軽量暗号の安全性に関する調査及び評価 (Ascon, Grain-128AEAD, TinyJambu)

藤堂　洋介 *

2022 年 12 月

1　調査結果・評価結果の概要

昨今の IoT の広がりから、非常に安価な端末でも高い安全性を確保する必要性が高まってきている。これらの需要を受けて、アメリカ国立標準技術研究所 (NIST) は軽量暗号方式の標準化コンペティション (NIST LWC) を 2019 年に開始した。59 方式の Round 1 候補が 2019 年 4 月に発表され、同年 8 月に Round 2 候補として 32 方式まで絞られた。2021 年 3 月に Finalist として 10 方式 (ASCON, Elephant, GIFT-COFB, Grain-128AEAD, ISAP, PHOTON-Beetle, Romulus, SPARKLE, TinyJambu, Xoodyak) が選定された。本報告書は、上記 10 方式のうち、ASCON, Grain-128AEAD, および TinyJambu の安全性に関する既存結果の調査及び評価を与える。

1.1　ASCON

認証暗号 ASCON は 2014 年から 2019 年まで開催された認証暗号のコンペティション『CAESAR Competition』の候補として初めて提案された [DEMSb]。ASCON は CAESAR Competition のユースケース 1『Lightweight applications (resource constrained environments)』の第 1 選択に選定されている。最初の提案から 8 年が経過し、また、CAESAR Competition の Final portfolio にも選定されていることから、第三者の安全性解析による実績は多岐に及んでいる。ASCON は新たにハッシュモードが追加され、NIST が主催する軽量暗号コンペティションの候補にあがっている [DEMSa]。

ASCON は Initialization、Iteration (データ処理部)、Finalization で異なる繰り返し数を持つ暗号学的置換を利用する。Primary recommendation である ASCON-128 は Initialization および Finalization では繰り返し数が 12 である暗号学的置換、Iteration では繰り返し数が 6 である暗号学的置換が利用されている。Initialization をターゲットとした既知の最良な攻撃手法は Cube 攻撃 (高階差分攻撃) による鍵回復攻撃であり 12 段中 7 段まで、Finalization をターゲットとした既知の最良攻撃手法は差分解読法による偽造攻撃であり 12 段中 4 段まで、Iteration をターゲットとした既知の最良攻撃手法は SAT ソルバーを用いた内部状態回復攻撃あり 8 段中 2 段まで、それぞれ攻

* NTT 社会情報研究所

撃方法が知られている。

1.2 Grain-128AEADv2

認証暗号 Grain-128AEAD[HJM+a] は NIST LWC で初めて提案された方式ではあるが、2004 年から 2008 年にかけて行われたストリーム暗号のコンペティション eSTREAM 提案 Grain[HJM05, HJM07] の系譜を継ぎ、詳細な暗号構造も 2011 年に Symmetric Key Encryption Workshop (SKEW) で提案された Grain-128a[ÅHJM11] を踏襲している。Grain-128AEAD を対象として攻撃論文の数は少ないが、ほぼ等価な暗号である Grain-128a に対する解析論文は多く出版されている。したがって、Grain-128a に対する第三者の安全性解析の実績も包括して考えた場合、Grain-128AEAD も多くの第三者解析の実績を持っていると判断できる。一方で、Grain-128a の解析が直接的に Grain-128AEAD の安全性に影響を与えるわけではないことに注意されたい。例えば、Grain-128a のストリーム暗号モードは高速相関攻撃により解読可能なことが指摘されているが、同様の攻撃手法は観測できるデータの制限のため Grain-128AEAD では動作しない。また、Grain-128AEAD は NIST LWC の Final Round で、Initialization の仕様変更が行われ、現在は Grain-128AEADv2 となっている [HJM+b]。

1.3 TinyJAMBU

認証暗号 TinyJAMBU[WH19] は CAESAR Competition の第三次候補の一つである JAMBU[WH16] の軽量版として NIST LWC で初めて提案された。JAMBU がブロック暗号 AES およびブロック暗号 Simon を利用したブロック暗号利用モードであったのに対して、TinyJAMBU は、内側のブロック暗号に相当する部分も軽量な鍵付き暗号学的置換を一から設計しており、安全性評価の観点では、JAMBU と TinyJAMBU は大きく異なる。NIST LWC Finalist 10 方式の中でも、軽量実装という基準で優れた性能を有している一方、その非常に軽量な構造のため、多くの安全性上の懸念も指摘されている。例えば、TinyJAMBU はブロック暗号をベースとした認証暗号ではあるが、そのブロック暗号は、ブロック暗号としては安全でないことが指摘されている。また、TinyJAMBU は Finalist Round で P1 の段数を 384 段から 640 段に修正したが [WH21]、オリジナルの 384 段は線形解読法で鍵回復攻撃が可能なことが指摘されている。また、関連鍵攻撃において、192 ビット安全な TinyJAMBU-192 および 256 ビット安全な TinyJAMBU-256 に対して、実用的な計算量で偽造攻撃が可能なことも指摘されている。

2 ASCON

CAESAR Competition の Final portfolio の一つでもある ASCON を対象とした解析論文は多い。本稿では ASCON の現在の安全性に関する調査結果を整理する。

2.1 ASCON の仕様

名称	鍵長	ナンス長	タグ長	データブロック長	ラウンド数 a	ラウンド数 b
ASCON-128	128	128	128	64	12	6
ASCON-128a	128	128	128	128	12	8

表 1　ASCON の認証暗号モードのパラメータ

名称	最大出力長	データブロック長	ラウンド数 a	ラウンド数 b
ASCON-Hash	256	64	12	12
ASCON-Hasha	256	64	12	8

表 2　ASCON のハッシュモードのパラメータ

ASCON は認証暗号モードとハッシュモードをサポートしている。

認証暗号モードでは、鍵長が k ビット、レート (データブロックサイズ) 長が r ビット、内部で利用する暗号学的置換のラウンド数が a 回と b 回でパラメータ化された、暗号化関数 $\mathcal{E}_{k,r,a,b}$ と復号関数 $\mathcal{D}_{k,r,a,b}$ で定義される。暗号化関数 $\mathcal{E}_{k,r,a,b}$ は k ビットの秘密鍵 K、128 ビットのナンス N、任意長の認証データ A、任意長の平文 P を入力とし、平文長と同じ長さの暗号文 C および 128 ビットのタグ T を出力する。

$$\mathcal{E}_{k,r,a,b}(K, N, A, P) = (C, T)$$

復号関数 $\mathcal{D}_{k,r,a,b}$ は、秘密鍵 K およびナンス N、暗号文 C およびタグ T を入力とし、認証が正しければ平文 P、認証が間違っている場合はエラー $\perp$ を出力する。

$$\mathcal{D}_{k,r,a,b}(K, N, A, C, T) \in \{P, \perp\}$$

表 1 に ASCON の認証暗号モードのパラメータを示す。著者らは primary recommendation を ASCON-128、secondary recommendation を ASCON-128a としている。

ハッシュモードでは、レート (データブロックサイズ) 長が r ビット、内部で利用する暗号学的置換のラウンド数が a 回と b 回、最大出力長 h ビット ($h = 0$ のときは無制限) でパラメータ化され

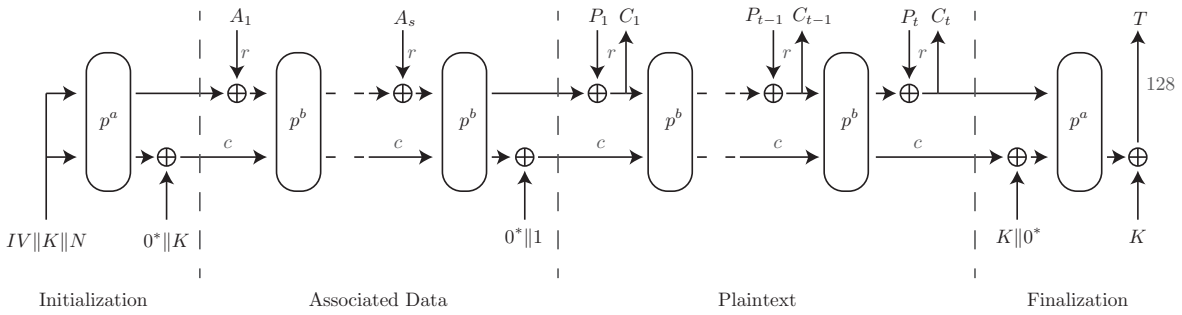

図 1　ASCON の認証暗号モードの暗号化

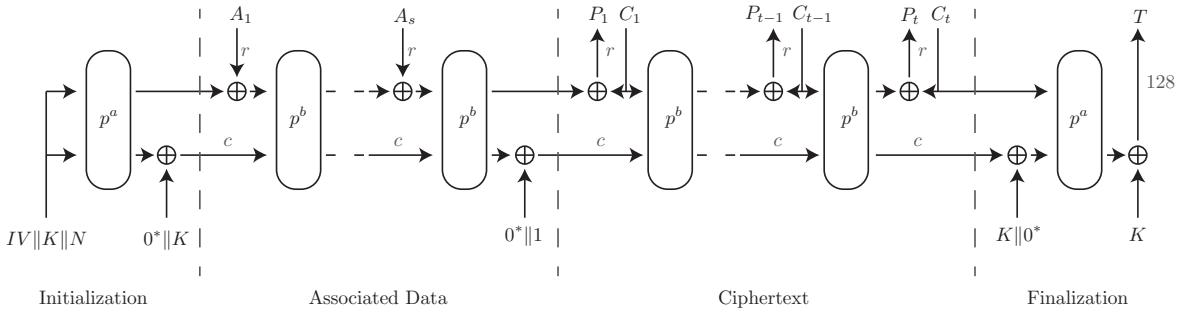

図 2　ASCON の認証暗号モードの復号

た、ハッシュ関数 $\mathcal{X}_{h,r,a,b}$ で定義される。ハッシュ関数 $\mathcal{X}_{h,r,a,b}$ は任意長の平文 M および出力ハッシュ長 $\ell \leq h$ を入力とし、ハッシュ値 H を出力する。

$$\mathcal{H}_{h,r,a,b}(M, \ell) = H.$$

表 2 に ASCON のハッシュモードのパラメータを示す。著者らは primary recommendation を ASCON-Hash、secondary recommendation を ASCON-Hasha としている。

ASCON には上記パラメータのほかに ASCON-Xof $\mathcal{X}_{0,64,12,12}$ や ASCON-Xofa $\mathcal{X}_{0,64,12,8}$、ASCON-80pq $\mathcal{E}, \mathcal{D}_{160,64,12,6}$ も提示されている。

2.1.1　認証暗号モードの詳細

図 1 および 2 に ASCON の認証暗号モードの暗号化および復号を示す。暗号化および復号アルゴリズムは、鍵長・レート長・ラウンド数 a および b から生成される定数 IV・秘密鍵 K・ナンス N をロードし攪拌する Initialization フェーズ、Associated Data を吸収する Associatad Data 処理フェーズ、平文を暗号化・復号する平文・暗号文処理フェーズ、認証タグを出力する Finalization フェーズからなる。

Initialization フェーズ　Initialization フェーズでは、初めに 320 ビットの初期状態を以下のように生成する。

$$IV_{k,r,a,b} \leftarrow k\|r\|a\|b\|0^{160-k} = \begin{cases} \texttt{0x80400c0600000000} & \text{for ASCON-128} \\ \texttt{0x80800c0800000000} & \text{for ASCON-128a} \\ \texttt{0xa0400c06} & \text{for ASCON-80qp} \end{cases}$$

$$S \leftarrow IV_{k,r,a,b}\|K\|N$$
$$S \leftarrow p^a(S) \oplus (0^{320-k}\|K)$$

ここで、k, r, a, b は 8 ビットの値で表現される。

Associated Data 処理フェーズ　Associated Data A を処理するとき、初めに r ビットの定数倍となるように A にパディングを施し、それを s 個のブロックに分割する。パディングは 1 ビット 1 を足し、その後、r ビットの最小の定数倍になるように 0 で埋める。Associated Data が空の場合、パディングは行わず $s = 0$ となる。

$$A_1, \ldots, A_s \leftarrow \begin{cases} r\text{-bit blocks of } A\|1\|0^{r-1-(|A| \bmod r)} & \text{if } |A| > 0 \\ \varnothing & \text{if } |A| = 0 \end{cases}$$

内部状態 S の最初の r ビットを S_r、後ろの c ビット S_c としたとき、各ブロック A_i は以下のように処理される。

$$S \leftarrow p^b((S_r \oplus A_i)\|S_c), \quad 1 \le i \le s$$

A_s の処理後、1 ビットの Domain Separation が排他的論理和される。

$$S \leftarrow S \oplus (0^{319}\|1)$$

平文・暗号文処理フェーズ　平文 P を処理するとき、初めに r ビットの定数倍となるように P にパディングを施し、それを t 個のブロックに分割する。パディングは 1 ビット 1 を足し、その後、r ビットの最小の定数倍になるように 0 で埋める。

$$P_1, \ldots, P_t \leftarrow r\text{-bit blocks of } P\|1\|0^{r-1-(|P| \bmod r)}$$

