スポンジ構造を基にした 超高速かつ低消費電力な認証暗号の設計

阪本 光星[1,2,a)]　Subhadeep Banik[3]　Andrea Caforio[4]　石川 達也[2]　五十部 孝典[2]
Mustafizar Rahman[2]

概要：Beyond 5G の発達にともない，さらなる需要の増加が予想されるデータセンターでは，これを運用するために莫大な電力が消費されている．国際エネルギー機関の報告によると，2022 年には約 460TWh であった消費電力量は，2026 年には 1000TWh 以上に到達する可能性があると報告されており，低消費電力なクリーンな技術の開発は各分野において急務である．これらをふまえて，本稿では低消費電力認証暗号 AETHER を提案する．設計においては，認証暗号 AEGIS 及び Rocca の内部構造を見直すことで，暗号化速度と消費電力の両方の観点で最適化を行う．具体的には，AEGIS 及び Rocca のラウンド関数内部で利用されている AES ラウンド関数を，遅延と消費電力の観点で最適化された暗号学的置換に置き換る．また，安全性の観点では，設計した暗号学的置換を基に，偽造攻撃に対して 128 ビット安全性を達成するラウンド更新関数の設計を行う．結果として，AETHER は同等の安全性を持つ AES-256-GCM の約 136 倍の 3.15Tbps の暗号化速度を達成し，かつ約 1/32 の消費電力量を実現する．

キーワード：認証暗号，低消費電力，Beyond 5G

Design of an Ultra-High Throughput and Low-Energy Authenticated Encryption Scheme

Kosei Sakamoto[1,2,a)]　Subhadeep Banik[3]　Andrea Caforio[4]　Ishikawa Tatsuya[2]
Isobe Takanori[2]　Mustafizar Rahman[2]

Abstract: In this paper, we introduce AETHER, an authenticated encryption scheme that achieves ultra-high throughput and low energy consumption, supporting a 256-bit key and a 128-bit tag. While inspired by an AEGIS-like structure, AETHER stands out with a completely redesigned round-update function. We replace the AES round function with a new inner function optimized for ultra-low latency and energy consumption. This function incorporates Orthros's S-box and a 16 × 16 binary matrix from Akleylek et al., leading to a 1.56 times reduction in energy consumption and a 1.25 times reduction in delay compared to the AES round function. To further optimize hardware performance, we design the general construction of the round-update function to be more hardware-friendly, allowing parallel execution of the inner function on all 128-bit words, thereby enhancing both throughput and security against collision-based forgery attacks. AETHER achieves a remarkable throughput of 3.15 Tbit/s and an energy consumption of only 204.31 nJ, outperforming existing AEADs.

Keywords: Authenticated encryption, Low energy, High throughput, Beyond 5G

[1] 三菱電機株式会社, Mitsubishi Electric Corporation, Kamakura, Japan
[2] 大阪大学, The University of Osaka, Osaka, Japan
[3] University of Lugano, Lugano, Switzerland
[4] lowRISC C.I.C., Cambridge, United Kingdom

[a)] Sakamoto.Kosei@dc.MitsubishiElectric.co.jp

1. はじめに

1.1 背景

高画質ストリーミング，クラウド・コンピューティング，IoT（Internet of Things）に後押しされたデータ消費の急増により，データセンターにはかつてないデータ処理速度が求められている．Cisco の Annual internet report によると，世界の IP トラフィック量は 2021 年に 3 ゼタバイトに達し，クラウドに保存されるデータは 2025 年までに 100 ゼタバイトに膨れ上がると予測されている[*1]．Beyond 5G アプリケーションやリアルタイムのデータ処理と広帯域アプリケーションに対するこの急増する需要に対応するため，データセンター・ネットワークはテラビット毎秒（Tbps）のスループットを実現する処理性能が求められる．

消費電力については，データセンターの需要の増加により，データーセンターで消費される電力の増加が環境・社会・経済面で大きな問題となっている．国際エネルギー機関（IEA）の年次電力報告書によると，データセンターの消費電力は 2022 年に 460TWh，最悪のシナリオでは 2026 年までに 1,000TWh 以上に増加する可能性があると示されている[*2]．実際に，米国エネルギー省は，データセンターの環境への影響を軽減するため，エネルギー効率の高い技術を開発することの重要性を強調している．これらのことから，データセンター等で利用される暗号方式についても，将来的に超高スループット，かつ低消費電力な暗号方式が要求されることが予想される．

1.2 既存研究

既存研究において，Banik らにより，アンロールド実装されたストリーム暗号がハードウェア実装において 1 サイクルで複数ビットのキーストリームを出力できることから，消費電力の観点で優れた性能を実現可能であることが示されている [6]．また，Caforio らは，ストリーム暗号内部の内部状態更新関数と消費電力の関係性を調査し， 既存のストリーム暗号を消費電力の観点において最適化した Trivium-LE and Triad-LE を提案した [7].

暗号化処理速度については，Tiaoxin [10]， Rocca [11]，および Rocca-S [2] に代表される，認証暗号 AEGIS [14] を基にしたソフトウェア向け認証暗号（AEGIS-like な認証暗号）がハードウェア上でも高速な暗号化処理を実現可能であることが示されている．実際に，Anand らは，Rocca-S の提案論文において，認証暗号内部のラウンド更新関数を１サイクルで更新し，そのたびに平文/暗号文ブロックを入力/出力することで，ハードウェア実装においても 1 Tbps 以上のスループットを実現可能であることが，提案論文で示している [2].

