置換の Ice Lake 環境を除く並列実行速度において最も高速であり、提案する 256-bit 置換より 6%以上高速である。
- 提案する構成は、256-bit 置換の Ice Lake 環境での並列実行速度において最も高速であり、既存のラージブロック置換より 13%以上高速である。
- 提案する構成は、512-bit 置換の並列実行速度において最も高速である。Ice Lake 以外の環境では、*Simpira-4* と同じ速度であったが、Ice Lake 環境においては既存のラージブロック置換より 10%以上高速である。

提案する構成は、全てのアーキテクチャにおいて 1 ブロックの暗号化が既存のラージブロック置換より高速であり、最新のアーキテクチャである Ice Lake では、1 ブロックの暗号化は 30%程度高速であり、並列実行においても 10%以上の高速化を達成した。特に 512-bit 置換における 1 ブロックの暗号化速度は、既存のラージブロック置換と比較し、17～46% 高速である。

本稿の構成を以下に示す。2 章で基礎事項として AES-NI と既存構成である *Simpira v2* の概要について説明し、Active S-box により差分/線形攻撃の安全性評価について説明する。3 章では、本稿で提案する構成の概要について説明した後、提案構成の利点について説明する。4 章では、提案構成と *Simpira v2* 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 *Simpira v2* 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。

2.1 AES-NI

AES-NI (AES New Instructions set) [2] は、Intel や AMD の CPU に搭載される AES の処理を高速に行うための命令セットであり、AES ラウンド関数とその逆関数を実行する命令、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,

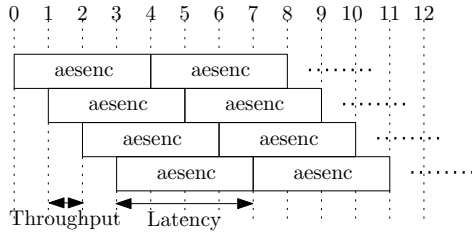


図 1 aesenc の並列実行

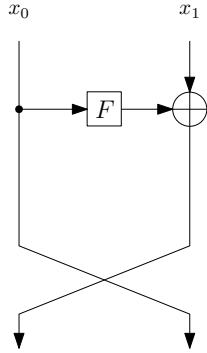


図 2 Simpira-2 の 1 ラウンド

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。各命令の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 Simpira v2

Simpira v2 [1] は、Feistel 構造を持つラージブロック置換であり、そのブロック長は 128 bit のサブブロックの倍数の bit 長に対応している。本稿ではサブブロック数が 2 (ブロック長 256 bit) の場合と 4 (ブロック長 512 bit) の場合に注目する。サブブロック数が 2 の場合と 4 の場合の Simpira v2 の 1 ラウンド処理をそれぞれ図 2 と図 3 に示す。本稿では、サブブロック数が 2 の Simpira v2 を Simpira-2、サブブロック数が 4 の Simpira v2 を Simpira-4 とする。

F は、2 回の AES ラウンド関数の適用から構成され、1 回目のラウンド関数の適用の際には鍵としてラウンド定数を用いる。2 回目のラウンド関数の適用は、図 2、3 における XOR の処理と対応する。これにより aesenc のみで実装可能になっている。

2.3 Active S-box による差分/線形攻撃の安全性の評価

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

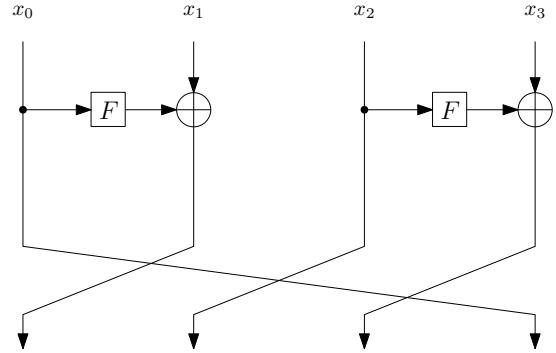


図 3 Simpira-4 の 1 ラウンド

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 つ目は理論的な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] を使い、提案するラージブロック置換に対して、Mouha らと同様の手法を用いて、安全性の評価を行う。

3. 対象構成

本章では、本稿で検討する Simpira v2 のような AES-NI 命令のみで実装可能な 256-bit、512-bit 置換の構成について示す。その後、本稿で対象とする構成の表記方法について明記する。