内部状態 S の最初の r ビットを S_r、後ろの c ビット S_c としたとき、各ブロック P_i は以下のように処理される。

$$C_i \leftarrow S_r \oplus P_i$$
$$S \leftarrow \begin{cases} p^b(C_i\|S_c) & \text{if } 1 \le i < t \\ C_i\|S_c & \text{if } 1 \le i = t \end{cases}$$

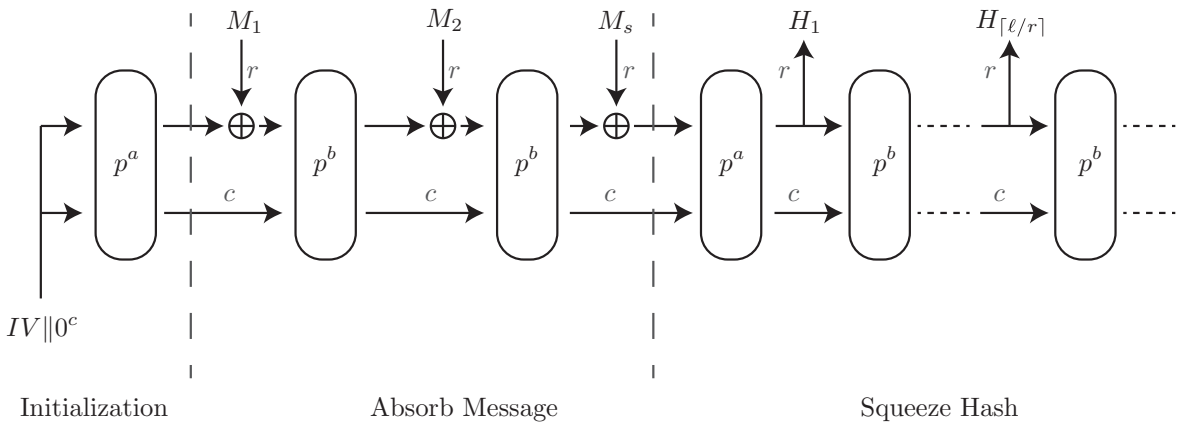

図 3　ASCON のハッシュモードの処理

最後の暗号文ブロック C_t は平文長と同じ長さになるように切り詰められる。

$$\tilde{C}_t \leftarrow \lfloor C_t \rfloor_{|P| \bmod r}$$

復号では、各ブロック C_i は以下のように処理される。

$$P_i \leftarrow S_r \oplus C_i$$
$$S \leftarrow p^b(C_i \| S_c), \quad 1 \le i < t$$

最終ブロックでは、切り詰められた ℓ ビットの暗号文 $\tilde{C}_t$ に対して、以下のように処理する。

$$\tilde{P}_t \leftarrow \lfloor S_r \rfloor_\ell \oplus \tilde{C}_t$$
$$S \leftarrow S_r \oplus (\tilde{P}_t \| 1 \| 0^{r-1-\ell}) \| S_c$$

Finalization フェーズ　Finalization フェーズでは、初めに内部状態に秘密鍵 K を排他的論理和したのち、p^a を適用、最後に下位 128 ビットに秘密鍵 K の下位 128 ビットを排他的論理和した結果をタグ T として出力する。

$$S \leftarrow p^a(S \oplus (0^r \| K \| 0^{c-k}))$$
$$T \leftarrow \lceil S \rceil^{128} \oplus \lceil K \rceil^{128}$$

暗号化アルゴリズムは暗号文 $C_1, C_2, \ldots, C_{t-1}, \tilde{C}_t$ とタグ T を出力する。復号アルゴリズムでは計算されたタグと受け取ったタグが一致する場合、復号した平文 $P_1, P_2, \ldots, P_{t-1}, \tilde{P}_t$ を出力する。

2.1.2　ハッシュモードの詳細

図 3 に ASCON のハッシュモードの処理を示す。固定長を出力するハッシュ関数と、拡張可能な出力関数となる XOF 関数はともに同じアルゴリズムを利用する。

Initialization フェーズ　Initialization フェーズでは、初めに 320 ビットの初期状態を以下のように生成する。

$$IV_{h,r,a,b} \leftarrow 0^8 \| r \| a \| a-b \| h = \begin{cases} \texttt{0x00400c0000000100} & \text{for ASCON-Hash} \\ \texttt{0x00400c0400000100} & \text{for ASCON-Hasha} \\ \texttt{0x00400c0000000000} & \text{for ASCON-Xof} \\ \texttt{0x00400c0400000000} & \text{for ASCON-Xofa} \end{cases}$$

$$S \leftarrow p^a(IV_{h,r,a,b} \| 0^{256})$$

ここで、r, a, $a-b$ は 8 ビットの値で、h は 32 ビットの値でそれぞれ表現される。

Absorb Message フェーズ　Absorb Message フェーズではメッセージを内部状態を更新させながら吸収していく。初めに r ビットの定数倍となるように、認証暗号モードの平文と同様に M にパディングを施す。

$$M_1, \ldots, M_s \leftarrow r\text{-bit blocks of } M \| 1 \| 0^{r-1-(|M| \bmod r)}$$

内部状態 S の最初の r ビットを S_r、後ろの c ビット S_c としたとき、各ブロック M_i を以下のように処理する。

$$S \leftarrow \begin{cases} p^b((S_r \oplus M_i) \| S_c), & \text{if } 1 \le i < s \\ (S_r \oplus M_i) \| S_c, & \text{if } 1 \le i = s \end{cases}$$

Squeezing フェーズ　ハッシュ値を出力する前に、a ラウンドの p^a を適用する。

$$S \leftarrow p^a(S)$$

その後、要求された出力長 ℓ ビットまで、ハッシュ値を以下のように出力する。

$$\begin{aligned} H_i &\leftarrow S_r \\ S &\leftarrow p^b(S) \quad 1 \le i \le t = \lceil \ell / r \rceil \end{aligned}$$

最終ブロック H_t は $\ell \bmod r$ ビット長、$\lfloor H_t \rfloor_{\ell \bmod r}$ に切り詰められ、ハッシュ値 $H_1 \| \cdots \| \tilde{H}_t$ を出力する。

2.1.3 ASCON Permutation の詳細

ASCON の認証暗号モードおよびハッシュモードは主な構成関数として、Permutation p^a および p^b を利用する。Permutation p は SPN 型のラウンド関数であり、定数加算 p_C、非線形部 p_S、線形部 p_L から、以下のように成る。

$$p = p_L \circ p_S \circ p_C$$

p^{12}	p^8	p^6	Constant c_r	p^{12}	p^8	p^6	Constant c_r
0			0x00000000000000f0	6	2	0	0x0000000000000096
1			0x00000000000000e1	7	3	1	0x0000000000000087
2			0x00000000000000d2	8	4	2	0x0000000000000078
3			0x00000000000000c3	9	5	3	0x0000000000000069
4	0		0x00000000000000b4	10	6	4	0x000000000000005a
5	1		0x00000000000000a5	11	7	5	0x000000000000004b

表 3　p^{12}、p^8、p^6 で利用されるラウンド定数

ラウンド関数は 320 ビットの内部状態 S に適用され、この内部状態は 5 個の 64 ビットワード x_i を用いて $S = x_0 \| x_1 \| x_2 \| x_3 \| x_4$ と表現される。

定数加算 p_C　定数加算 p_C は 3 番目の 64 ビットワード x_2 にラウンド数に依存する定数 c_r を排他的論理和する。

$$x_2 \leftarrow x_2 \oplus c_r$$

ここで表 3 に示すラウンド定数 c_r が各ラウンドで利用される。

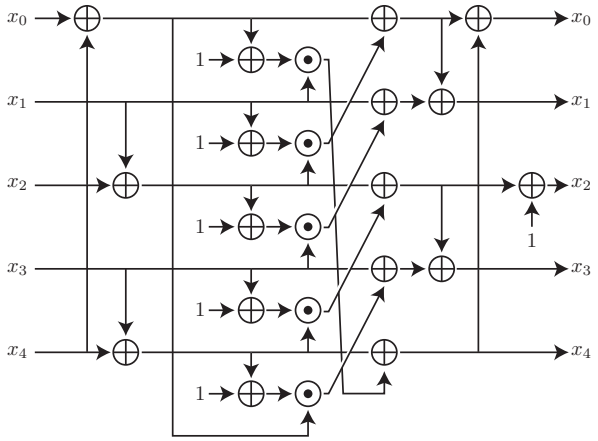

図 4　ASCON の 5 ビット S-box

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	10	11	12	13	14	15	16	17	18	19	1a	1b	1c	1d	1e	1f
$S(x)$	4	b	1f	14	1a	15	9	2	1b	5	8	12	1d	3	6	1c	1e	13	7	e	0	d	11	18	10	c	1	19	16	a	f	17

表 4　ASCON の 5 ビット S-box の真理値表

非線形層 p_S　非線形層 p_S は 5 ビット S-box $S(x)$ を 64 個並列に実行する。詳細には、5 個の 64 ビットワード x_0, x_1, x_2, x_3, x_4 に対して、図 4 に示す演算を実行する。また、対応する真理値表を表 4 に示す。この真理値表では、x_0 が MSB、x_4 が LSB であることに注意されたい。

ASCON の 5 ビット S-box は SHA3[sha15] の χ 関数と類似している。具体的には、ASCON の S-box は、SHA3 の χ 関数の前後にアフィン変換を適用することで構成される。

線形層 p_L　線形層 p_L は各 64 ビットワード x_i 内部を線形演算 Σ_i で以下のように攪拌する。

$$
\begin{aligned}
x_0 &\leftarrow \Sigma_0(x_0) = x_0 \oplus (x_0 \ggg 19) \oplus (x_0 \ggg 28) \\
x_1 &\leftarrow \Sigma_1(x_1) = x_1 \oplus (x_1 \ggg 61) \oplus (x_1 \ggg 39) \\
x_2 &\leftarrow \Sigma_2(x_2) = x_2 \oplus (x_2 \ggg 1) \oplus (x_2 \ggg 6) \\
x_3 &\leftarrow \Sigma_3(x_3) = x_3 \oplus (x_3 \ggg 10) \oplus (x_3 \ggg 17) \\
x_4 &\leftarrow \Sigma_4(x_4) = x_4 \oplus (x_4 \ggg 7) \oplus (x_4 \ggg 41)
\end{aligned}
$$

2.2　差分特性および線形特性について

共通鍵暗号に対する汎用的な解読法として著名な二つに差分解読法 [BS90] および線形解読法 [Mat93] がある。本章で紹介するのは、ASCON Permutation に対する差分特性確率および線形特性確率である。元来、差分確率や線形確率は鍵が存在するブロック暗号に対して定義されるものである。本章で紹介する結果を正しく理解するため、はじめに、ブロック暗号における、差分 (線形) 確率の基本から解説する。

2.2.1　ブロック暗号における差分 (線形) 確率・平均差分 (線形) 確率・差分 (線形) 特性確率の定義

定義 1 (E_K の差分確率 (differential probability))**.** ある固定の秘密鍵 K を用いたブロック長 n ビットであるブロック暗号 E_K の差分確率は以下のように定義される。

$$
DP(\Delta_I \xrightarrow{E_K} \Delta_O) = 2^{-n} \cdot \#\{x \in \{0,1\}^n | E_K(x) \oplus E_K(x \oplus \Delta_I) = \Delta_O\}
$$

定義 2 (E_K の線形確率 (linear squared correlation))**.** ある固定の秘密鍵 K を用いたブロック長 n ビットであるブロック暗号 E_K の線形確率は以下のように定義される。

$$
LP(\Gamma_I \xrightarrow{E_K} \Gamma_O) = \left(2^{-n} \cdot \sum_{x \in \{0,1\}^n} (-1)^{\langle x, \Gamma_I \rangle + \langle E_K(x), \Gamma_O \rangle} \right)^2
$$

鍵は攻撃者にとって未知である。したがって、通常、差分 (線形) 確率の鍵平均が用いられる。

定義 3 (E の平均差分確率 (expected differential probability))**.** 鍵長 κ ビットであるブロック暗号 E の平均差分確率は以下のように定義される。

$$
EDP(\Delta_I \xrightarrow{E} \Delta_O) = 2^{-\kappa} \cdot \sum_{K \in \{0,1\}^\kappa} DP(\Delta_I \xrightarrow{E_K} \Delta_O)
$$

定義 4 (E の平均線形確率 (expected linear squared correlation))**.** 鍵長 κ ビットであるブロック暗号 E の平均線形確率は以下のように定義される。

$$
ELP(\Delta_I \xrightarrow{E} \Delta_O) = 2^{-\kappa} \cdot \sum_{K \in \{0,1\}^\kappa} LP(\Gamma_I \xrightarrow{E_K} \Gamma_O)
$$

平均差分 (線形) 確率はもとより、差分 (線形) 確率ですら、その確率を正確に評価することは技術的に困難である。したがって、共通鍵暗号がラウンド関数の繰り返しによって構成されることに着目し、各ラウンド関数が独立であると仮定した差分 (線形) 特性確率がしばしば利用される。