1.3 貢献

本稿では，超高速，かつ低消費電力性能の両方を実現する認証暗号 AETHER を提案する．AETHER は，256 ビットの秘密鍵と 128 ビットの認証タグを提供する AEGIS-like な認証暗号である．設計手法としては，AEGIS-like な認証暗号のラウンド更新関数内部で利用されている AES ラウンド関数を，よりハードウェア実装に適した内部関数に置き換えることで低消費電力，かつ高スループットな認証暗号を構成する．具体的には，ラウンド更新関数で利用する内部関数を Orthros の 4 ビット S-box[5] と Akleylek らが示した 16×16 バイナリ行列 [1] で構成する．その後，内部衝突を用いた偽造攻撃に対して 128 ビット安全性を満たすラウンド更新関数を設計する．

結果として，AETHER は 1.28 メガビットの平文の暗号化をする場合，AES-256-GCM の約 135 倍である 3.15 Tbps のスループットを達成し，かつ約 1/32 である 204.31 nJ の消費電力を実現する．表 1 に AETHER のハードウェア実装評価結果を示す．

表 1: AETHER のハードウェア実装評価結果.

	Area		Latency	Throughput	Power	Energy (nJ)	
	μm^2	GE	ps	Tbit/s	mW	Short	Long
AETHER	10598	53904	130	3.15	0.605	2.965	204.31

2. 認証暗号 AETHER

認証暗号 AETHER は任意長の平文 M と関連データ AD（Associated data）及び，128 ビットのナンス N と 256 ビットの秘密鍵 K を入力として持つ．出力としては，入力された平文と同じ長さの暗号文 C と 128 ビットの認証タグ T を持つ．復号プロセスでは，受信者は関連データ AD と暗号文 C 及びナンス N と秘密鍵 K から認証タグ T' を計算し，受信した認証タグ T と計算した認証タグ T' が一致した場合，復号された平文 M を得る．一致しない場合，エラーが返される．以下に AETHER の暗号プロセス $\mathsf{ENC}_{\mathsf{AETHER}}$ と復号プロセス $\mathsf{DEC}_{\mathsf{AETHER}}$ を示す．

$$\mathsf{ENC}_{\mathsf{AETHER}}(K, N, M, AD) \to (C, T),$$

$$\mathsf{DEC}_{\mathsf{AETHER}}(K, N, C, AD, T) \to \begin{cases} M & \text{if } T = T' \\ \perp & \text{otherwise} \end{cases}.$$

2.1 ラウンド更新関数の仕様

AETHER では 1152 ビット（9×128 ビット）の内部状態をラウンド更新関数により随時更新することで，暗号化（復号）を行う．図 1 にラウンド更新関数を示す．

[*1] https://cybersecurityventures.com/the-world-will-store-200-zettabytes-of-data-by-2025/

[*2] https://www.datacenterdynamics.com/en/news/global-data-center-electricity-use-to-double-by-2026-report/

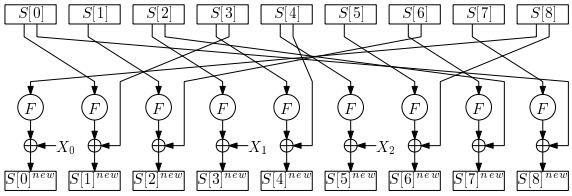


図 1: ラウンド更新関数.

ここで $S[i]$ と $S[i]^{new}$ はそれぞれ更新前と更新後の 128 ビットワードを示し，X_0, X_1 および X_2 はラウンド更新関数の入力を示す．以降では，入力 $\boldsymbol{X} = (X_0, X_1, X_2)$ について，内部状態 $\boldsymbol{S} = (S[0], \cdots, S[8])$ を更新する処理を，$R(\boldsymbol{S}, \boldsymbol{X})$ とする．また，図 1 のおける F は各 128 ビットワードを処理する暗号学的置換であり，以下の式で示される．

$$F = \mathsf{Permutation} \circ \mathsf{ApplySbox} \circ \mathsf{MatrixMul} \circ \mathsf{ApplySbox}. \tag{1}$$

以下に，式 (1) における各処理の説明を行う．

Permutation． 128 ビットステートに対し，以下に示す 4 ビット単位の置換を適用する．

$$x_i \xrightarrow{\mathsf{Permutation}} \begin{cases} x_{i/2} & \text{if } i \text{ is even} \\ x_{(i-1)/2+16} & \text{otherwise} \end{cases}.$$

ここで，x_i は 128 ビットステートにおける i 番目のニブルを示す．

ApplySbox． 128 ビットステートに対し，表 2 に示す 4 ビット S-box を適用する．

表 2: 4 ビット S-box.

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
$Sbox(x)$	1	0	2	4	3	8	6	d	9	a	b	e	f	c	7	5

MatrixMul． 128 ビットステートに対し，以下に示す 4 ビット単位の 16×16 行列 M_b を並列に 2 回適用する．

$$M_b = \begin{pmatrix} M_0 & M_0 & M_0 & M_1 \\ M_0 & M_0 & M_1 & M_0 \\ M_0 & M_1 & M_0 & M_0 \\ M_1 & M_0 & M_0 & M_0 \end{pmatrix}.$$

ここで，M_0 は 4 ビット単位の 4×4 の単位行列であり，M_1 は全ての成分が 1 の 4×4 の行列である．したがって，各ニブルは以下の通りに更新される．

$$(x_{16i+0}, \cdots, x_{16i+15})^T \leftarrow M_b \cdot (x_{16i+0}, \cdots, x_{16i+15})^T, \quad i \in \{0, 1\}.$$

ここで，$(x_{16i+0}, \cdots, x_{16i+15})^T$ は転置行列を表す．

2.2 AETHER の詳細な仕様

AETHER は初期化，関連データ処理，暗号化（復号），及び最終化の 4 つの処理部からなる．図 2 に AETHER の概要図を示す．

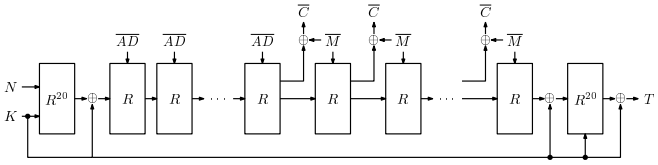


図 2: AETHER の概要図.