3.1 256-bit 置換

本稿で対象とする 256-bit 置換の 1 ラウンドの処理を図 4 に示す。Simpira-2 と異なり、分岐処理によって生成された 2 つの x_0 の値のいずれにも aesenc、または aesenclast で構成される関数 F_1 、 F_2 が適用されるような構成を考える。この構成は、通常の Feistel 構造と異なり逆関数が必要となるが、1 ラウンドにおいて各関数の入力には互いに依

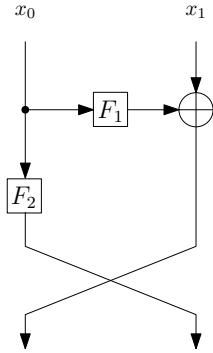


図 4 対象とする 256-bit 置換

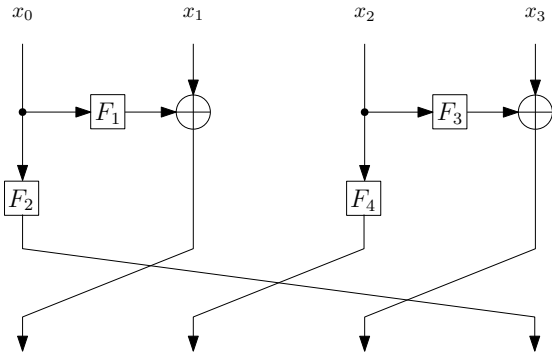


図 5 対象とする 512-bit 置換

存性が無く、これらの関数はいずれも AES-NI 命令のみで実装できる。その為、Simpira-2 よりもパイプラインを有効活用することが可能である。

3.2 512-bit 置換

本稿で対象とする 512-bit 置換の 1 ラウンドの処理を図 5 に示す。Simpira-4 を元にして、分岐処理によって生成された 2 つの x_0 の値と 2 つの x_2 の値全てに対して、aesenc、または aesencast から構成される関数 F_1, F_2, F_3, F_4 が適用されるような構成を考える。この構成も 3.1 で示した 256-bit 置換と同様に逆関数を必要とするが、1 ラウンドにおいて各関数の入力には互いに依存性が無く、これらの関数はいずれも AES-NI 命令のみで実装できる。その為、Simpira-4 よりもパイプラインを有効活用することが可能である。

3.3 本稿の構成の利点

本稿の構成は、以下 2 つの様な利点がある。

1. 並列実行可能なラウンド関数の追加 Simpira v2 の構成に加えて、並列実行可能なラウンド関数を追加した。ラウンド関数を追加することにより、1 ラウンドで Simpira v2 より多くの AES-NI の命令を実行でき、それらの命令の並列処理が可能になる。これにより、1 ラウンドで Simpira v2 より多くの AES-NI 命令を実行でき、それらの命令の並列処理が可能になる。よって本稿で対象とする 256-bit 置換と 512-bit 置換のク

ラス内に Simpira v2 よりも高速に実行可能な構成が存在する可能性がある。

2. AES-NI 命令のみを用いた構成 AES-NI 命令に加えて他の命令セットを用いた場合より多くの命令をパイプライン実行できる。また、CPU の世代が新しくなる毎に AES-NI 命令のレイテンシが小さくなっているため、AES-NI 命と他の SIMD 命令を同時に利用する場合は、AES-NI 命令のみを用いる場合と比較してオーバーヘッドが大きくなる。よって、AES-NI 命令のみを用いた置換の構成を考えることは、今後も進化していくと考えられる CPU 上でのラージブロック置換の実装を検討する上で重要である。

本稿では、256-bit 置換と 512-bit 置換の各構成を以下のように表す。

$$(F_1, F_2) = (N_1 - I, N_2 - I)$$

$$(F_1, F_2, F_3, F_4) = (N_1 - I, N_2 - I, N_3 - I, N_4 - I)$$

ここで、 N_i は、 F_i $i \in \{1, 2, 3, 4\}$ 内部の AES-NI 命令の数であり、 I は aesencast の数と場所を表す変数である。 $I = 0$ のとき、aesencast は 0 個、 $I = 1$ のとき、aesencast は先に実行される側に 1 つ、 $I = 2$ のとき、aesencast は後に実行される側に 1 つ、 $I = 3$ のとき、aesencast が 2 つであることを表す。