定義 5 (E の差分特性確率 (differential characteristic probability))**.** ブロック暗号 E はラウンド関数 $F_1, F_2, \ldots, F_R$ の繰り返し構造を取る。ラウンド関数 F_r の差分確率 $DP(\Delta_{r-1} \xrightarrow{F_r} \Delta_r)$ より、ブロック暗号 E の差分特性確率は以下のように定義される。

$$DCP(\Delta_0 \xrightarrow{F_1} \Delta_1 \cdots \xrightarrow{F_R} \Delta_R) = \prod_{r=1}^{R} DP(\Delta_{r-1} \xrightarrow{F_r} \Delta_r)$$

定義 6 (E の線形特性確率 (squared correlation of linear characteristic))**.** ブロック暗号 E はラウンド関数 $F_1, F_2, \ldots, F_R$ の繰り返し構造を取る。ラウンド関数 F_r の線形確率 $LP(\Delta_{r-1} \xrightarrow{F_r} \Delta_r)$ より、ブロック暗号 E の線形特性確率は以下のように定義される。

$$LCP(\Gamma_0 \xrightarrow{F_1} \Gamma_1 \cdots \xrightarrow{F_R} \Gamma_R) = \prod_{r=1}^{R} LP(\Gamma_{r-1} \xrightarrow{F_r} \Gamma_r)$$

2.2.2 暗号学的置換の差分確率および線形確率

定義 3 および定義 4 から明らかなように、ブロック暗号の差分確率や線形確率は鍵の平均値として考えられてきた。また、差分特性確率や線形特性確率が尤もらしく議論できた理由の一つに Markov Cipher 仮定 [LMM91] がある。Markov Cipher とは、ラウンド関数の任意の入出力差分に対して、ラウンド鍵が一様ランダムな場合、ラウンド関数の入力値とは独立に差分確率を計算可能な暗号を言う。確かに、差分特性確率や線形特性確率は、全てのラウンド鍵が一様ランダムに独立に選択され、かつ、一つの差分特性確率や線形特性確率がその他と比べて顕著に高い場合、実際の評価と大きく離れないことが経験的に知られている。

今、解析対象は ASCON Permutation のような暗号学的置換である。鍵はなく、ラウンド鍵もない。したがって、Markov Cipher 仮定を正当化する理由は、実際には存在しない。それにも関わらず、ASCON Permutation だけに限らず、任意の暗号学的置換に対して、各ラウンド関数を独立とみなしたうえで評価する差分特性確率や線形特性確率の結果は多く報告されている。実際、仮定に尤もらしさはないが、差分解読法や線形解読法に対して**おそらく**安全であるという傍証としては十分であり、現状、暗号学的置換に対して差分特性確率や線形特性確率を評価することは無価値である、という結論には至っていない。一方で、特性確率を評価する尤もらしさはブロック暗号のそれと比べると乏しいことに注意したうえで、次節以降の結果を参照されたい。次節で、ASCON Permutation は 6 段で差分特性確率と線形特性確率が 2^{-128} を下回り安全だろう、という結果を紹介するが、この評価の妥当性は乏しい。将来、大きく理解が進展し、何らかの形で結果が覆る可能性は否定できないことに、注意されたい。

2.2.3 ASCON-Permutatio の差分特性および線形特性

差分特性確率および線形特性確率の上界・下界 ASCON Permutation の差分特性確率および

R	差分特性確率の下界 確率	差分特性確率の上界 確率
1	2^{-2}	2^{-2}
2	2^{-8}	2^{-8}
3	2^{-40}	2^{-40}
4	2^{-107}	$\leq 2^{-86}$
5	2^{-190}	$\leq 2^{-100}$
6	2^{-305}	$\leq 2^{-129}$
7		$\leq 2^{-131}$
8		$\leq 2^{-172}$
9		$\leq 2^{-186}$
10		$\leq 2^{-215}$
11		$\leq 2^{-229}$
12		$\leq 2^{-258}$

表 5　ASCON の差分特性確率の上界・下界 (文献 [HMMD22] より)

R	線形特性確率の下界 確率	線形特性確率の上界 確率
1	2^{-2}	2^{-2}
2	2^{-8}	2^{-8}
3	2^{-28}	2^{-28}
4	2^{-98}	$\leq 2^{-88}$
5	2^{-184}	$\leq 2^{-96}$
6		$\leq 2^{-132}$
7		$\leq 2^{-134}$
8		$\leq 2^{-176}$
9		$\leq 2^{-184}$
10		$\leq 2^{-220}$
11		$\leq 2^{-228}$
12		$\leq 2^{-264}$

表 6　ASCON Permutation の線形特性確率の上界・下界 (文献 [HMMD22] より)

線形特性確率の現状は [HMMD22] に整理されており、同論文でまとめられている現在の結果を表 5 および 6 に示す。ASCON Permutation の差分特性確率や線形特性確率は整数計画法ソルバ (MILP) や SAT ソルバ、制約プログラミング (CP) などのソルバを用いる方法 [EME22, MR22] や、専用ツールを用いる方法 [HMMD22] が提案されている。

2^{-128} を上回る差分特性確率を発見できているのは、4 段で、その確率は 2^{-107} である。また、6 段で差分特性確率が 2^{-129} 以下となることが示されている。

2^{-128} を上回る線形特性確率を発見できているのは、4 段で、その確率は 2^{-98} である。また、6 段で線形特性確率が 2^{-132} 以下となることが示されている。

これらの結果は単純な差分解読法や線形解読法では、ASCON の解読が極めて困難なことを示している。さらに、ASCON Permutation の差分・線形特性だけでは ASCON の解読には不十分であり、実際の解読には、攻撃者が制御可能なレート領域のみを利用した差分・線形特性が必要不可欠である。

差分線形特性　差分特性確率や線形特性確率を用いた差分識別攻撃・線形識別攻撃は高々 4 段までしか解読が出来ない。しかしながら、これらの結果から、差分や線形を用いた統計学的な攻撃が ASCON には全く適用出来ないと結論付けることは出来ない。差分解読法および線形解読法から派生した一つの拡張攻撃法に、差分解読法と線形解読法を混在させる差分線形攻撃がある [LH94]。差分線形攻撃は、入力差分 Δ_I および出力マスク Γ_O に対して、以下の確率がランダムよりも優位

	Data	Time	Method	# Keys	引用
Single-key attack	$2^{77.2}$†	2^{104}	Conditional cube	2^{128}	[LDW17]
	2^{64}	2^{123}	Cube	2^{128}	[RHSS21]
Weak-key attack	$2^{77.2}$†	2^{77}	Conditional cube	2^{117}	[LDW17]
	2^{64}	2^{97}	Cube	$2^{116.34}$	[RS21]
	2^{63}	$2^{115.2}$	Cube	$2^{116.34}$	[RS21]

† ASCON は同一の鍵で処理可能なデータ量を 2^{64} 以下と制限している。したがって、文献 [LDW17] の攻撃はデータ量制限を超過している。

表 7　7 段 ASCON に対する鍵回復 Cube 攻撃のまとめ

に高いことを利用した解読法である。

$$DLP(\Delta_I \xrightarrow{E_K} \Gamma_O) = \left(2^{-n} \cdot \sum_{x \in \{0,1\}^n} (-1)^{\langle \Gamma_O, E_K(x) \rangle + \langle \Gamma_O, E_K(x \oplus \Delta_I) \rangle} \right)^2$$

ASCON に対する差分線形攻撃は、設計者自らによる攻撃論文で導入され [DEMS15]、モードも含めて ASCON-128 の Initialization が 5 段に縮退された方式を検証可能な形で解読に成功している。このことから、ASCON に対しては、単純な差分解読法や線形解読法よりも、差分線形攻撃の方が強力となっている。一般に、差分線形攻撃は、特定の段数まで非常に高い差分確率および線形確率を持つが、ある特性の段数を超えると著しく確率が低下するような暗号に対して、通常の差分解読法や線形解読法よりも強力な攻撃法となることが知られている。確かに表 5 や 6 に整理された結果を見ると、ASCON Permutation は 2 段目から 3 段目に確率の大きな低下が観測されており、差分線形攻撃が単純な差分解読法や線形解読法よりも優れる条件が整っている。

設計者らによって発見された 5 段に対する差分線形攻撃は、発見的なものであり、その理論的な理解は [DEMS15] では示されていない。その理論的な解析は Bar-On らによって [BDKW19] で示された。また、代数的な性質を利用した改良手法は Liu らによって [LLL21] で示されている。しかしながら、差分線形攻撃を用いて解読可能な段数は 5 段から改良はされていない。

2.3　Cube 攻撃に対する耐性について

現状、ASCON に対する最良解読法は Cube 攻撃[1)]によって得られる。ASCON に対する Cube 攻撃は、Division Property に基づく自動探索ツールを利用したものや、代数的な構造を詳細に解析することによる発見的な手法の大きく二通りが議論されている。

Division Property は暗号の仕様が与えられた状態で Integral Distinguisher を探索する汎用的なツールであり、藤堂が 2015 年に導入した [Tod15]。Division Property は代数次数と密接な関係があり [BC16]、効率的な代数次数の上界評価ツールとみなすことで Cube 攻撃への応用が可能

1)　Cube 攻撃 [DS09] と同じ原理の解読法には、Integral 攻撃 [KW02]、高階差分攻撃 [Lai94]、Square 攻撃 [DKR97] など別名が多数ある。ASCON に対する同種の解読論文は Cube 攻撃の名称を取ることが多いため Cube 攻撃で統一する。

である [TIHM17]。ASCON に対して Division Property を用いた解析は数多く取り組まれている [Tod15, GRW16, YLW+19, GD21, RHSS21]。現在、任意の鍵を持つ ASCON に対する最良解読法は、この手法により得られており、2^{64} 個の選択ナンスを用いて 2^{123} の計算量で秘密鍵の回復に成功している [RHSS21]。Nonce respecting なシナリオでは、攻撃者は同一の Nonce に対して複数個の平文をクエリすることは出来ない。したがって、Cube 攻撃の対象は、ASCON の Initialization であり、攻撃対象は p^b ではなく p^a であることに注意されたい。p^a は ASCON-128、ASCON-128a、ともに 12 段を用いるため、12 段中 7 段が攻撃可能であることから、5 段分の安全性マージンを有していることが分かる。

Division Property のような汎用ツールを利用することなく、ASCON の詳細な代数的性質を解析することによって得られた Cube 攻撃も多数、提案されている [LZWW17, LDW17, RS21]。文献 [RS21] では、特定の弱鍵に対してのみ成立する識別攻撃や鍵回復攻撃が示された。攻撃可能段数は 7 段で、使われた秘密鍵が $2^{116.34}$ 個の弱鍵から選択された場合、2^{64} 個の選択ナンスを用いて 2^{97} の計算量で秘密鍵の回復に成功している[2)]。表 7 に 7 段 ASCON に対する Cube 攻撃に基づいた鍵回復攻撃の結果を整理する。

2.4 ASCON Permutation に対する考察

ASCON は Sponge 構造をベースとした認証暗号・ハッシュ関数である。Sponge 構造は、内部置換としてランダム置換モデルを用いることで、Indifferentiability の証明が可能である [BDPA08, JLM14]。実際には、ASCON Permutation と呼ばれる固定の暗号学的置換のみを用いるため、この Indifferentiability の証明は ASCON そのものの安全性に寄与するものではない。一方で、Indifferentiability の証明は、ASCON Permutation そのものに、暗号学的な脆弱性と思える性質がなければ、ASCON の認証暗号モード・ハッシュモードは安全であろう、という含意となる。上記の議論を受けて、Sponge 構造で用いられる暗号学的置換がランダム置換と**意味のある識別**[3)]が出来るか、を主題とする解析論文は多く提案されている [BC10, BCC11, FLN+21]。

Zero-Sum Distinguisher と Zero-Sum Partition　Zero-sum distinguisher (zero-sum partition)[BC10] がしばしば、この文脈で議論される。Zero-sum distinguisher とは、攻撃対象 P に対して、$\sum_{x\in S} x = 0$ かつ $\sum_{x\in S} P(x) = 0$ となる集合 S を見つけることである。集合 S のサイズが十分に小さいとき、このような S を発見することは確かに困難である。一方で、集合 S のサイズが置換 P のブロック長程度である場合、このような集合を発見することは Gauss の消去法を用いることで容易である。したがって、Zero-sum distinguisher が、Generic algorithm よりも優れるためには、かなりの少量の S で Zero-sum distinguisher を構成する必要がある。文献 [GPT21] によると、

2)　Division Property、正確には、その正確性を高めた Parity Set [BC16] は次数評価において、tight な評価が可能であることが指摘されている [HLM+21, HLLT20]。したがって、同様の結果を Division Property を用いて再評価することは、おそらく可能である。しかしながら、元来、汎用解析ツールである Division Property は、弱鍵のようなアドホックな解析には不向きと言える。

3)　理論的には、固定の暗号学的置換はランダム置換と 1 クエリで識別可能である。単純に、適当に選択した入力をクエリし、クエリ結果が固定の暗号学的置換の出力と一致するかを確認すればよい。ここで、意味のある識別とは、このような自明な識別ではなく、差分特性や線形特性など、暗号学的な解析を伴う識別であることに注意されたい。

Generic algorithm よりも高速に Zero-sum distinguisher を構成できるのは 6 段までで、その計算量は 2^{10} となっている。