以降では，各処理部について説明を行う．

初期化 はじめに，256 ビットの K が 2 つの 128 ビット鍵 K_0 と K_1 ($K = K_0||K_1$) に分割され，N とともに内部状態として以下の通りにロードされる．

$$S[0] = Z_1, \quad S[1] = K_0, S[2] = N + K_0, \quad S[3] = 0, \quad S[4] = Z_0,$$
$$S[5] = 0, \qquad S[6] = N, S[7] = K_1, \qquad S[8] = Z_2.$$

ここで，Z_0，Z_1 及び Z_2 はラウンド定数であり，以下の通りである．

$$Z_0 = \texttt{0x428a2f98d728ae227137449123ef65cd},$$
$$Z_1 = \texttt{0xb5c0fbcfec4d3b2fe9b5dba58189},$$
$$Z_2 = \texttt{0x7137449123ef65cd428a2f98d728ae22}.$$

次に，$R(\boldsymbol{S}, \boldsymbol{Z})$ を 20 回適用し，その後，以下に示す通りに鍵のフィードフォワード処理を行う．

$$S[0] = S[0] \oplus K_0, \ S[1] = S[1] \oplus K_0, \ S[2] = S[2] \oplus K_0,$$
$$S[3] = S[3] \oplus K_0, \ S[4] = S[4] \oplus K_1, \ S[5] = S[5] \oplus K_0,$$
$$S[6] = S[6] \oplus K_1, \ S[7] = S[7] \oplus K_1, \ S[8] = S[8] \oplus K_1.$$

関連データ処理 はじめに，パディング処理として関連データ AD は 384 の倍数のビット長となるよう末尾に `0x100..0` が追加される．その後，AD は１２８ビットブロック毎に以下の通りに分割される．

$$\overline{AD} = (\overline{AD_0}||\overline{AD_1}||\ldots||\overline{AD_{i-1}}), \ i = \frac{|\overline{AD}|}{128}.$$

ここで，$\overline{AD}$ はパディング処理後の AD を表し，$|\overline{AD}|$ はパディング処理後の AD のビット長を表す．次に，以下の通りに $\overline{AD}$ を入力する．

$$R(\boldsymbol{R}, \overline{AD_{3j}}, \overline{AD_{3j+1}}, \overline{AD_{3j+2}}), \ 0 \le j < \frac{|\overline{AD}|}{384}.$$

暗号化 はじめに，関連データ処理と同様のパディング処理を行い $\overline{M}$ を生成する．次に，以下の通りに $\overline{M}$ を入力することで内部状態を更新し，暗号文 $\overline{C}$ を生成する．

$$\overline{C_j} = F(S[0] \oplus S[1]) \oplus S[4] \oplus \overline{M_j},$$
$$\overline{C_{j+1}} = F(S[2] \oplus S[6]) \oplus S[7] \oplus \overline{M_{j+1}},$$
$$\overline{C_{j+2}} = F(S[3] \oplus S[5]) \oplus S[8] \oplus \overline{M_{j+2}},$$
$$R(\boldsymbol{R}, \overline{M_{3j}}, \overline{M_{3j+1}}, \overline{M_{3j+2}}),\ 0 \le j < \frac{|\overline{M}|}{384}.$$

最後に，パディング処理を行った場合は，パディングビットの切り詰めを行う．

最終化 はじめに，以下に示す通りに鍵のフィードフォワード処理を行う．

$$S[0] = S[0] \oplus K_0,\ S[1] = S[1] \oplus K_0,\ S[2] = S[2] \oplus K_1,$$
$$S[3] = S[3] \oplus K_1,\ S[4] = S[4] \oplus K_0,\ S[5] = S[5] \oplus K_0,$$
$$S[6] = S[6] \oplus K_1,\ S[7] = S[7] \oplus K_0,\ S[8] = S[8] \oplus K_1.$$

次に，$R(\boldsymbol{R}, K_0, Z_0, K_1)$ を 20 回適用し，以下の通りに鍵のフィードフォワード処理を行う．

$$S[0] = S[0] \oplus K_1,\ S[1] = S[1] \oplus K_0,\ S[2] = S[2] \oplus K_0,$$
$$S[3] = S[3] \oplus K_0,\ S[4] = S[4] \oplus K_1,\ S[5] = S[5] \oplus K_0,$$
$$S[6] = S[6] \oplus K_0,\ S[7] = S[7] \oplus K_1,\ S[8] = S[8] \oplus K_1.$$

最後に，以下の通りに認証タグ T を生成する．

$$\bigoplus_{i=0}^{8} S[i] = T.$$

2.3 主張する安全性

Nonce-respecting な仮定において，AETHER は鍵回復攻撃に対して 256 ビット安全性を主張し，識別攻撃に対して 128 ビット安全性を主張する．メッセージ長および関連データ長については，固定鍵においてそれぞれ 2^{128} および 2^{64} ブロックまでとする．また，固定鍵において，異なるメッセージを暗号化できる最大数は 2^{128} とする．