4. Active S-box 評価における最適な構成の検討

本章では、3 章で示した構成について MILP を用いた Active S-box 評価により、差分/線形攻撃に対して安全な置換の探索を行う。まず、対象とする置換の探索範囲について示し、Active S-box 評価において最適な構成の探索を行う。その後で、実際に実行速度の計測に用いる置換について述べる。

4.1 対象とする置換の探索範囲

256-bit 置換の探索範囲

図 4 に示される置換において、ラウンド関数 F_1, F_2 それぞれに aesenc、aesencast を 1 つ、または 2 つ適用した全ての置換 36 ($\approx 2^{5.17}$) 通りに対して 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} であることから、256-bit ブロックの置換において、Active S-box の最小数が 43 個 ($2^{-6 \times 43} < 2^{-256}$) となる置換の探索を行う。Simpira-2 で MILP による Active S-box 評価を行ったところ、8 ラウンドで 43 以上の Active S-box を達成した。Simpira-2 は、1 ラウンドで 2 つの AES-NI 命令を用いており、8 ラウンドでは 16 (2×8) 個の AES-NI 命令を用いている。暗号化の実行速度での優

位性を確保するため、暗号化に必要な AES-NI 命令の個数が 16 個以下で、Active S-box 数の下界が 43 になるような構成の探索を行う。また、aesenc のみで実装可能な置換については、10 ラウンド以内で Active S-box を達成出来るパターンを探索する。

512-bit 置換の探索範囲 図 5 に示される置換において、ラウンド関数 F_1, F_2, F_3, F_4 それぞれに aesenc, aes-enclast を 1 つ、または 2 つ適用した全ての置換、全ての 128-bit シャッフル 29808 ($\approx 2^{14.86}$) 通りに対して MILP による Active S-box 評価を行う。256-bit の場合と同様に考えると、AES の S-box の最大差分/線形特性確立が 2^{-6} であることから、512-bit ブロックの置換において、差分/線形攻撃にたいして十分な安全を確保できる Active S-box の最小数は 86 個 ($2^{-6 \times 86} < 2^{-512}$) となる。したがって、512-bit 置換の置換においては Active S-box 数の最小数が 86 個以上になるような構成を探索する。Simpira-4 で、MILP による Active S-box 評価を行ったところ、10 ラウンドで 86 以上の Active S-box を達成した。Simpira-4 は、1 ラウンドで 4 つの AES-NI 命令を用いており、10 ラウンドでは 40 (4×10) 個の AES-NI 命令を用いている。暗号化の実行速度での優位性を確保するため、暗号化に必要な AES-NI 命令の個数が 40 個以下で、Active S-box 数の最小数が 86 個になるような構成の探索を行う。aesenc のみで実装可能な置換については、7 ラウンド以内で Active S-box を達成し、AES-NI 命令の実行回数が 50 以下となるパターンを探索する。複数の 128-bit シャッフルにおいて、ラウンド関数が同じであり、同じラウンド数で条件を達成出来た場合、暗号化の実行速度は変わらないのでそれを 1 通りとし、達成ラウンドにおいて最も Active S-box が多かったパターンの 1 つを採用する。

4.2 探索結果

256-bit 置換の場合

256-bit ブロック暗号については、実装に必要な AES-NI 命令の個数が 16 個以下で、Active S-box の最小数が 43 個以上の構成を 3 通り発見した。また、aesenc のみで実装可能な置換については、10 ラウンド以内で Active S-box を達成出来るパターンを 3 通り発見した。発見した置換を図 6 に示す。また、発見した置換の内容、ラウンド数、AES-NI 命令の実行回数を表 1 に示す。

512-bit 置換の場合 512-bit ブロック暗号については、実装に必要な AES-NI 命令の個数が 40 個以下で、Active S-box の最小数が 86 個以上の構成を 1 通り発見した。また、aesenc のみを用いて実装可能であり、7 ラウンド以内で Active S-box を達成し、AES-NI 命令の実装

表 1 発見した 256-bit 置換のラウンド数と AES-NI の実装数

置換	ラウンド数	AES-NI 命令数
(1-0, 2-0)	6	18
(2-0, 1-0)	6	18
(2-0, 2-0)	6	24
(1-0, 1-1)	8	16
(2-0, 1-1)	5	15
(2-0, 2-2)	4	16