Zero-sum distinguisher は、しばしば、Zero-sum partition という拡張において、その是非が議論となる。Zero-sum partition とは、上記の集合 S を $2^n/|S|$ 個、独立に構成することである。Zero-sum distinguisher を代数次数評価で構成した場合、Zero-sum partition への拡張が容易となる。はじめに $P = P_2 \circ P_1$ と分割し、その後 P_2 の代数次数と P_1^{-1} の代数次数を評価する。結果的に $d = \max(\deg(P_2), \deg(P_1^{-1}))$ において、$d+1$ 階差分を取るような入力集合 S を用意し、$\{P_1^{-1}(x)|x \in S\}$ を入力集合とすることで、2^{d+1} の計算量で Zero-sum distinguisher が構成出来る。$d+1$ 階差分を取るような入力集合 S と完全に disjoint である $d+1$ 階差分を取るような別の集合 S' を構成することも容易であり、これを繰り返すことで、Zero-sum partition が構成できる。確かに、Zero-sum partition を Gauss の消去法などを用いて構成することは容易ではなく、Zero-sum partition が意味のある識別と言えるかどうかは要議論と言える [GPT21]。Zero-sum partition は設計者らの解析ですでに示されており [DEMS15]、12 段の Full round に対して構成可能である。同論文で、設計者は、仮に Zero-sum partition があったとしても、ASCON の認証暗号モードの脆弱性にはつながらないため、問題にはならないと主張している。

Limited Birthday Distinguisher Limited Birthday Distinguisher[GP10] とは、制限下で、Birthday problem を解く問題である。攻撃対象 P と集合 $D_{in}, D_{out} \subseteq \mathbb{F}_2^n$ において、$x \oplus x' \in D_{in}$ かつ $P(x) \oplus P(x') \in D_{out}$ となるペア (x, x') を出力する問題である。Zero-sum partition との違いとして、Limited Birthday は Generic に構成可能な最良手法が以下のようになることが分かっている。

$$\max\left\{\min\left\{\sqrt{\frac{2^{n+1}}{|D_{in}|}}, \sqrt{\frac{2^{n+1}}{|D_{out}|}}\right\}, \frac{2^{n+1}}{|D_{in}||D_{out}|}\right\}$$

文献 [GPT21] で、7 段 ASCON に対して Generic には $2^{37.14}$ 以上の計算量を要する Limited Birthday Problem を 2^{34} の計算量で解く手法が示されている。

2.5 ハッシュモードに対する安全性評価

認証暗号モードとは異なり、ASCON のハッシュモードに対する解析論文は多いとは言えない。設計者らは 2019 年に Preliminary Analysis としてハッシュモードに対する解析を公開した [DEMS19]。初めに代数的な性質を利用した攻撃として、一部を推測したのちに線型方程式を解くという手法で、2 段まで、2^{39} の計算量で Preimage 攻撃が可能なことが示された。また、Permutation の代数次数が小さい場合、より少ないビット操作で原像回復する手法が知られており [Ber10]、この手法を用いることで、6 段に対して $2^{63.3}$ の計算量で Preimage 攻撃が可能と示されている。

衝突攻撃に関しては 2 段まで、64 ビット出力の衝突は 2^{15} の計算量、256 ビット出力の衝突は 2^{125} の計算量で可能なことが示されている [ZDW19]。これらの衝突攻撃は差分特性を利用して発見されている。

3 Grain-128AEADv2

Grain-128AEADv2 はストリーム暗号 Grain の系譜を継ぐ認証暗号である [HJM+b]。ストリーム暗号 Grain は、2004 年から 2008 年にかけて行われたストリーム暗号の Compeitition『eSTREAM』で提案された [HJM05]。Grain は提案まもなく、線形識別攻撃や相関攻撃などで解読され [BGM06]、このオリジナル版は Grain v0 と呼ばれる。Grain v0 の脆弱性に対策を施す形で提案された Grain v1 は、eSTREAM の Final Portfolio に選定されている [HJM07]。Grain の系譜を継ぐストリーム暗号として、2006 年に 128 ビット安全を主張する Grain-128 が提案された [HJMM06]。しかしながら、2011 年に、Dynamic Cube Attack により解読された [DS11]。この脆弱性を補完する形で Grain-128a が 2011 年に提案された [ÅHJM11]。Grain-128a は 128 ビット安全を確保するとともに、ストリーム暗号モードの他に、新たに認証暗号モードに対応した。2018 年に、Grain v1、Grain-128、および Grain-128a(ストリーム暗号モードのみ) は高速相関攻撃によって、解読された [TIM+18]。Grain-128AEAD は Grain-128a を基に、2018 年に提案された高速相関攻撃に対策を施し提案され [HJM+a]、NIST LWC 期間中に、既知内部状態の仮定から秘密鍵回復攻撃を行う攻撃 [CT21] を受けて、さらに Version 2、Grain-128AEAD v2 に仕様が更新された [HJM+b]。

Grain-128AEAD を対象とした安全性評価論文は多くない。したがって、本報告書では、Grain の系譜を継ぐ暗号方式の脅威となってきた、Cube 攻撃および高速相関攻撃を紹介し、これらの脅威に対して、Grain-128AEADv2 がどのように安全性を確保しているかを整理する。

3.1 Grain-128AEADv2 の仕様

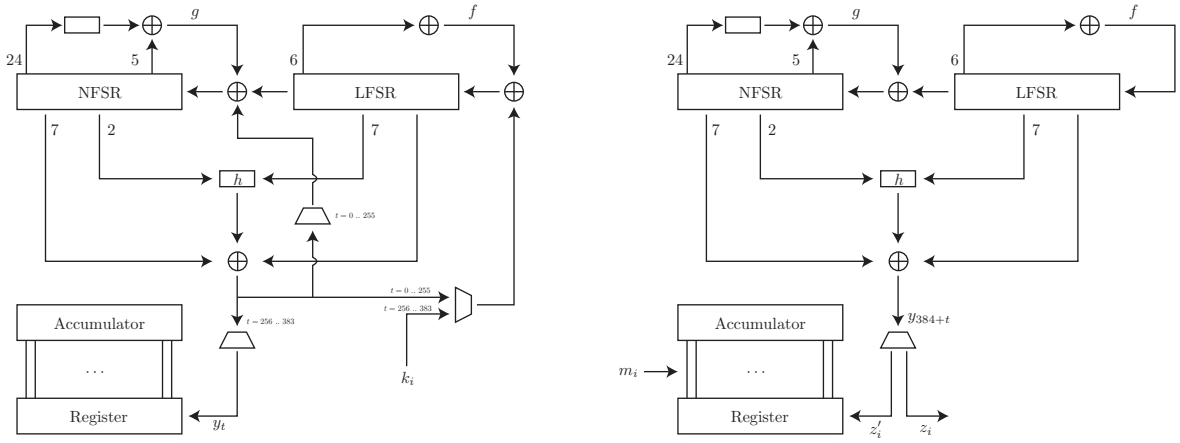

図 5　Grain-128AEADv2 の仕様。左図は Initialization、右図は平文処理およびキーストリーム出力。

図 5 に Grain-128AEADv2 の仕様を示す。Grain-128AEADv2 は LFSR 型ストリーム暗号ベースの認証暗号であり、128 ビットの線形フィードバックシフトレジスタ (LFSR) と 128 ビットの非

線形フィードバックシフトレジスタ (NFSR)、認証タグ生成のための 64 ビット Accumulator と 64 ビットレジスタから成る。

時刻 t の LFSR の状態を $S_t = [s_0^t, s_1^t, \ldots, s_{127}^t]$、時刻 t の NFSR の状態を $B_t = [b_0^t, b_1^t, \ldots, b_{127}^t]$ とする。このとき LFSR の帰還関数は

$$f(x) = 1 + x^{32} + x^{47} + x^{58} + x^{90} + x^{121} + x^{128}$$

すなわち

$$s_{127}^{t+1} = \mathcal{L}(S_t) = s_0^t + s_7^t + s_{38}^t + s_{70}^t + s_{81}^t + s_{96}^t$$

となる。一方で、NFSR の帰還関数は

$$\begin{aligned} g(x) =& 1 + x^{32} + x^{37} + x^{72} + x^{102} + x^{128} + x^{44}x^{60} + x^{61}x^{125} \\ &+ x^{63}x^{67} + x^{69}x^{101} + x^{80}x^{88} + x^{110}x^{111} + x^{115}x^{117} \\ &+ x^{46}x^{50}x^{58} + x^{103}x^{104}x^{106} + x^{33}x^{35}x^{36}x^{40} \end{aligned}$$

LFSR からの出力も足し、

$$\begin{aligned} b_{127}^{t+1} &= s_0^t + \mathcal{F}(B_t) \\ &= s_0^t + b_0^t + b_{26}^t + b_{56}^t + b_{91}^t + b_{96}^t + b_3^t b_{67}^t + b_{11}^t b_{13}^t \\ &+ b_{17}^t b_{18}^t + b_{27}^t b_{59}^t + b_{40}^t b_{48}^t + b_{61}^t b_{65}^t + b_{68}^t b_{84}^t \\ &+ b_{22}^t b_{24}^t b_{25}^t + b_{70}^t b_{78}^t b_{82}^t + b_{88}^t b_{92}^t b_{93}^t b_{95}^t \end{aligned}$$

となる。暗号化および MAC で利用される pre-output の出力は関数 h を用いて、

$$y_t = h(x) + s_{93}^t + b_2^t + b_{15}^t + b_{36}^t + b_{45}^t + b_{64}^t + b_{73}^t + b_{89}^t$$

であらわされ、ここで h の出力は、入力 $x_0, \ldots, x_8$ として $b_{12}^t, s_8^t, s_{13}^t, s_{20}^t, b_{95}^t, s_{42}^t, s_{60}^t, s_{79}^t, s_{94}^t$ を用いて、以下の非線形ブール関数の出力である。

$$h(x) = x_0x_1 + x_2x_3 + x_4x_5 + x_6x_7 + x_0x_4x_8$$

Initialization フェーズ Pre-output の出力の前に、LFSR、NFSR、Accumulator、レジスタ、すべてを鍵とナンスを用いて初期化する。初めに 128 ビットの鍵を NFSR にロードする。次に 96 ビットのナンスを LFSR の先頭 96 ビットにロードし、残りの 32 ビットのうち先頭 31 ビットには 1 ($s_i^0 = 1, (96 \leq i \leq 126)$ を、最終ビットには 0 ($s_{127}^0 = 0$) を埋める。鍵とナンスを LFSR および NFSR にロード後、320 段、LFSR および NFSR を更新する。Initialization フェーズでは、y_t を pre-output の出力とすると、この出力を LFSR および NFSR の更新値に加算する。

$$\begin{aligned} s_{127}^{t+1} &= \mathcal{L}(S_t) + y_t, & 0 \leq t \leq 319 \\ b_{127}^{t+1} &= s_0^t + \mathcal{F}(B_t) + y_t, & 0 \leq t \leq 319 \end{aligned}$$

320 段かけて更新後、64 段かけて、鍵を再度、以下のように加算する。

$$s_{127}^{t+1} = \mathcal{L}(S_t) + y_t + k_{t-256}, \qquad 320 \le t \le 383$$
$$b_{127}^{t+1} = s_0^t + \mathcal{F}(B_t) + y_t + k_{t-320}, \qquad 320 \le t \le 383$$

さらに 128 段かけて、認証タグ生成のために 64 ビット Accumulator の初期値 $A_0 = [a_0^0, a_1^0, \ldots, a_{63}^0]$ およびレジスタの初期値 $R_0 = [r_0^0, r_1^0, \ldots, r_{63}^0]$ を以下のように生成する。

$$a_j^0 = y_{384+j} \qquad 0 \le j \le 63$$
$$r_j^0 = y_{448+j} \qquad 0 \le j \le 63$$
$$s_{127}^{t+1} = \mathcal{L}(S_t) \qquad 384 \le t \le 511$$
$$b_{127}^{t+1} = s_0^t + \mathcal{F}(B_t) \qquad 384 \le t \le 511$$

以上の手順の全てを合わせると、Grain-128AEADv2 の Initialization は 512 段を要する。

Associatad Data、平文・暗号文処理フェーズ Grain-128AEADv2 は L ビットのメッセージ $\mathbf{m} = m_0, m_1, \ldots, m_{L-1}$ を処理する際に、$\mathbf{m}$ と $\mathbf{m}\|0$ を区別するため、パディングとして $m_L = 1$ を付加する。