3. 認証暗号 AETHER の設計

図 3 に Rocca 及び，Rocca-S の提案論文において示されている AEGIS-like な認証暗号におけるラウンド更新関数の一般構造を示す．図 3 における A は 1 ラウンドの AES ラウンド関数を示しており，各 128 ビットステートは 1 ラウンドの AES ラウンド関数と 1 回の XOR，もしくはそのどちらか一方のみ適用される．

本稿では，図 3 における AES ラウンド関数を，他の暗号学的置換に置き換えることで消費電力および，スループットの最適化を行う．その後，ラウンド更新関数の一般構造を見直し，偽造攻撃に対して 128 ビット安全性を保証するラウンド更新関数の設計を行う．以降では，置き換える暗号学的置換を内部関数と呼ぶ．

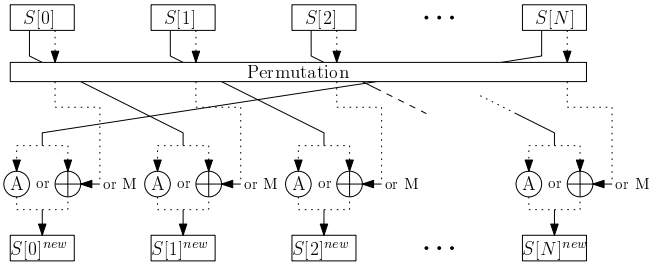


図 3: AEGIS-like な認証暗号におけるラウンド更新関数の一般構造．

3.1 内部関数の設計

AEGIS-like な認証暗号では，ラウンド更新関数を適用するたびに入力される平文ブロックを用いて暗号文ブロックを生成し，内部状態の更新を行う．したがって，ラウンド更新関数内部で適用される内部関数の遅延性能が暗号化速度（スループット）に大きく影響する．そこで，本稿では，遅延性能と消費電力性能の両方の観点で優れた内部関数の設計を行う．以下に，設計する内部関数の実装性能と安全性について述べる．

実装性能要件． 実装性能の観点では，Banic らにより，SPN (Substitution Permutation Network) 構造を採用することで，消費電力と遅延性能の観点で優れたブロック暗号を設計可能であることが示されている [4]．そこで，本稿では消費電力および，遅延の両方が 1 ラウンドの AES ラウンド関数のそれらよりも小さい内部関数を設計する．

安全性要件． AEGIS-like な認証暗号では，AES ラウンド関数を利用することで，内部衝突に対する偽造攻撃に対して十分な安全性を保証することを可能としている．具体的には，2 ラウンドの AES ラウンド関数は 2^{-30} の差分特性確率を保証できるため，これにより内部衝突による偽造攻撃に対して 128 ビットセキュリティを保証することを容易にしている．そこで本稿では，連続した 2 つの内部関数の差分特性確率が 2^{-30} 以下となる内部関数を設計する．

3.1.1 非線形層（S-box）の設計

S-box については，既存の低遅延・低消費電力暗号の設計の際に，様々な低遅延・低消費電力な S-box が提案されている．表 3 に主な低遅延・低消費電力な S-box と AES の S-box の暗号学的特性と，消費電力，および遅延性能を示す．ここで，表 3 において，$\mathcal{E}$ と $\mathcal{L}$ はそれぞれ消費電力性能と遅延性能に最適化した実装時における結果を示す．また，DP 及び C^2 は差分確率と線形確率を示す．なお，表 3 の結果の評価環境は NanGate 15 nm セルライブラリを用いており，クロック周波数は 100 MHz である．以降の実装評価においても，特に記載がない限り，同様の環境で評価を行うものとする．

表 3 より，Midori で用いられている 4 ビット S-box であ

表 3: 各 S-box の実装評価結果および，暗号学的特性.

Scheme	Width	Area (μm^2)		Delay (ps)		Energy (pJ)		DP	C^2	Degree	Full
		$\mathcal{L}$	$\mathcal{E}$	$\mathcal{L}$	$\mathcal{E}$	$\mathcal{L}$	$\mathcal{E}$				Diffusion
AES	8	248.17	136.15	24.96	96.88	0.5908	0.1105	2^{-6}	2^{-6}	7	✓
BipBip	6	17.89	12.04	13.70	24.99	0.0414	0.0358	2^{-4}	2^{-4}	2	-
SPEEDY	6	15.08	9.73	7.43	11.97	0.0389	0.0198	2^{-3}	$2^{-2.83}$	5	✓
Gleeok (χ)	5	9.04	9.73	6.61	12.97	0.0244	0.0088	2^{-2}	2^{-2}	2	-
Orthros	4	5.99	3.34	5.17	9.74	0.0144	0.0080	2^{-2}	2^{-2}	3	✓
Midori (Sb0)	4	6.93	3.04	4.81	8.96	0.0178	0.0059	2^{-2}	2^{-2}	3	-
Midori (Sb1)	4	5.99	3.29	7.37	10.48	0.0138	0.0080	2^{-2}	2^{-2}	3	✓
QARMAv2 (ρ)	4	5.55	3.29	7.37	10.86	0.0137	0.0079	2^{-2}	2^{-2}	3	✓
QARMAv2 (σ_0)	4	9.58	3.19	4.68	16.51	0.0262	0.0075	2^{-2}	2^{-2}	3	-
PRINCE	4	5.94	3.78	6.05	15.62	0.0131	0.0093	2^{-2}	2^{-2}	3	✓
Gleeok (χ)	3	4.27	2.31	6.61	13.21	0.0112	0.0053	2^{-2}	2^{-2}	2	-