表 2 発見した 512-bit 置換のラウンド数と AES-NI の実装数

置換	ラウンド数	AES-NI 命令数
(1-0, 2-0, 2-0, 1-0)	7	49
(2-0, 1-0, 1-0, 2-0)	7	49
(2-0, 1-0, 2-0, 1-0)	7	49
(2-0, 2-0, 2-0, 2-0)	6	48
(2-0, 2-2, 2-0, 2-2)	5	40

数が 50 以下のパターンを 4 通り発見した。発見した置換を図 7 に示す。また、発見した置換の構成内容、ラウンド数、AES-NI 命令の実行回数を表 2 に示す。

5. 実装による速度評価

本章では、Simpira v2, Haraka v2, Pholkos と 4 章で発見した置換について AES-NI と SIMD 命令を用いた実装を行い、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、Simpira v2 と測定を行う置換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・環境についての考察を述べ、各構成に置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 各構成における暗号化処理の理論値

本節では、4 章で示した速度計測を行う提案置換と Simpira v2 について実行速度の理論値についての検討を行う。まず、1 ラウンドでかかるサイクル数の理論値を算出する。その後、1 ラウンドでかかるサイクル数にラウンド数を掛けることにより暗号化処理の理論値を算出する。本研究では、理論値の計算過程においては、レイテンシのみを考慮している。以下に、提案構成と Simpira v2 の 1 ラウンドでかかるサイクル数の算出方法を示す。

Simpira v2 の場合

Simpira v2 は、ラウンド関数として 2 ラウンドの AES ラウンド関数から構成されている。よって、aesenc(aesenclast) のレイテンシを L_{aesenc} とする。1 ラウンドでかかるサイクル数を