Initialization 後、pre-output の出力の偶数番目のビットは暗号化のキーストリームとして利用し、奇数番目のビットは認証に利用する。

$$z_i = y_{512+2i}$$
$$z_i' = y_{512+2i+1}$$

i 番目の暗号化は

$$c_i = m_i \oplus z_i \qquad 0 \le i \le L$$

で処理する。認証では、Accumulator を以下のように更新する。

$$a_j^{i+1} = a_j^i + m_i r_j^i \qquad 0 \le j \le 63,\ 0 \le i \le L$$

さらに z_i' を用いてレジスタを以下のように更新する。

$$r_{63}^{i+1} = z_i'$$
$$r_j^{i+1} = r_{j+1}^i \qquad 0 \le j \le 62$$

L ビットのメッセージを処理したのち、Accumulator の値を認証タグとして利用する。

認証に含まれるが暗号化の対象とならない Associated Data には、AEAD mask $d = d_0, d_1, \ldots, d_{L-1}$ を利用する。i 番目の平文を暗号化する際に用いるキーストリームを z_i としたとき、

$$c_i = m_i \oplus z_i \cdot d_i \qquad 0 \le i \le L$$

とすることで、暗号化されないビットを指定する。すなわち、事前に AEAD mask を用いることで任意のビット位置を暗号化の対象から除外することが可能である。

NIST LWC では、Associated Data は平文の前に付加されるものであり、また、処理するメッセージもバイト単位となる。したがって、平文のエンコードとして

$$\mathrm{Encode}(adlen)\|ad\|m\|\texttt{0x80}$$

を利用する。ここで Encode() = y は、y の先頭が 0 の場合、残りの 7 ビットは associated data のバイト長 (最大 127 バイトまで) となる。y の先頭が 1 の場合、残りの 7 ビットは、その後、associated data のバイト長を表記するために使用されるバイトの量を記載する。これらのエンコードされたメッセージに対して、Encode($adlen$)$\|ad$ に対して $d_i = 0$、それ以降は $d_i = 1$ となる AEAD mask を利用する。

3.2 高速相関攻撃

Grain 型ストリーム暗号に対して、強力な解読法の一つが高速相関攻撃である。Grain v0 は相関攻撃により解読され [BGM06]、Grain v1、Grain-128、Grain-128a のストリーム暗号モードもまた、有限体の可換性を用いた改良高速相関攻撃によって解読されている [TIM+18]。Grain-128a は Grain-128AEADv2 と同一の LFSR および NFSR を利用する。ここでは、Grain-128a のストリーム暗号モードの高速相関攻撃を解説したのちに、なぜ、その攻撃が Grain-128AEADv2 には適用できないかを示す。

3.2.1 高速相関攻撃の基本

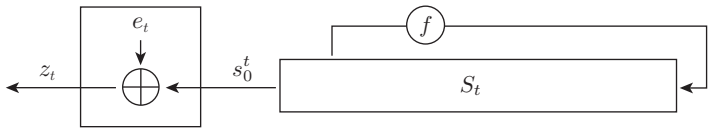

図 6　LFSR 型ストリーム暗号

図 6 に LFSR 型ストリーム暗号の概念図を示す。時刻 t における LFSR の内部状態 S_t から s_0^t が出力され、その出力に非線形で生成されるノイズ e_t が加算されたものが時刻 t のキーストリームとなる[4)]。仮にノイズ e_t に偏りがあり、$e_t = 1$ となる確率が p、相関が $c = 1 - 2p$ だったと仮定しよう。このとき、N ビットのキーストリームを用いて、以下を評価する。

$$\sum_{t=0}^{N-1}(-1)^{s_0^t \oplus z_t} = \sum_{t=0}^{N-1}(-1)^{e_t}$$

上記の値は、およそ正規分布 $\mathcal{N}(Nc, N)$ に従うが、真にランダムならば $\mathcal{N}(0, N)$ に従う。すなわち、$N = O(1/c^2)$ を用いることで、この偏りを観測、識別が可能である。

4)　実際の Grain は、LFSR の一部のビットに対して非線形なフィルタ関数を通したうえでキーストリームを生成することから、ここまで簡略化したモデルでは表現できない。

Grain をはじめ、現代のストリーム暗号において、e_t そのものに偏りがあるような脆弱なケースは稀である。そのような場合、解析をより一般化させ、以下のようにノイズを定義する。

$$e'_t = \bigoplus_{i \in \mathbb{T}_s} \langle S_{t+i}, \Gamma_i \rangle \oplus \bigoplus_{i \in \mathbb{T}_z} z_{t+i}$$

ここで、s_0^t の代わりに、$\bigoplus_{i \in \mathbb{T}_s} \langle S_{t+i}, \Gamma_i \rangle$、すなわち複数時刻の内部状態に対して特定の線形マスクを適用した結果の総和に、複数時刻のキーストリームの総和を加算する形で再定義する。このようにすることで、Grain のように、複雑なフィルタ関数を用いるケースでも、高速相関攻撃が適用可能になる。

高速相関攻撃の目的は LFSR の初期状態 S_0 を回復することである。そこで、$t+i$ ステップ後の内部状態は S_0 と帰還多項式 f を 2 進 $n \times n$ 行列で表現した F を用いて、

$$S_{t+i} = S_0 \times F^{t+i}$$

と表せることを利用する。このとき

$$e'_t = \left\langle S_0, \left(\bigoplus_{i \in \mathbb{T}_s} (\Gamma_i \times^{\mathrm{T}} F^i) \right) \times^{\mathrm{T}} F^t \right\rangle \oplus \bigoplus_{i \in \mathbb{T}_z} z_{t+i}$$

となり、$\Gamma = \bigoplus_{i \in \mathbb{T}_s} (\Gamma_i \times^{\mathrm{T}} F^i)$ とすることで

$$e'_t = \left\langle S_0, \Gamma \times^{\mathrm{T}} F^t \right\rangle \oplus \bigoplus_{i \in \mathbb{T}_z} z_{t+i} \tag{1}$$

のように変形できる。S_0 を推測したのちに $t=0$ から $t=N-1$ まで用いることで、$\sum_{t=0}^{N-1} (-1)^{e'_t}$ を計算し、計算結果が正規分布 $\mathcal{N}(Nc, N)$ に従うかどうかを評価する。

仮に $t=N-1$ まで用いて識別が可能だとしても、単純なアルゴリズムによる評価は $O(N2^n)$ の計算量を要する。Chose らは、鍵の推測と評価は Walsh-Hadamard 変換と等価であることを示し [CJM02]、高速 Walsh-Hadamard 変換 (FWHT) を用いることで、計算量を $O(N + n2^n)$ まで削減可能なことを示した。たとえ FWHT を用いたとしても、計算量は 2^n を超えることに注意されたい。すなわち、LFSR のサイズが少なくとも鍵サイズ以上であれば、上記の攻撃法は全数探索攻撃よりも非効率となる。Grain の LFSR のサイズは鍵長と同じであるため、Grain への高速相関攻撃の適用には、LFSR の全サイズの推測を回避したショートカット攻撃が必要不可欠である。例えば、$\Gamma \times^{\mathrm{T}} F^t$ のうち、(一般性を損なわず) 上位 b ビットが 0 のもののみを利用して高速相関攻撃を行う方法や、generalized birthday problem を利用する方法などがある [BGM06]。

3.2.2 有限体の可換性を利用した高速相関攻撃

藤堂らが CRYPTO2018 で示した改良高速相関攻撃は、LFSR の帰還多項式を有限体上の乗算とみなすことで、線形近似と初期状態推測には可換性があることを見出し、その性質を用いて LFSR の全サイズの推測を回避した [TIM+18]。結果、Grain v1, Grain-128, Grain-128a(ストリーム暗号モード) が 2^{128} 未満の計算量で解読可能なことを示した。

式 (1) において、$\langle S_0, \Gamma \times^{\mathrm{T}} F^t \rangle$ は、原始多項式が LFSR の帰還多項式である有限体上の乗算で計算可能である。α を有限体の原始元とすると α^t は ${}^{\mathrm{T}}F^t$ と同値となり。線形マスク $\Gamma \in \{0,1\}^n$ は自然な方法で $\gamma \in GF(2^n)$ と変換する。このとき $\gamma\alpha^t \in GF(2^n)$ と $\Gamma \times^{\mathrm{T}} F^t$ は同値となる。有限体の乗算は可換である。すなわち、$\gamma\alpha^t = \alpha^t\gamma$ であり、$\alpha^t\gamma$ と同値な

$$A_t \times^{\mathrm{T}} M_\gamma$$

が定義可能である。具体的には A_t とは ${}^{\mathrm{T}}F^t$ の第一行ベクトル、M_γ は 2 進 $n \times n$ 行列、その i 番目の行ベクトルは $\gamma\alpha^{i-1}$ の自然な変換となる。この可換性を利用することで、

$$\langle S_0 \times F^t, \Gamma \rangle = \langle S_0, \Gamma \times^{\mathrm{T}} F^t \rangle = \langle S_0, A_t \times^{\mathrm{T}} M_\gamma \rangle = \langle S_0 \times M_\gamma, A_t \rangle$$

が得られる。

今、高い偏りを持つ複数個の線形近似が m 個あったと仮定する。上記の可換性による解析は、線形マスクに A_t を用いた場合、m 個の推測全てで高い偏りが観測されることを意味する。すなわち、偏る線形近似の数が多ければ多いほど、偏る解の数が増加する。この性質を利用して、$m \times 2^{-b} \gg 1$ のとき、攻撃者は (一般性を損なわず) 上位 b ビットの推測を取りやめたとしても、高い確率で偏る解を得ることが可能である。偏る解を得たあとは、m 個の線形マスクの逆行列 M_γ^{-1} を乗ずることで、LFSR の初期状態の回復が可能となる。

Grain-128a の場合、

$$\mathbb{T}_z = \{0, 26, 56, 91, 96, 128\} \tag{2}$$

において相関が $\pm 2^{-54.2381}$ となる線形マスクが $49152 \times 64 \times 32 \approx 2^{26.58}$ 個ある [TIM+18]。これらの結果を用いることで、$2^{113.8}$ ビットのキーストリームおよび計算量 $2^{115.4}$ を用いて LFSR の初期状態が回復可能である。

3.2.3 Grain-128AEADv2 の高速相関攻撃対策

NIST LWC に提案された Grain-128AEAD (Grain-128AEADv2) は、高速相関攻撃が脅威にならないように以下の対策を施している。

1. 有限体の可換性を利用した高速相関攻撃は Grain-128a のストリーム暗号モードには適用可能だが、認証暗号モードには適用不可能である。その理由は式 (2) で示されたキーストリームのタップ位置に起因する。$0, 26, 56, 96, 128$ は偶数番目のキーストリームを利用するが、91 という奇数番目のキーストリームも必ず利用しなければならない。Grain-128a のストリーム暗号モードは pre-output function の出力全体をキーストリームとして活用するため、偶数番目のキーストリームも奇数番目のキーストリームも、(既知平文攻撃の仮定で) 観測可能である。しかしながら、Grain-128a の認証暗号モード (Grain-128AEAD も同様) は、偶数番目はキーストリームとして出力したとしても、奇数番目は MAC 計算のためレジスタやアキュムレータで利用される。したがって、自然な仮定では、攻撃者はこれらのビットを観測できない。

Grain-128a の認証暗号モードを攻撃するためには、利用するキーストリーム位置が偶数番目のみ、もしくは奇数番目のみで、偏りを発見する必要がある。しかしながら、その条件下では、現状、高い偏りを持つ線形マスクは発見されていない。

2. Grain-128AEADv2 は同一の鍵と IV(ナンス) から生成できるキーストリーム量に制限がある。これは NIST LWC の設計要件を反映したものであり、同一の鍵と IV(ナンス) から生成できるキーストリームの量を 2^{80} までに制限している。2^{80} の制限下では、仮に pre-output function の出力全てが観測可能と仮定したとしても、LFSR の初期状態を回復するには不十分なデータ量しか収集できない。
3. Grain-128AEADv2 は Initialization の途中に、秘密鍵を再ロードするよう Initialization を修正している。そのため、仮に LFSR の Initialization 後の初期状態を回復できたとしても、そこから秘密鍵を回復することは非自明である。

上記の理由から、Grain-128AEADv2 は、現在の最新の高速相関攻撃に対しても安全性マージンを有していることが分かる。

3.3 Cube 攻撃

Grain 型ストリーム暗号に対する暗号解読として多くの結果が報告されているもう一つの手法は Cube 攻撃 [DS09] である。Cube 攻撃の攻撃対象は、ストリーム暗号における Initialization である。鍵を $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$、IV を $\boldsymbol{v} = (v_1, v_2, \ldots, v_m)$ としたとき、Initialization を経て 1 ビット目のキーストリームをブール関数 $f(\boldsymbol{x}, \boldsymbol{v})$ と表現する。このとき、ブール関数 f は公開関数であり、以下のように分解できると仮定する。

$$f(\boldsymbol{x}, \boldsymbol{v}) = t_I \cdot p(\boldsymbol{x}, \boldsymbol{v}) + q(\boldsymbol{x}, \boldsymbol{v})$$

ここで $t_I = v_{i_1} \cdots v_{i_{|I|}}$ であり、p は $\{v_{i_1}, v_{i_2}, \ldots, v_{i_{|I|}}\}$ とは独立なブール関数、q は $\{v_{i_1}, v_{i_2}, \ldots, v_{i_{|I|}}\}$ から少なくとも一つの項が利用されない単項式の和からなるブール関数である。通常、ブール関数 q は非常に複雑な関数となるが、適切に選択された t_I において、ブール関数 p は単純になりえる。そして、このブール関数 p を Cube 攻撃では Superpoly と呼ぶ。Superpoly の出力は index を active にして残りの IV を 0 とした $2^{|I|}$ 個の IV の集合 C_I(これを Cube と呼ぶ) を用いて、以下のように計算できる。

$$\bigoplus_{\boldsymbol{v} \in C_I} f(\boldsymbol{x}, \boldsymbol{v}) = \bigoplus_{\boldsymbol{v} \in C_I} t_I \cdot p(\boldsymbol{x}, \boldsymbol{v}) + q(\boldsymbol{x}, \boldsymbol{v}) = p(\boldsymbol{x}, \boldsymbol{0})$$