る Sb0 が遅延と消費電力の観点で最も優れていることがわかる．しかしながら，暗号学特性の観点では最適ではない．これらの観点から，本稿では，最適な暗号学的特性を有し，かつ遅延と消費電力の観点で最も優れている Orthros の S-box を内部関数に利用する．

3.1.2 線形層（行列演算）の設計

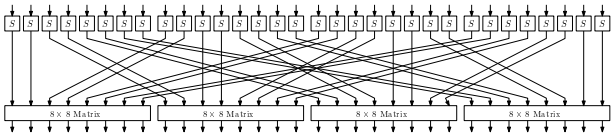
低遅延・低消費電力ブロック暗号内部で用いられる行列は，実装性能と安全性の観点から 4×4 almost MDS 行列（4×4 バイナリ行列）が用いたれることが多い．しかしながら，設計する内部関数は，少ない演算数で十分低い差分特性確率を持つ必要があるため，本稿では，より優れた拡散性能を有する 8×8，16×16，および 32×32 のバイナリ行列も候補とし，比較検討を行う．本稿では，Aslan らと，Akleylek らが示した，最適なブランチ数を有する 8×8[3]，16×16[1]，および 32×32[1] バイナリ行列を候補として採用する．各行列の詳細については，Aslan らおよび，Akleylek らの論文を参照されたい [1], [3].

表 4 にこれらのバイナリ行列と AES 内部の MDS 行列の実装評価結果を示す．ここで，各行列の要素のサイズは全て 4 ビットであり，行列数は 128 ビットの SPN 構造に適用することを想定している．

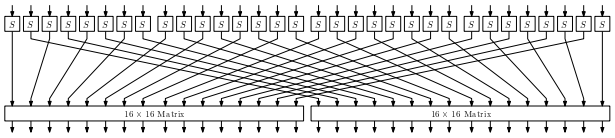
表 4 より，遅延と消費電力の観点で 4×4 バイナリ行列が最適である．しかしながら，暗号学的特性を考慮した場合，ブランチ数あたりの遅延と消費電力は，32×32 バイナリ行列が最も優れており，8×8，16×16 バイナリ行列も同等の性能を有していることがわかる．これらの結果から，本稿では，32×32，16×16 および，8×8 バイナリ行列を内部関数の線形演算の候補とする．

3.1.3 内部関数の構成

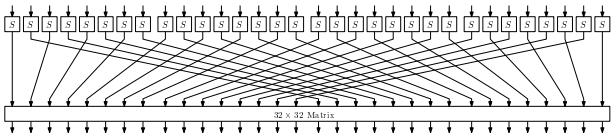
本項では，3.1.1 項と 3.1.2 項ので選択した，S-box とバイナリ行列を組み合わせることで，安全性と実装性能の観点で最適な内部関数の構成する．図 4a，4b および，4c に Orthros の S-box と 8×8，16×16 および，32×32 バイナリ行列を基にした，SPN 構造を持つラウンド関数をそれぞれ示す.



(a) ラウンド関数 F_8 （8×8 matrix）.



(b) ラウンド関数 F_{16} （16×16 matrix）.



(c) ラウンド関数 F_{32} （32×32 matrix）.

図 4: 32×32, 16×16, and 8×8 行列と Orthros の S-box を基にした SPN 構造を持つラウンド関数.

次に，AES ラウンド関数と，F_8，F_{16}，および F_{32} を基とし，3.1 節に示した安全性要求を満たす最小の構成についての実装評価結果と最大差分特性確率を表 5 に示す．なお，表 5 に示す S_{layer} は Orthros の S-box により構成された非線形層を示す.

表 5 より，3.1 節に示した安全性要求を満たす内部関数の候補において $S_{layer} \circ F_{16}$ が遅延と消費電力の観点で最良であることがわかる．これらの結果から，本稿では $S_{layer} \circ F_{16}$ をラウンド更新関数の内部関数として利用する.

3.2 ラウンド更新関数の設計

既存の AEGIS-like な認証暗号のラウンド関数の設計においては，ソフトウェア実装性能を考慮していることから，

表 4: 各行列の実装評価結果および，暗号学的特性

Matrix	Delay (ps)		Energy (pJ)		ブランチ数	Delay / BR	Energy/ BR *
	$\mathcal{L}$	$\mathcal{E}$	$\mathcal{L}$	$\mathcal{E}$	(BR)	(with better delay)	(with better energy)
MDS matrix in AES	16.00	28.64	0.1574	0.1463	5	3.2	0.1170 (0.0292×4)
4×4 binary matrix [4]	9.82	12.39	0.0558	0.0514	4	2.45	0.1028 (0.0128×8)
8×8 binary matrix [3]	15.62	21.67	0.1141	0.0825	5	3.12	0.0660 (0.0165×4)
16×16 binary matrix [1]	21.01	37.22	0.3030	0.3220	8	2.62	0.0757 (0.0378×2)
32×32 binary matrix [1]	25.34	42.78	0.9023	1.2540	12	2.11	0.0751

* 適用する行列数は，128 ビットの SPN 構造に適用されることを想定.