$$2 \times L_{\text{aesenc}}$$

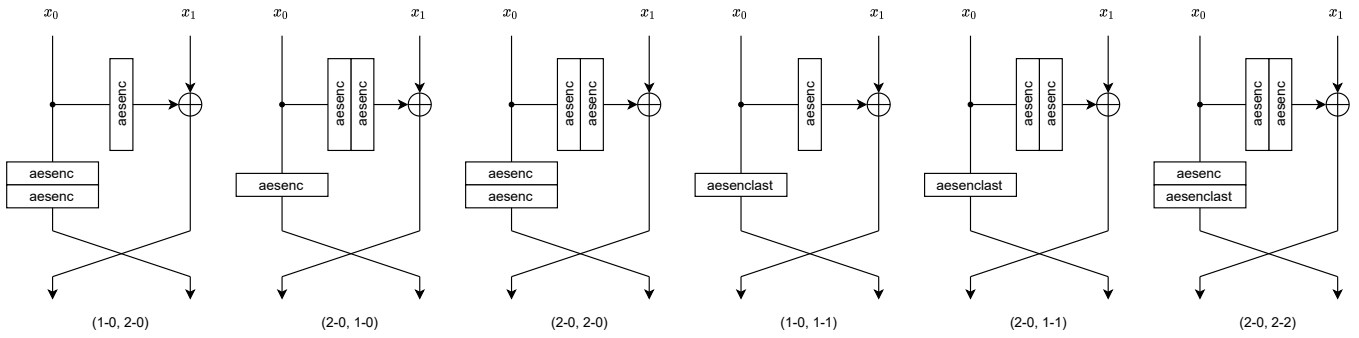


図 6 発見した 256-bit 置換

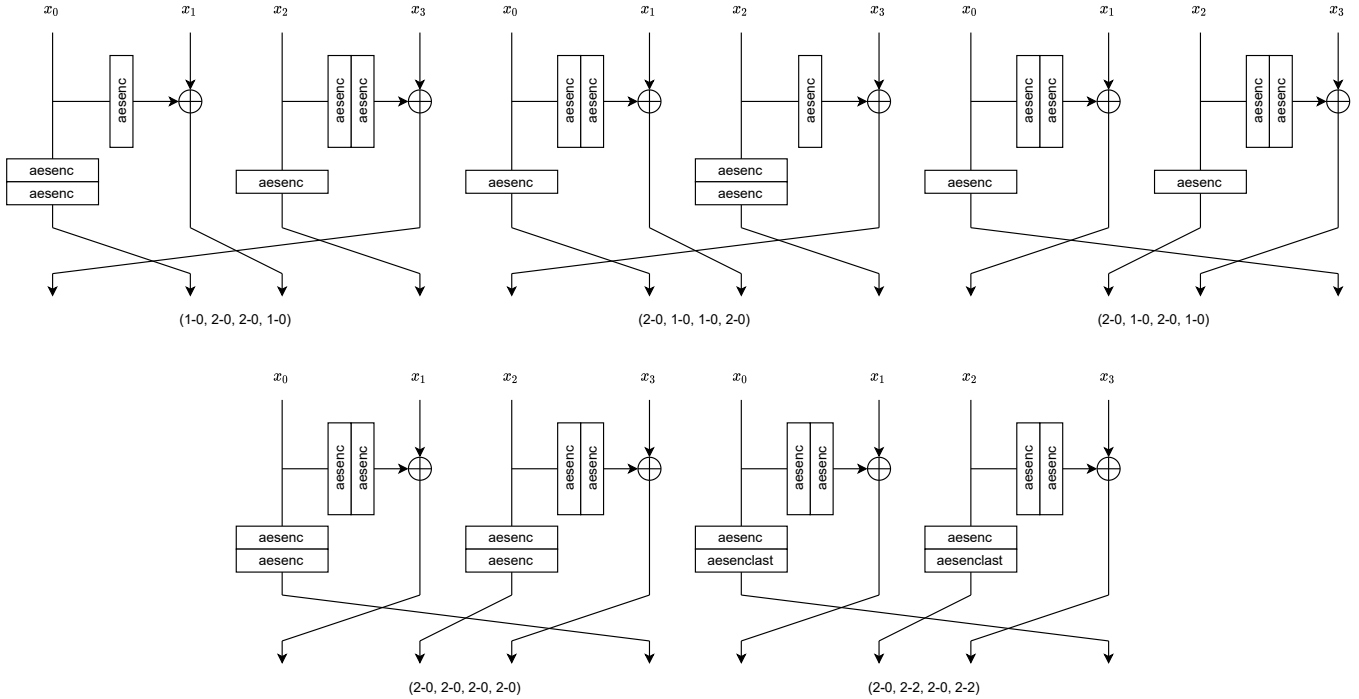


図 7 発見した 512-bit 置換

とする。

提案構成の場合

提案置換では、Simpira v2 に加え、並列化出来るラウンド関数を実装している。1 ラウンド中のラウンド関数は全て並列化出来るので、1 ラウンドのサイクル数は最も多く AES-NI 命令を用いているラウンド関数のラウンド数と近似できる。よって、ラウンド関数で用いている最も多い AES-NI 命令の個数を m とする。1 ラウンドでかかるサイクル数を、

$$m \times L_{\text{aesenc}}$$

とする。

L_{aesenc} の取りうる値は各アーキテクチャによって異なる。また、SIMD レジスタ数、スループットの値も各アーキテクチャごとに異なる。速度評価に用いる各アーキテクチャの SIMD レジスタ数、aesenc または aesencclast のレイテンシ、スループットを表 3 に示す。 L_{aesenc} の値を 4 または 3

表 3 SIMD レジスタ数と aesenc (aesencclast) のレイテンシとスループット

プロセッサ	SIMD 数	レイテンシ	スループット
Skylake	16	4	1
Kaby Lake	16	4	1
Coffee Lake	16	4	1
Ice Lake	32	3	0.5

とし、理論値を算出する。算出した理論値のサイクル数を 256-bit 置換の場合は表 4 に、512-bit 置換の場合は表 5 に示す。表 4, 5 より、提案する全ての置換が Simpira v2 より高速であると予測できる。

5.2 測定手法/環境

本稿では、1 ブロックあたりの暗号化処理の速度と、並列処理により数ブロックを同時に暗号化する際の速度についての計測を行う。暗号化の並列実行における速度計

表 4 256-bit 置換のサイクル数の理論値

置換	ラウンド数	$L_{\text{aesenc}} = 3$	$L_{\text{aesenc}} = 4$
Simpira-2	8	48	64
(1-0,2-0)	6	36	48
(2-0,1-0)	6	36	48
(2-0,2-0)	6	36	48
(1-0,1-1)	8	24	32
(2-0,1-1)	5	30	40
(2-0, 2-2)	4	24	32