したがって、攻撃者が Superpoly の多項式をすでに回復している場合、Cube の各キーストリームの総和を得ることで、秘密鍵に関する多項式を得ることができる。この多項式から、鍵に関する(高々)1 ビットの情報を回復できる。

128 ビット安全を主張した最初の Grain 型ストリーム暗号、Grain-128 は Cube 攻撃を拡張した Dynamic Cube 攻撃により解読された [DS11, DGP+11]。Dynamic Cube 攻撃は、鍵の推測に依存しながら Cube を変えていくことで、鍵の推測が正しければ多項式は単純化され、Cube の各キー

Step	Grain-128AEAD	Grain-128AEADv2
空回し	0 ∼ 255	0 ∼ 319
鍵再ロード	256 ∼ 383	320 ∼ 383
Register Accumulator 初期化	256 ∼ 383	384 ∼ 511

表 8　Grain-128AEADv2 と Grain-128AEAD の Initialization の違い

ストリームの総和は 0 に偏る。そのような性質を利用して解析することで、約 2^{90} の計算量で 128 ビットの秘密鍵を回復した。

オリジナルの Cube 攻撃は発見的な攻撃だった。すなわち、$|I|$ は高々 32 など、$\bigoplus_{\boldsymbol{v}\in C_I} f(\boldsymbol{x},\boldsymbol{v})$ が実時間で評価可能な範囲内の解析だった。2017 年に、Division Property を用いることで、$|I|$ のサイズの制限が事実上なくなり、理論的な方法で Cube 攻撃の安全性評価を可能とする方法が示された [TIHM17]。この攻撃方法は、その後、複数の発展を経て [WHT+18, WHG+19, HJL+20]、一切の仮定なく、非常に大きな $|I|$ における Superpoly を実時間で回復する方法が示された [HLM+20, HLM+21]。文献 [HLM+20] で、$j \in J = \{27, 30, 31, 32, 34, 41, 44, 45, 46, 48, 58, 59, 64, 70, 72\}$ において、$I = \{1, \ldots, 96\} \setminus j$ となる 15 個の cube において、190 段までの Superpoly を回復し、これら回復した Superpoly を用いて 2^{123} の計算量の秘密鍵回復攻撃が示された。一方で、上述した通り、この Superpoly 回復には一切の仮定がないため、言い換えると、これ以上改良できる余地が乏しいことを意味する。したがって、このアプローチでは 190 段より非常に多くの段数を攻撃対象とすることは事実上不可能であり、Initialization の総段数が 512 段におよぶ Grain-128AEADv2 は、十分な大きな安全性マージンを有していると言える。

3.4　その他の攻撃

Grain-128a の Initialization を対象としたその他の解析論文として Conditional Differential Cryptanalysis がある [LM12]。Conditional Differential Cryptanalysis は、その後、Ma らによって、改良される [MTQ17]。また、文献 [DG13, BMST13] では、Grain-128a に対する関連鍵攻撃が示された。両攻撃ともに、攻撃対象は Initialization であり、256 段の Initialization であった Grain-128a に対し、Grain-128AEADv2 は 512 段を用いる。したがって、これらの攻撃が Grain-128AEADv2 に対して脅威になるリスクは小さいと期待できる。

Grain-128AEAD と Grain-128a の Initialization の違いとして、鍵の再ロードがある。鍵の再ロードは内部状態回復攻撃が直接的に秘密鍵回復攻撃に繋がらなくなるため、非常に有効である。しかしながら、Chang と Turan は、Initialization 後の内部状態が既知であるという仮定から Grain-128AEAD の秘密鍵回復攻撃が可能であることを示した [CT21]。この攻撃を受けて、Grain-128AEADv2 では Grain-128AEAD とは異なる Initialization を利用する。表 8 に二方式の Initialization の違いを整理する。

4 TinyJAMBU

4.1 TinyJAMBU の仕様

TinyJAMBU は CAESAR Competition の第三次候補の一つである JAMBU[WH16] の軽量版として NIST LWC で初めて提案された [WH19]。TinyJAMBU は 2 種類の 128 ビットの鍵付き置換 $P1$ および $P2$ を Sponge like に利用する認証暗号モードである。NIST LWC Finalist Round で仕様の改定が行われた [WH21]。

4.1.1 鍵付き置換 P_n

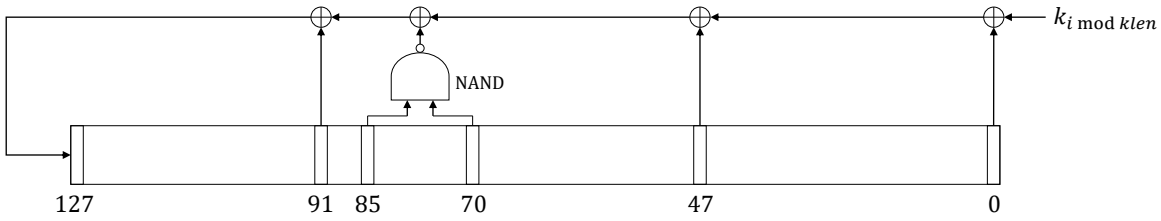

図 7　TinyJAMBU の Keyed Permutation

TinyJAMBU は 128 ビットの鍵付き置換を利用する。P_n は n 回繰り返し構造を持つ鍵付き置換であり、ラウンド関数には 128-bit の非線形フィードバックシフトレジスタを用いる。図 7 に鍵付き置換のラウンド関数を、アルゴリズム 1 に鍵長 $klen$ のときの鍵付き置換 P_n の仕様を示す。

Algorithm 1 鍵付き置換 P_n

function $P_n(\boldsymbol{S} = (s_0, \ldots, s_{127}), \boldsymbol{K} = (k_0, \ldots, k_{klen-1}))$
  for $r = 0$ to $n - 1$ **do**
    $\mathrm{tmp} \leftarrow s_0 \oplus s_{47} \oplus (\sim (s_{70} \wedge s_{85}) \oplus s_{91} \oplus k_{r \bmod klen}$
    for $i = 0$ to 126 **do**
      $s_i \leftarrow s_{i+1}$
    end for
    $s_{127} \leftarrow \mathrm{tmp}$
  end for
  return $\boldsymbol{s}$
end function

TinyJAMBU には 128 ビット安全の TinyJAMBU-128、192 ビット安全の TinyJAMBU-192、256 ビット安全の TinyJAMBU-256 がある。表 9 に各パラメータで利用される繰り返し回数を整理す

Version	Key size	Nonce size	Tag size	$P1$ の段数	$P2$ の段数
TinyJAMBU v1 [WH19]	128	96	64	384	1024
	192	96	64	384	1152
	256	96	64	384	1280
TinyJAMBU v2 [WH21]	128	96	64	640	1024
	192	96	64	640	1152
	256	96	64	640	1280

表 9　TinyJAMBUv1 および TinyJAMBUv2 のパラメータ

る。TinyJAMBU は NIST LWC の finalist round でバージョン 2 に仕様を更新している。本稿では、特に区別する場合、更新前の仕様を TinyJAMBU v1、更新後の仕様を TinyJAMBU v2 とそれぞれ表記し、特に記載がない場合は TinyJAMBU v2 を指すこととする。TinyJAMBU v1 から v2 での主な変更点は $P1$ の段数であり、384 段から 640 段へ大幅に増加している。

4.1.2　TinyJAMBU のモードオペレーション

図 8 に TinyJAMBU の認証暗号モードを示す。TinyJAMBU は二つの異なる鍵付き置換 $P1$ および $P2$ を利用 (実際には、段数の異なる鍵付き置換) し、メッセージは 32 ビットごとに処理する。また、表記法として、$s_{i,\ldots,j}$ は $(s_i \| s_{i+1} \| \cdots \| s_j)$ を意味するものとする。

Algorithm 2 TinyJAMBU の Initialization

function $Init(\boldsymbol{K}, \boldsymbol{N})$
  $\boldsymbol{S} \leftarrow 0^{128}$
  $\boldsymbol{S} \leftarrow P2(\boldsymbol{S}, \boldsymbol{K})$
  for $i \in \{0, 1, 2\}$ **do**
    $s_{36,37,38} \leftarrow s_{36,37,38} \oplus 001$
    $\boldsymbol{S} \leftarrow P1(\boldsymbol{S}, \boldsymbol{K})$
    $s_{96,\ldots,127} \leftarrow s_{96,\ldots,127} \oplus N_{32i,\ldots,32i+31}$
  end for
  return $\boldsymbol{S}$
end function

Initialization　Initialization は Key Setup と Nonce Setup からなる。その仕様を Algorithm2 に示す。

初めに、Key Setup では、初期状態 0^{128} を用意し、鍵付き置換 $P2$ を適用する。適用後の 128 ビット状態は鍵に依存した秘密状態となる。

次に、Nonce Setup では、$s_{36} \| s_{37} \| s_{38}$ の 3 ビットに FrameBits(Nonce の処理中は 001) を排他的論理和し、$P1$ を適用する。適用後、$s_{96} \| s_{97} \| \cdots \| s_{127}$ に Nonce の先頭 32 ビットを排他的論理和す

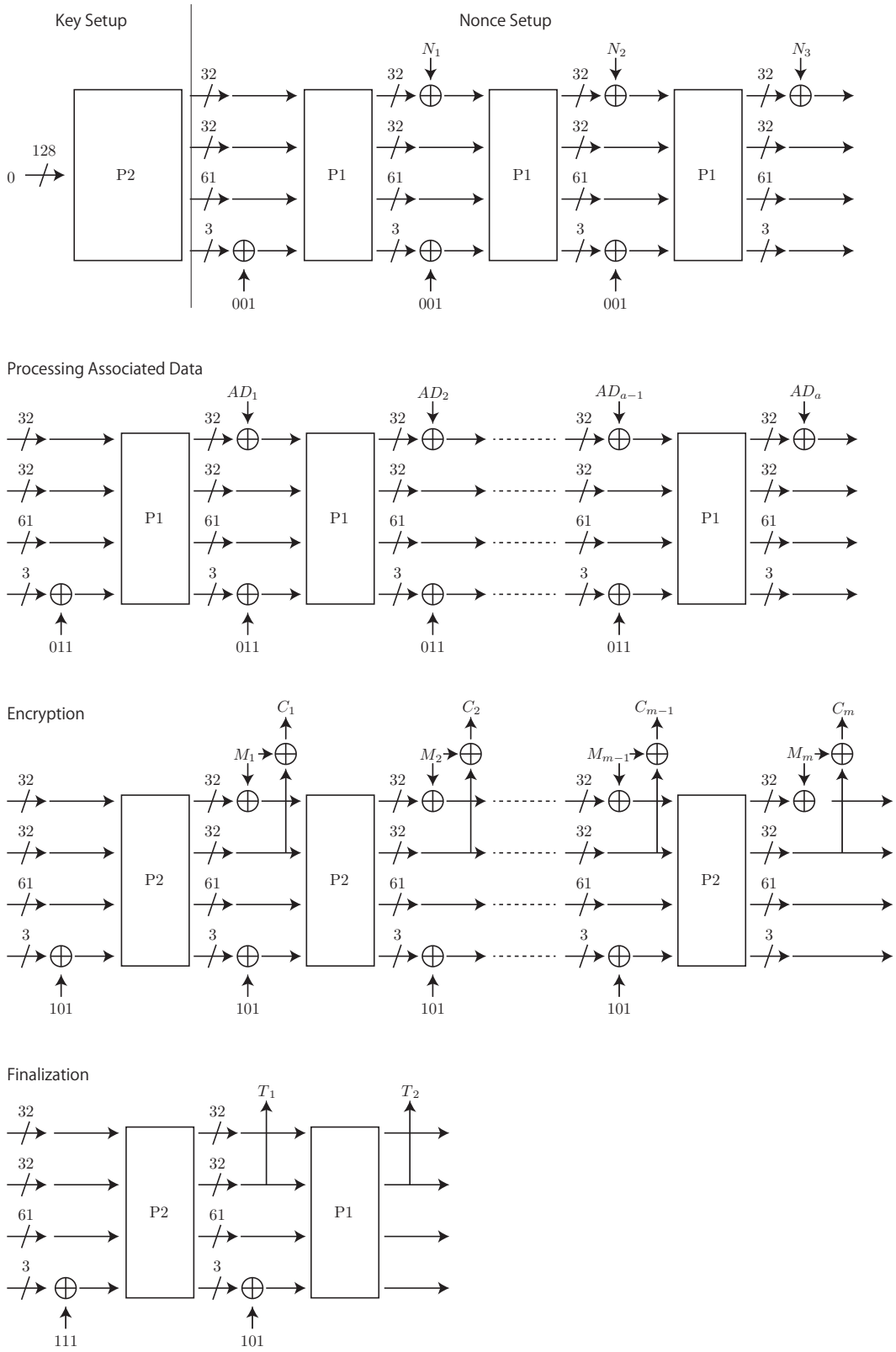

図 8　TinyJAMBU の認証暗号モード

Algorithm 3 TinyJAMBU の Associated Data の処理

function $Proc_{ad}(\boldsymbol{S}, \boldsymbol{K}, (ad_0, \ldots, ad_{adlen-1}))$
  for $i = 0$ to $\lfloor adlen/32 \rfloor$ **do**
    if $adlen - 32i \geq 32$ **then**
      $s_{36,37,38} \leftarrow s_{36,37,38} \oplus 011$
      $\boldsymbol{S} \leftarrow P1(\boldsymbol{S}, \boldsymbol{K})$
      $s_{96,\ldots,127} \leftarrow s_{96,\ldots,127} \oplus ad_{32i,\ldots,32i+31}$
    else if $adlen - 32i > 0$ **then**
      $s_{36,37,38} \leftarrow s_{36,37,38} \oplus 011$
      $\boldsymbol{S} \leftarrow P1(\boldsymbol{S}, \boldsymbol{K})$
      $lenp = adlen \bmod 32$
      $s_{96,\ldots,96+lenp-1} \leftarrow (s_{96,\ldots,96+lenp-1}) \oplus ad_{32i,\ldots,32i+lenp-1}$
      $s_{32,33} \leftarrow s_{32,33} \oplus (lenp/8)$
    end if
  end for
  return $\boldsymbol{S}$
end function