表 5: 各内部関数の候補の実装性能と最大差分特性確率 (DCP_{max})

内部関数	構成	DCP_{max}	Delay (ps)		Energy (pJ)	
		(1 ラウンド)	$\mathcal{L}$	$\mathcal{E}$	$\mathcal{L}$	$\mathcal{E}$
F_{AES}	F_{AES}	2^{-30} (2^{-6})	54.79	131.79	8.7880	2.8040
F_8	$F_8 \circ F_8$	2^{-32} (2^{-10})	70.85	91.85	2.8320	2.5440
F_{16}	$S_{layer} \circ F_{16}$	2^{-32} (2^{-16})	43.81	67.49	1.9190	1.7966
F_{32}	$S_{layer} \circ F_{32}$	2^{-48} (2^{-24})	50.90	79.51	2.1936	2.9580

ラウンド更新関数内部で利用する AES ラウンド関数の適用数に制限を加える必要がある．一方，ハードウェア実装の際は，遅延性能と消費電力性能を向上させるためにアンロールド実装を前提としているため，内部関数の適用数によらず，全ての内部関数を並列実行することが可能である．この場合，暗号化処理速度は主にクリティカルパスにより決定される遅延性能に依存するため，内部関数の適用数に大きく影響を受けない．また，安全性についても既存研究に示されている通り，内部関数の適用数が増えるにつれ内部衝突を用いた偽造攻撃に対する安全性を保証することが容易となる [2], [11]．しかしながら，内部関数の適用数が増加するにつれ回路実装規模が大きくなるため，消費電力は増加する．

そこで本稿では，ラウンド更新関数に入力する平文ブロック数を増やすことにより，1 ビットあたりの平文の暗号化の際に要求される消費電力の向上を図る．図 5 に本稿で設計するラウンド更新関数の一般構造を示す．

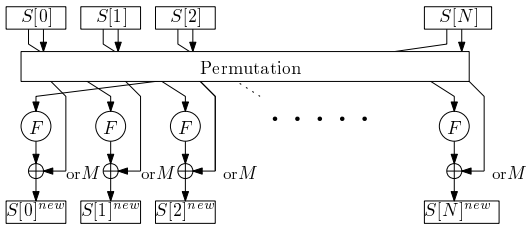


図 5: 設計するラウンド更新関数の一般構造.

以下にラウンド更新関数の実装性能要求と安全性要求を示す．以降では $\#M$ と $\#S$ は，それぞれ入力する平文ブロック数と 128 ビットワード数を示す．

実装性能．

1．（消費電力性能，遅延性能） 可能な限り多くの $\#M$ を持つ構成.

2．（消費電力性能） 可能な限り少ない $\#S$ を持つ構成.

安全性． 内部衝突を利用した偽造攻撃に対して，128 ビット安全性を保証する構成.

3.2.1 内部衝突を用いた偽造攻撃に対して 128 ビット安全性を保証する構成の探索

本項では，内部衝突を用いた偽造攻撃に対して 128 ビット安全性を保証する構成を探索する．探索の際には，Sun らの SAT を用いた最大差分特性確率の評価手法により，安全性評価を行う [12]．評価するラウンド更新関数の候補数は $\#M = m$ および，$\#S = s$ とした場合，$s! \times \binom{s}{m} \times (m!)^{-1}$ となる．ここで，$\#M$ と $\#S$ が増加するにつれ，評価する候補数は膨大な数となるため，本稿では各構成の評価時間を可能な限り短くするために，ワード単位の切詰差分による内部衝突耐性評価を行う．ワード単位の切詰差分評価においては，表 5 に示される通り内部関数は差分が入力された場合，2^{-16} の差分特性確率を保証するため，本稿では 8 個以上の差分が入力される内部関数（アクティブな内部関数，$\#AF$）を保証するラウンド更新関数の探索を行う ($2^{-128} \leq 2^{-16\times 8}$)．探索結果を表 6 に示す.

表 6: 内部衝突耐性の安全性評価結果.

$\#S$	$\#M$	Total	# searched	# found	$\#S$	$\#M$	Total	# searched	# found
4	**1**	**96**	**All**	**16**	8	3	376320	All	0
4	2	72	All	0	8	4	117600	All	0
4	3	16	All	0	8	5	18816	All	0
5	2	600	All	0	8	6	1568	All	0
5	3	200	All	0	8	7	64	All	0
5	4	25	All	0	**9**	**3**	**5080320**	**All**	**8478**
6	**2**	**5400**	**All**	**54**	9	4	1905120	All	0
6	3	2400	All	0	9	5	381024	All	0
6	4	450	All	0	9	6	42336	All	0
6	5	36	All	0	9	7	2592	All	0
7	3	29400	All	0	9	8	81	All	0
7	4	7350	All	0	10	4	31752000	All	0
7	5	882	All	0	10	5	38102400	All	0
7	6	49	All	0	11	4	548856000	2^{26}	0

表 6 より，$\#S = 9$ かつ，$\#M = 3$ の構成において，内部衝突による偽造攻撃に対して 128 ビット安全性を保証する構成が 8478 個発見した．これらの構成をさらに絞り込むために，拡散性能の評価を行い，これら 8478 個のの構成の中で 5 個の構成が拡散性能の観点で最適であった．これらの結果から，本稿ではこの 5 個の構成の 1 つを AETHER

のラウンド更新関数として決定する．決定したラウンド更新関数は図 1 に示す通りである．

4. 安全性評価

本章では，AETHER の差分攻撃，線形攻撃，integral 攻撃，および偽造攻撃に対する安全性評価について述べる．

4.1 差分攻撃

表 7 に初期化部の各ラウンド数における $\#AF$ の下界を示す．鍵回復攻撃に対する 256 ビット安全性を保証するためには $\#AF \geq 16$ ($2^{-256} \leq 2^{-16\times16}$) を保証する必要があるが，5 ラウンドにおいて既にこれを上回るため，AETHER は差分攻撃に対して安全であると期待できる．