表 5 512-bit 置換のサイクル数の理論値

置換	ラウンド数	$L_{\text{aesenc}} = 3$	$L_{\text{aesenc}} = 4$
Simpira-4	10	60	80
(1-0, 2-0, 2-0, 1-0)	7	42	56
(2-0, 1-0, 1-0, 2-0)	7	42	56
(2-0, 1-0, 2-0, 1-0)	7	42	56
(2-0, 2-0, 2-0, 2-0)	6	36	48
(2-0, 2-2, 2-0, 2-2)	5	30	40

測については、Ice Lake 以外のプロセッサの場合は、ブロック長に対応する入力を 4 ブロック並列実行で暗号化した場合についての速度を計測する。Ice Lake の場合は、SIMD レジスタが他のプロセッサに比べて 2 倍の 32 個あるので、8 ブロックを並列実行で暗号化した場合を計測する。計測する速度の単位を 1-byte あたりの処理に必要なクロックサイクル数 (cpb) とし、これらのパターンをアーキテクチャの異なる複数の CPU で計測する。比較対象とする Haraka v2 と Pholkos のラウンド数は、差分攻撃と線形攻撃に対して十分な安全性を満たすように設定する。Haraka v2 と Pholkos における 1 ラウンドは、2 回の AES-NI の命令の並列実行とその出力に対するワード置換の適用から構成される。MILP による Active S-box 数の評価から、256-bit の Haraka v2 と Pholkos は 3 ステップと更に一回の aesenc1ast の並列実行 (3.5 ラウンド) で、512-bit の Haraka v2 と Pholkos は 4 ステップと更に一回の aesenc1ast の並列実行 (4.5 ラウンド) で、差分/線形攻撃に対して安全と言える。よって、256-bit 置換の実測値計測には、256-bit ブロックで 3.5 ラウンドの Haraka v2 と Pholkos (AES-NI 命令数 14, ワード置換命令数 3) を用い、512-bit 置換の実測値計測には、512-bit ブロックで 4.5 ラウンドの Haraka v2 と Pholkos (AES-NI 命令数 34, ワード置換命令数 8) を用いる。

5.3 測定結果および比較評価

表 6, 7, 8, 9 に各測定結果を示す。表 6, 7 は、256-bit 置換での各アーキテクチャで計測した速度、表 8, 9 は、512-bit 置換での各アーキテクチャで計測した結果を示している。表 6, 7, 8, 9 より、以下のことが言える。

1 ブロック暗号化における実行速度

表 6 256-bit ブロック暗号の各アーキテクチャにおける 1 ブロック暗号化の実行速度

置換	Skylake	Kaby Lake	Coffee Lake	Ice Lake
Simpira-2	2.13	2.01	2.00	1.51
(1-0, 2-0)	1.18	1.13	1.13	0.85
(2-0, 1-0)	1.61	1.54	1.54	1.16
(2-0, 2-0)	1.64	1.54	1.53	1.17
(1-0, 1-1)	1.10	1.14	1.14	0.76
(2-0, 1-1)	1.24	1.19	1.18	0.88
(2-0, 2-2)	1.08	1.03	1.03	0.77
Haraka v2	1.11	1.07	1.07	1.05
Pholkos	1.21	1.16	1.16	1.06

表 7 256-bit ブロック暗号の各アーキテクチャにおける並列暗号化処理の実行速度

置換	Skylake	Kaby Lake	Coffee Lake	Ice Lake
Simpira-2	0.53	0.50	0.51	0.29
(1-0, 2-0)	0.60	0.57	0.56	0.30
(2-0, 1-0)	0.59	0.57	0.56	0.31
(2-0, 2-0)	0.79	0.75	0.75	0.40
(1-0, 1-1)	0.52	0.50	0.51	0.29
(2-0, 1-1)	0.50	0.47	0.47	0.25
(2-0, 2-2)	0.52	0.50	0.50	0.26
Haraka v2	0.46	0.44	0.44	0.31
Pholkos	0.46	0.44	0.44	0.32

- (2-0, 2-2) 構成は、256-bit 置換の Ice Lake 環境を除く 1 ブロックを暗号化する実行速度において最も高速であり、既存のラージブロック置換より 2%以上高速である。
- (1-0, 1-1) 構成は、256-bit 置換の Ice Lake 環境での 1 ブロックを暗号化する実行速度において最も高速であり、既存のラージブロック置換より 27%以上高速である。
- (2-0, 2-2, 2-0, 2-2) 構成は、512-bit 置換の 1 ブロックを暗号化する実行速度において最も高速であり、既存のラージブロック暗号より 17%以上高速である。