る。この処理を 3 回繰り返すことで、96 ビットの Nonce 全てを内部状態に吸収する。適用後、出力は鍵と Nonce に依存した 128 ビットの秘密状態となる。

Associated Data の処理　Associated Data の処理の仕様を Algorithm3 に示す。

処理は Nonce 吸収時とほぼ同一であり、FrameBits として 011 を利用する。Associated Data を 32 ビットごとに分割して処理する。Associated Data のビット長が 32 の倍数でない場合、長さ分を処理したうえで、さらに $s_{32}\|s_{33}$ に最終ブロックのバイト長を排他的論理和する。これは、ad と $ad\|0^8$ を区別し、バイト長の異なるデータを異なるデータとして取り扱うためである[5)]。詳細は Algorithm3 を参照されたい[6)]。

Encryption の処理　Encryption の処理の仕様を Algorithm4 に示す。

平文の暗号化処理中は FrameBits として 101 を利用する。Encryption の処理は $P1$ の代わりに $P2$ を利用する。$s_{96}\|s_{97}\|\cdots\|s_{127}$ に平文の 32 ビットを排他的論理和、暗号化は同平文と $s_{64}\|s_{65}\|\cdots\|s_{95}$ を排他的論理和することで行われる。全ての平文を処理するまで上記の処理が繰り返されるが、平文のビット長が 32 の倍数ではない場合、Associated Data の処理と同様、長さ分を処理したうえで、さらに $s_{32}\|s_{33}$ に最終ブロックのバイト長を排他的論理和する。

Finalization の処理　Finalization の処理の仕様を Algorithm5 に示す。

5)　TinyJAMBU の仕様を見る限り、入力は必ずバイト単位であり、ビット単位のデータは想定されていない。実際、Associated Data の処理や Encryption の処理は、ビット長が 8 の倍数でないデータに関しては、ビット長が異なっていたとしても同一データとして取り扱う処理となっている。

6)　仕様 [WH21] の記述では、Associated data が空の場合の処理が曖昧である。Associated data が空の場合と 0^{32} の場合を区別するためには、Associated data が空の場合は $Proc_{ad}$ の処理全体が省略される必要がある。

Algorithm 4 TinyJAMBU の Encryption の処理

function $Proc_m(\boldsymbol{S}, \boldsymbol{K}, (m_0, \ldots, m_{mlen-1}))$
  for $i = 0$ to $i = \lfloor mlen/32 \rfloor$ **do**
    if $mlen - 32i \geq 32$ **then**
      $s_{36,37,38} \leftarrow s_{36,37,38} \oplus 101$
      $\boldsymbol{S} \leftarrow P2(\boldsymbol{S}, \boldsymbol{K})$
      $s_{96,\ldots,127} \leftarrow s_{96,\ldots,127} \oplus m_{32i,\ldots,32i+31}$
      $c_{32i,\ldots,32i+31} \leftarrow s_{64,\ldots,95} \oplus m_{32i,\ldots,32i+31}$
    else if $mlen - 32i > 0$ **then**
      $s_{36,37,38} \leftarrow s_{36,37,38} \oplus 101$
      $\boldsymbol{S} \leftarrow P2(\boldsymbol{S}, \boldsymbol{K})$
      $lenp = mlen \bmod 32$
      $s_{96,\ldots,96+lenp-1} \leftarrow s_{96,\ldots,96+lenp-1}) \oplus m_{32i,\ldots,32i+lenp-1}$
      $c_{32i,\ldots,32i+lenp-1} \leftarrow s_{64,\ldots,64+lemp-1}) \oplus m_{32i,\ldots,32i+lenp-1}$
      $s_{32,33} \leftarrow s_{32,33} \oplus (lenp/8)$
    end if
  end for
  return $\boldsymbol{S}$
end function

Finalization の処理中は、FrameBits として 111 を利用する。FrameBits の排他的論理和後、$P2$ を適用。$s_{64} \| s_{65} \| \cdots \| s_{95}$ の 32 ビットをタグ T の先頭 32 ビットとして利用する。さらに FrameBits を排他的論理和後、$P1$ を適用、$s_{64} \| s_{65} \| \cdots \| s_{95}$ の 32 ビットをタグ T の後ろ 32 ビットとして利用する。

4.2 差分解読法・線形解読法

Saha らは、TinyJAMBU に対する最初の第三者解析として、差分解読法および線形解読法の安全性を評価した [SSS+20]。複数個の active AND gate が実際には独立ではないことを利用したうえで、差分特性および線形特性を探索した。結果として、$P1$ の段数が 338 段に縮退された状況で、確率 $2^{-62.68}$ で偽造攻撃が可能であることを示した。TinyJAMBU v1 の $P1$ の段数は 384 段であったため、安全性マージンは 12%まで縮小した。4.1 章で示したように、TinyJAMBU の鍵付き置換 P_n の入出力において、攻撃者が操作可能な範囲は限定される。この限定を除外し、鍵付き置換 P_n をブロック暗号とみなした場合、384 段で 2^{-19} の差分特性があることも示された。

Saha らの攻撃を受けて、設計者らは、NIST LWC Finalist Round で、$P1$ の段数を 384 段から 640 段に修正、安全性マージンを大きく広げた [WH21]。

その後、Li らは線形解読法のアルゴリズムの一つ、松井のアルゴリズム 1 に関して、Linear Hull を考慮したモデルとすることで、$P1$ の段数が 384 段のとき、鍵回復攻撃が可能なことを

Algorithm 5 TinyJAMBU の Encryption の処理

function $Fin(\boldsymbol{S}, \boldsymbol{K})$
  $s_{\{36,\ldots,38\}} = s_{\{36,\ldots,38\}} \oplus 111$
  $\boldsymbol{S} \leftarrow P2(\boldsymbol{S}, \boldsymbol{K})$
  $t_{\{0,\ldots,31\}} = s_{\{64,\ldots,95\}}$
  $s_{\{36,\ldots,38\}} = s_{\{36,\ldots,38\}} \oplus 111$
  $\boldsymbol{S} \leftarrow P2(\boldsymbol{S}, \boldsymbol{K})$
  $t_{\{32,\ldots,63\}} = s_{\{64,\ldots,95\}}$
  return $t_{\{0,\ldots,63\}}$
end function

示した [LMSW22]。これは、TinyJAMBU v1 は安全ではなかったことを意味する。一方で、文献 [LMSW22] で示されたように、Linear Hull を利用した解読法は $P1$ が 387 段までしか実行できない。TinyJAMBU v2 は 640 段の段数を持つため、TinyJAMBUv2 は十分に高い安全性マージンを持つと言える。

表 10 に、文献 [LMSW22] で整理された、TinyJAMBU v1 および v2 に対する現在の解析結果を転記する。

	TinyJAMBU v1			TinyJAMBU v2
Attack Phase	Nonce Setup, AD Processing	Initialization & Encryption	Tag Generation	Tag Generation
Rounds	338/384	2604/3200	384/384	387/640
Nonce respecting	No	Yes	Yes	Yes
Data complexity	$2^{62.68}$	2^{14}	$\geq 2^{96.8}$	$\geq 2^{96.8}$
Success prob.	$\approx 63\%$	N/A	$\geq 89\%$	$\geq 82\%$
Attack type	Forgery	Partial key rec.	Partial key rec.	Partial key rec.
Attack method	Differential	Cube	Linear hull	Linear hull
Target key len.	128,192,256	128	128, 192, 256	128, 192, 256
Referecne	[SSS+20]	[TSY+21]	[LMSW22]	[LMSW22]

表 10　TinyJAMBU v1 および v2 に対する単一鍵攻撃の整理 (文献 [LMSW22] より)

4.3　スライド攻撃

TinyJAMBU の鍵付き置換 P_n 単独の安全性に着目した論文に文献 [SST+22] がある。TinyJAMBU のモードはブロック暗号利用モードであるため、鍵付き置換 P_n は、それ自体がブロック暗号として安全であることが理想である。文献 [SST+22] で、Sibleyras らは、$P1$ や $P2$ の段数に関わらず slide property が維持され、slide pairs を birthday bound の data complexity で発見

可能なことを示した。実際に TinyJAMBU で利用される $P2$ の段数においては、TinyJAMBU-128 で利用される P_{1024} は 2^{65} KP, 2^{65} Time, 2^{64} Memory で、TinyJAMBU-192 で利用される P_{1152} は 2^{65} ACP, 2^{66} Time, 2^{65} Memory で、TinyJAMBU-128 で利用される P_{1280} は $2^{67.6}$ ACP, $2^{69.5}$ Time, $2^{67.5}$ Memory で、それぞれ解読可能なことを示した。興味深い点は、鍵付き置換 P_n 単体で見た場合、それぞれ鍵長が大きく異なるにも関わらず、約 2^{64} の計算量で全て解読可能な点である。

文献 [SST+22] でも示された通り、上記の攻撃は TinyJAMBU の内側の鍵付き置換 P_n をブロック暗号とみなした場合、ブロック暗号としての安全性は有していないことを示すのみである。実際の TinyJAMBU では、攻撃者が P_n に対して操作可能な範囲は限定され、スライド攻撃は実行できない。

4.4 関連鍵攻撃

Dunkelman らは関連鍵攻撃を用いて、TinyJAMBU-192 および TinyJAMBU-256 に対する Practical Forgery Attack[DLG22] を示した。

4.4.1 関連鍵差分特性

文献 [DLG22] で示された TinyJAMBU-256 の関連鍵差分特性を解説する。初めに k_0 が差分を持つ場合を考える。このとき 1 段目の処理後、s_{127} に差分がある状態となる。以降、鍵に差分を入れない場合、37 段目の処理後、s_{91} に差分がある状態となり、このビットは TinyJAMBU の非線形フィードバックシフトレジスタの仕様より、線形 (排他的論理和) でフィードバックされる。差分の攪拌を防止するために k_{37} に差分を追加することで、38 段目の処理後、s_{90} にのみ差分がある状態に出来る。43 段目と 58 段目の処理後、それぞれ s_{85} と s_{70} に差分がある状態となるが、NAND ゲートを通過しても差分が攪拌しない確率は $(1/2)^2 = 2^{-2}$ である[7)]。81 段目の処理後、s_{47} に差分がある状態となり、k_{81} に差分を入れることで、s_{47} の差分が広がることを抑制する。128 段目の処理後、s_0 に差分がある状態となるため、k_{128} に差分を入れることで、s_0 の差分が広がることを抑制する。

以上より、関連鍵差分 $\Delta \boldsymbol{K}$ として、

$$k_0, k_{37}, k_{81}, k_{128}$$

に差分がある場合、

$$\Pr[P_{256}(\boldsymbol{S}, \boldsymbol{K}) = P_{256}(\boldsymbol{S}, \boldsymbol{K} \oplus \Delta \boldsymbol{K})] = 2^{-2}$$

が成立する。TinyJAMBU-256 の $P2$ は 1280 段であるため、$P2$ は P_{256} を 5 回繰り返す構造を持つ。したがって、

$$\Pr[P_{1280}(\boldsymbol{S}, \boldsymbol{K}) = P_{1280}(\boldsymbol{S}, \boldsymbol{K} \oplus \Delta \boldsymbol{K})] = 2^{-10}$$

7) このとき、差分が攪拌しないことは、NAND ゲートの差分がない方のビット値が 0 であることを意味する。この性質を利用することで、鍵回復攻撃も可能である [DLG22]。