表 7: 初期化部の各ラウンド数における $\#AF$ の下界．

Rounds	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
# AF	2	4	8	12	16	19	22	24	28	34	36	39	43	45	48	50

4.2 線形攻撃

AEGIS-like な認証暗号では暗号化部において生成される暗号文ブロックの linear bias が最も強力な識別攻撃になりうることがしられているため，この評価を Eichlseder らの手法を用いて行う [8]．結果として，暗号文ブロックの linear bias の上界は 2^{-208} であった．Linear bias による識別攻撃に対して，128 ビット安全性を保証するためには，linear bais の上界が 2^{-128} であることを示す必要があるため，AETHER は linear bias による識別攻撃に対して安全であると期待できる．

4.3 Integral 攻撃

本項では，integral 識別子が構築できる最長のラウンド数を見積もることにより，inetgral 攻撃に対する安全性評価を行う．Integral 特性を探索する際は Todo により提案された division property を評価することで最長の integral 識別子を探索する [13]．結果として，初期化部における最長の識別子として 8 ラウンドの integral 識別子が構築可能であることがわかった．初期化部は 20 ラウンドであることを考慮すると，AETHER は integral 攻撃に対して安全であると期待できる．

4.4 偽造攻撃

内部衝突による偽造攻撃に対しては，3.2.1 項において，AETHER のラウンド更新関数は 128 ビット安全性を満たすことが示されている．ここでは，ビット単位の差分特性評価を行うことで，より詳細な安全性評価を行う．結果として，内部衝突を発生させる差分特性の差分特性確率の上界は 2^{-164} であった．したがって，AETHER 内部衝突による偽造攻撃に対して安全であることが期待できる．

5. ハードウェア実装評価

本章では AETHER のハードウェア実装評価を行う．比較対象として，米国標準化研究所により標準化された軽量暗号 Ascon-128a と 256 ビット鍵をサポートする AEGIS-256，AES-256-GCM，および Rocca-S を用いる．消費電力については，1024 ビットの平文（short）と 1.28×10^6 ビットの平文（long）を暗号化する場合についてそれぞれ評価を行う．なお，評価環境は，NanGate 15 nm セルライブラリを用いており，クロック周波数は 10 NHz である．また，AES は様々なハードウェア実装手法が提案されているため，暗号化関数内部で AES を用いている AEGIS-256，AES-256-GCM，および Rocca-S については以下に示す実装手法を用いた際の実装評価をそれぞれ行う．

LUT. AES の S-box をテーブル参照で回路を合成する．

SMALL. Maximov らにより提案さされた AES の S-box をマニュアルで最適化し，回路を合成する手法を用いる [9]．

DSE. AES のテーブル参照の入出力に特定のエンコーダーとデコーダーを配置し，回路を合成する．

TT. AES のソフトウェア実装の際に用いられる T-box により回路を合成する．

なお，AETHER については LUT と同様の手法を用いて実装する．ここで，1 章に示した表 1 の結果は，AETHER の遅延に最適化された回路による結果であることに留意されたい．

図 6 に評価結果を示す．

図 6 から，スループットの観点では AETHER は 2 Tbps を達成しており，表 1 に示す遅延に最適化された回路における評価においては 3 Tbps に達する．これは，比較した認証暗号の中で最速であり，既存方式で最も高速な Rocca-S の約 2 倍のスループットを実現している．

消費電力の観点では，短い平文と長い平文を暗号化した場合において，それぞれ 2.962 nJ と 204.19 nJ の消費電力を要する．他の認証暗号と比較した場合，短い平文の場合においては，AEGIS-256 と同等の消費電力性能を実現し，長い平文については，他の暗号方式の中で最も消費電力性能に優れる AEGIS-256 の約半分の消費電力を実現している．ここで短い平文を暗号化する場合は，初期化部で消費される電力が支配的になるため，他の AEGIS-like な認証暗号と消費電力の観点では同等の性能になることに留意されたい．

6. まとめ

本稿では低消費電力かつ，高スループットの両方を実現する認証暗号 AETHER を提案した．AETHER はソフトウェア上で高スループットを実現する AEGIS-like な認証暗

		Rocca-S			
AETHER	**10504**	*LUT*	22832	*DSE*	22931
	53428		116130		116138
Ascon-128a	7789	*SMALL*	11184	*TT*	28579
	39619		56889		145364

AEGIS-256				AES-256-GCM			
LUT	17403	*DSE*	17520	*LUT*	10023	*DSE*	10101
	88521		89116		50980		51381
SMALL	8703	*TT*	21743	*SMALL*	8265	*TT*	12599
	44266		110591		42038		64082

(a) Area (μm^2 / GE)

		Rocca-S			
AETHER	**0.185**	*LUT*	0.179	*DSE*	0.177
	2.066		1.431		1.451
Ascon-128a	0.583	*SMALL*	0.232	*TT*	0.154
	0.219		1.102		1.653

AEGIS-256				AES-256-GCM			
LUT	0.167	*DSE*	0.165	*LUT*	0.349	*DSE*	0.349
	0.766		0.776		0.023		0.023
SMALL	0.210	*TT*	0.132	*SMALL*	0.349	*TT*	0.349
	0.610		0.970		0.023		0.023

(b) Latency (ns) / Throughput (Tbit/s)

		Rocca-S			
AETHER	**49**	*LUT*	44	*DSE*	44
	2.962		6.165		3.368
Ascon-128a	28	*SMALL*	44	*TT*	44
	4.113		5.522		3.876