並列暗号化における実行速度

- Haraka v2 と Pholkos は、256-bit 置換の Ice Lake 環境を除く並列実行速度において最も高速であり、提案する 256-bit 置換より 6%以上高速である。
- (2-0, 1-1) 構成は、256-bit 置換の Ice Lake 環境での並列実行速度において最も高速であり、既存のラージブロック置換より 13%以上高速である。
- (2-0, 2-2, 2-0, 2-2) 構成は、512-bit 置換の並列実行速度において最も高速である。Ice Lake 以外の環境では、Simpira-4 と同じ速度であったが、Ice Lake 環境においては既存のラージブロック置換より 10%以上高速である。

提案する全ての置換は、1 ブロックを暗号化する実行速度において Simpira v2 より高速であった。これは、5.1 で予測した結果と同じであった。(2-0, 1-1) 構成は、256-bit 置

表 8 512-bit ブロック暗号の各アーキテクチャにおける 1 ブロック暗号化の実行速度

置換	Skylake	Kaby Lake	Coffee Lake	Ice Lake
Simpira-4	1.31	1.33	1.25	1.05
(1-0, 2-0, 2-0, 1-0)	0.82	0.88	0.81	0.70
(2-0, 1-0, 1-0, 2-0)	0.85	0.90	0.81	0.70
(2-0, 1-0, 2-0, 1-0)	0.92	0.93	0.87	0.74
(2-0, 2-0, 2-0, 2-0)	0.82	0.85	0.78	0.70
(2-0, 2-2, 2-0, 2-2)	0.70	0.74	0.67	0.61
Haraka v2	1.02	1.00	0.96	0.91
Pholkos	0.93	0.90	0.94	0.90

表 9 512-bit ブロック暗号の各アーキテクチャにおける並列暗号化処理の実行速度

置換	Skylake	Kaby Lake	Coffee Lake	Ice Lake
Simpira-4	0.65	0.63	0.63	0.37
(1-0, 2-0, 2-0, 1-0)	0.69	0.66	0.66	0.34
(2-0, 1-0, 1-0, 2-0)	0.69	0.66	0.66	0.34
(2-0, 1-0, 2-0, 1-0)	0.69	0.66	0.66	0.35
(2-0, 2-0, 2-0, 2-0)	0.79	0.76	0.75	0.39
(2-0, 2-2, 2-0, 2-2)	0.65	0.63	0.63	0.33
Haraka v2	0.71	0.92	0.80	0.58
Pholkos	0.67	0.79	0.75	0.63

換の並列実行環境において、Ice Lake 環境で Haraka v2, Pholkos より高速であった。これは、AES-NI 命令のレイテンシが 3 になることで、多くの AES-NI 命令をパイプライン実行できる Simpira v2 や提案構成が、AES-NI 命令の他にワード置換を行うための SIMD 命令が必要となる Haraka v2 や Pholkos よりも並列実行による暗号化処理に必要なクロックサイクル数が少なくなったためであると考えられる。

6. まとめ

本稿では、AES-NI 命令のみから構成される 256-bit, および 512-bit 置換について新たな構成を提案し、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により、既存のラージブロック置換である Simpira v2, Haraka v2, Pholkos と比較を行った。結果として、本稿で提案する構成は、256-bit 置換で並列実行を行う環境を除く全ての環境において最も高速であった。また、256-bit 置換で並列実行を行う環境においても、Ice Lake 環境においては最も高速であった。今後、AES-NI 命令のレイテンシは小さくなると予測できるので、本稿の構成がラージブロック置換の実装を検討する上で重要になることが期待できる。

謝辞 本研究は JST さきがけ (JPMJPR2031) と科研費基盤 (B19H02141) の助成を受けたものである。

参考文献

- [1] Shay Gueron and Nicky Mouha. Simpira v2: A family of efficient permutations using the aes round function. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT (1)*, Vol. 10031 of *Lecture Notes in Computer Science*, pp. 95–125, 2016.
- [2] Shay Gueron. Intel advanced encryption standard (aes) new instructions set, 2010.
- [3] 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.
- [4] 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.
- [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.