AEGIS-256				AES-256-GCM			
LUT	48	*DSE*	48	*LUT*	266	*DSE*	266
	5.309		2.945		13.85		13.85
FAST	48	*TT*	48	*SMALL*	266	*TT*	266
	4.868		3.317		13.36		15.33

(c) Energy Short (Cycles / nJ)

		Rocca-S			
AETHER	**3377**	*LUT*	5036	*DSE*	5036
	204.19		705.6		385.5
Ascon-128a	10012	*SMALL*	5036	*TT*	5036
	1470.8		632.1		443.7

AEGIS-256				AES-256-GCM			
LUT	10032	*DSE*	10032	*LUT*	160010	*DSE*	160010
	1109		615.6		8328		6674
FAST	10032	*TT*	10032	*SMALL*	160010	*TT*	160010
	1017		693.3		8035		9224

(d) Energy Long (Cycles / nJ)

		Rocca-S			AEGIS-256			AES-256-GCM		
AETHER	**0.605**	*LUT*	1.401	*DSE*	0.765	*LUT*	1.106	*DSE*	0.613	*LUT*	0.521	*DSE*	0.417
Ascon-128a	2.812	*SMALL*	1.255	*TT*	0.881	*SMALL*	1.014	*TT*	0.691	*SMALL*	0.502	*TT*	0.577

(e) Power (mW @ 10 MHz)

図 6: AETHER , Ascon-128a , AES-256-GCM , AEGIS-256 , および Rocca-S のハードウェア実装評価結果.

号の内部で利用されている AES ラウンド関数を，ハードウェア実装に適した他の内部関数に置き換え，それに合わせてラウンド更新関数の再設計を行うことで，低消費電力と高スループットの両方を実現する暗号方式となっている．

結果として，AETHER は既存方式である Rocca-S と比較し，約 2 倍の 3.15 Tbps のスループットを実現し，同時に Rocca-S の約 1/2 である 204.31 の消費電力性能を実現する．

謝辞

本研究は, JSPS 科研費 JP24H00696 と JST AIP 加速課題 JPMJCR24U1 の支援を受けたものである．

参考文献

[1] S. Akleylek, V. Rijmen, M. T. Sakalli, and E. Öztürk. Efficient methods to generate cryptographically significant binary diffusion layers. *IET Inf. Secur.*, 11(4):177–187, 2017.

[2] R. Anand, S. Banik, A. Caforio, K. Fukushima, T. Isobe, S. Kiyomoto, F. Liu, Y. Nakano, K. Sakamoto, and N. Takeuchi. An ultra-high throughput aes-based authenticated encryption scheme for 6g: Design and implementation. In G. Tsudik, M. Conti, K. Liang, and G. Smaragdakis, editors, *Computer Security - ESORICS 2023 - 28th European Symposium on Research in Computer Security, The Hague, The Netherlands, September 25-29, 2023, Proceedings, Part I*, volume 14344 of *Lecture Notes in Computer Science*, pages 229–248. Springer, 2023.

[3] B. Aslan and M. T. Sakalli. Algebraic construction of cryptographically good binary linear transformations. *Secur. Commun. Networks*, 7(1):53–63, 2014.

[4] S. Banik, A. Bogdanov, T. Isobe, K. Shibutani, H. Hiwatari, T. Akishita, and F. Regazzoni. Midori: A block cipher for low energy. In *ASIACRYPT (2)*, volume 9453 of *Lecture Notes in Computer Science*, pages 411–436. Springer, 2015.

[5] S. Banik, T. Isobe, F. Liu, K. Minematsu, and K. Sakamoto. Orthros: A low-latency PRF. *IACR Trans. Symmetric Cryptol.*, 2021(1):37–77, 2021.

[6] S. Banik, V. Mikhalev, F. Armknecht, T. Isobe, W. Meier, A. Bogdanov, Y. Watanabe, and F. Regazzoni. Towards low energy stream ciphers. *IACR Trans. Symmetric Cryptol.*, 2018(2):1–19, 2018.

[7] A. Caforio, S. Banik, Y. Todo, W. Meier, T. Isobe, F. Liu, and B. Zhang. Perfect trees: Designing energy-optimal symmetric encryption primitives. *IACR Trans. Symmetric Cryptol.*, 2021(4):36–73, 2021.

[8] M. Eichlseder, M. Nageler, and R. Primas. Analyzing the linear keystream biases in AEGIS. *IACR Trans. Symmetric Cryptol.*, 2019(4):348–368, 2019.

[9] A. Maximov and P. Ekdahl. New circuit minimization techniques for smaller and faster AES sboxes. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2019(4):91–125, 2019.

[10] I. Nikolić. Tiaoxin-346: Version 2.0. CAESAR Competition, 2014.

[11] K. Sakamoto, F. Liu, Y. Nakano, S. Kiyomoto, and T. Isobe. Rocca: An efficient aes-based encryption scheme for beyond 5g. *IACR Trans. Symmetric Cryptol.*, 2021(2):1–30, 2021.

[12] L. Sun, W. Wang, and M. Wang. Accelerating the search of differential and linear characteristics with the SAT method. *IACR Trans. Symmetric Cryptol.*, 2021(1):269–315, 2021.

[13] Y. Todo. Structural Evaluation by Generalized Integral Property. In E. Oswald and M. Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, volume 9056 of *Lecture Notes in Computer Science*, pages 287–314. Springer, 2015.

[14] H. Wu and B. Preneel. AEGIS: A fast authenticated encryption algorithm. *IACR Cryptol. ePrint Arch.*, page 695, 2013.