となる。

TinyJAMBU-256 の $P1$ の段数は 640 段であり、640 は 256 の倍数ではない。したがって、上記の関連鍵差分特性を用いた場合、$P1$ 適用後は内部状態に差分が、具体的には s_0 に差分が残る状態となる。この問題を回避するため、鍵差分の位置を動かす。例えば、関連鍵差分 $\Delta \boldsymbol{K}$ として、

$$k_{127}, k_{164}, k_{208}, k_{255}$$

に差分を入れる。P_{256} を適用する限りは、$k_0, k_{37}, k_{81}, k_{128}$ に差分を入れていた場合と同様に、確率 2^{-2} で、ゼロ差分はゼロ差分へ伝搬する。このとき、$P1$ 適用後の内部状態は、s_{127} に差分が残る状態となる。したがって $P1$ 適用後の Nonce に適切な差分を入れることで、s_{127} の差分を打ち消すことができる。

上記の関連鍵差分特性は TinyJAMBU-192 でも同様に構成可能である。しかしながら、TinyJAMBU-128 に関しては、129 段目での鍵差分による差分の打ち消しが出来ないため同様の特性は存在しない。しかしながら、攻撃者が鍵付き置換 P_n 単独を解析し、入力 $\boldsymbol{S}$ に差分を入れられる場合は、P_{1024} で確率 2^{-16} となる関連鍵差分特性は構成可能である。

4.4.2 関連鍵偽造攻撃

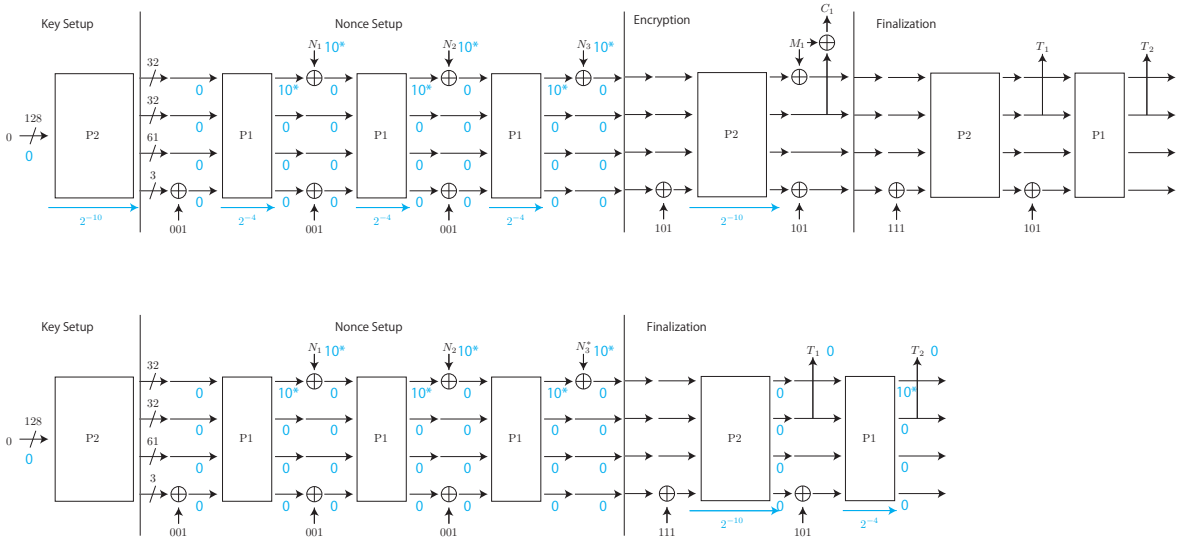

図 9　TinyJAMBU-256 に対する関連鍵偽造攻撃の概要

図 9 に、TinyJAMBU-256 に対する関連鍵偽造攻撃の概要を示す。

攻撃者は “Golden Key and Nonce”、すなわち、Key Setup と Nonce Setup で希望する関連鍵差分特性を満足する鍵と Nonce の探索から始める。Key Setup と Nonce Setup で希望する関連鍵差分特性を満たす確率は 2^{-22} であり、Encryption で希望する関連鍵差分特性を満たす確率は 2^{-10} である。したがって、32 ビットの平文を暗号化し、32 ビットの暗号文の差分の有無を確認する。ラン

ダムで差分がない確率と、関連鍵差分特性を満たす確率はともに 2^{-32} であるため、2^{32} 個の鍵、Nonce ペアのペアを用いることで、攻撃者は高い確率で “Golden Key and Nonce” を手に入れることができる。

次に、“Golden Key and Nonce” を基本として、N_3 の値を変更する。Key Setup と Nonce Setup は希望する関連鍵差分特性を満たしているため、このとき、タグが一致する確率は 2^{-14} と非常に高い。したがって、$2^{32} + 2^{14}$ の計算量で、関連鍵偽造攻撃が可能である[8)]。

参考文献

[ÅHJM11] Martin Ågren, Martin Hell, Thomas Johansson, and Willi Meier. Grain-128a: a new version of grain-128 with optional authentication. *Int. J. Wirel. Mob. Comput.*, 5(1):48–59, 2011.

[BC10] Christina Boura and Anne Canteaut. A zero-sum property for the keccak-f permutation with 18 rounds. In *IEEE International Symposium on Information Theory, ISIT 2010, June 13-18, 2010, Austin, Texas, USA, Proceedings*, pages 2488–2492. IEEE, 2010.

[BC16] Christina Boura and Anne Canteaut. Another view of the division property. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I*, volume 9814 of *Lecture Notes in Computer Science*, pages 654–682. Springer, 2016.

[BCC11] Christina Boura, Anne Canteaut, and Christophe De Cannière. Higher-order differential properties of keccak and *Luffa*. In Antoine Joux, editor, *Fast Software Encryption - 18th International Workshop, FSE 2011, Lyngby, Denmark, February 13-16, 2011, Revised Selected Papers*, volume 6733 of *Lecture Notes in Computer Science*, pages 252–269. Springer, 2011.

[BDKW19] Achiya Bar-On, Orr Dunkelman, Nathan Keller, and Ariel Weizman. DLCT: A new tool for differential-linear cryptanalysis. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part I*, volume 11476 of *Lecture Notes in Computer Science*, pages 313–342. Springer, 2019.

[BDPA08] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. On the indifferentiability of the sponge construction. In Nigel P. Smart, editor, *Advances in Cryptology - EUROCRYPT 2008, 27th Annual International Conference on the Theory and Appli-*

8） 文献 [DLG22] では、32 ビットの平文を用いた関連鍵偽造攻撃が示されている。

cations of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings, volume 4965 of *Lecture Notes in Computer Science*, pages 181–197. Springer, 2008.

[Ber10] Daniel J. Bernstein. Second preimages for 6 (7? (8??)) rounds of keccak?, 2010. Posted on the NIST mailing list.

[BGM06] Côme Berbain, Henri Gilbert, and Alexander Maximov. Cryptanalysis of grain. In Matthew J. B. Robshaw, editor, *Fast Software Encryption, 13th International Workshop, FSE 2006, Graz, Austria, March 15-17, 2006, Revised Selected Papers*, volume 4047 of *Lecture Notes in Computer Science*, pages 15–29. Springer, 2006.

[BMST13] Subhadeep Banik, Subhamoy Maitra, Santanu Sarkar, and Meltem Sönmez Turan. A chosen IV related key attack on grain-128a. In Colin Boyd and Leonie Simpson, editors, *Information Security and Privacy - 18th Australasian Conference, ACISP 2013, Brisbane, Australia, July 1-3, 2013. Proceedings*, volume 7959 of *Lecture Notes in Computer Science*, pages 13–26. Springer, 2013.

[BS90] Eli Biham and Adi Shamir. Differential cryptanalysis of des-like cryptosystems. In Alfred Menezes and Scott A. Vanstone, editors, *Advances in Cryptology - CRYPTO '90, 10th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1990, Proceedings*, volume 537 of *Lecture Notes in Computer Science*, pages 2–21. Springer, 1990.

[CJM02] Philippe Chose, Antoine Joux, and Michel Mitton. Fast correlation attacks: An algorithmic point of view. In Lars R. Knudsen, editor, *Advances in Cryptology - EUROCRYPT 2002, International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, April 28 - May 2, 2002, Proceedings*, volume 2332 of *Lecture Notes in Computer Science*, pages 209–221. Springer, 2002.

[CT21] Donghoon Chang and Meltem Sönmez Turan. Recovering the key from the internal state of grain-128aead. *IACR Cryptol. ePrint Arch.*, page 439, 2021.

[DEMSa] Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. Ascon v1.2 - submission to nist.

[DEMSb] Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. Ascon v1.2 - submission to the caesar competition.

[DEMS15] Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. Cryptanalysis of ascon. In Kaisa Nyberg, editor, *Topics in Cryptology - CT-RSA 2015, The Cryptographer's Track at the RSA Conference 2015, San Francisco, CA, USA, April 20-24, 2015. Proceedings*, volume 9048 of *Lecture Notes in Computer Science*, pages 371–387. Springer, 2015.

[DEMS19] Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. Preliminary analysis of ascon-xof and ascon-hash, 2019.

[DG13] Lin Ding and Jie Guan. Related key chosen IV attack on grain-128a stream cipher.

IEEE Trans. Inf. Forensics Secur., 8(5):803–809, 2013.

[DGP+11] Itai Dinur, Tim Güneysu, Christof Paar, Adi Shamir, and Ralf Zimmermann. An experimentally verified attack on full grain-128 using dedicated reconfigurable hardware. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings*, volume 7073 of *Lecture Notes in Computer Science*, pages 327–343. Springer, 2011.

[DKR97] Joan Daemen, Lars R. Knudsen, and Vincent Rijmen. The block cipher square. In Eli Biham, editor, *Fast Software Encryption, 4th International Workshop, FSE '97, Haifa, Israel, January 20-22, 1997, Proceedings*, volume 1267 of *Lecture Notes in Computer Science*, pages 149–165. Springer, 1997.

[DLG22] Orr Dunkelman, Eran Lambooij, and Shibam Ghosh. Practical related-key forgery attacks on the full tinyjambu-192/256. *IACR Cryptol. ePrint Arch.*, page 1122, 2022.

[DS09] Itai Dinur and Adi Shamir. Cube attacks on tweakable black box polynomials. In Antoine Joux, editor, *Advances in Cryptology - EUROCRYPT 2009, 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April 26-30, 2009. Proceedings*, volume 5479 of *Lecture Notes in Computer Science*, pages 278–299. Springer, 2009.

[DS11] Itai Dinur and Adi Shamir. Breaking grain-128 with dynamic cube attacks. In Antoine Joux, editor, *Fast Software Encryption - 18th International Workshop, FSE 2011, Lyngby, Denmark, February 13-16, 2011, Revised Selected Papers*, volume 6733 of *Lecture Notes in Computer Science*, pages 167–187. Springer, 2011.

[EME22] Johannes Erlacher, Florian Mendel, and Maria Eichlseder. Bounds for the security of ascon against differential and linear cryptanalysis. *IACR Trans. Symmetric Cryptol.*, 2022(1):64–87, 2022.

[FLN+21] Antonio Flórez-Gutiérrez, Gaëtan Leurent, María Naya-Plasencia, Léo Perrin, André Schrottenloher, and Ferdinand Sibleyras. Internal symmetries and linear properties: Full-permutation distinguishers and improved collisions on gimli. *J. Cryptol.*, 34(4):45, 2021.

[GD21] Shibam Ghosh and Orr Dunkelman. Automatic search for bit-based division property. In Patrick Longa and Carla Ràfols, editors, *Progress in Cryptology - LATINCRYPT 2021 - 7th International Conference on Cryptology and Information Security in Latin America, Bogotá, Colombia, October 6-8, 2021, Proceedings*, volume 12912 of *Lecture Notes in Computer Science*, pages 254–274. Springer, 2021.

[GP10] Henri Gilbert and Thomas Peyrin. Super-sbox cryptanalysis: Improved attacks for aes-like permutations. In Seokhie Hong and Tetsu Iwata, editors, *Fast Software Encryption, 17th International Workshop, FSE 2010, Seoul, Korea, February 7-10, 2010, Revised*

Selected Papers, volume 6147 of *Lecture Notes in Computer Science*, pages 365–383. Springer, 2010.

[GPT21] David Gérault, Thomas Peyrin, and Quan Quan Tan. Exploring differential-based distinguishers and forgeries for ASCON. *IACR Trans. Symmetric Cryptol.*, 2021(3):102–136, 2021.

[GRW16] Faruk Göloglu, Vincent Rijmen, and Qingju Wang. On the division property of s-boxes. *IACR Cryptol. ePrint Arch.*, page 188, 2016.

[HJL+20] Yonglin Hao, Lin Jiao, Chaoyun Li, Willi Meier, Yosuke Todo, and Qingju Wang. Links between division property and other cube attack variants. *IACR Trans. Symmetric Cryptol.*, 2020(1):363–395, 2020.

[HJM+a] Martin Hell, Thomas Johansson, Alexander Maximov, Willi Meier, Jonathan Sönnerup, and Hirotaka Yoshida. Grain-128aead - a lightweight aead stream cipher.

[HJM+b] Martin Hell, Thomas Johansson, Alexander Maximov, Willi Meier, Jonathan Sönnerup, and Hirotaka Yoshida. Grain-128aeadv2 - a lightweight aead stream cipher.

[HJM05] Martin Hell, Thomas Johansson, and Willi Meier. Grain - a stream cipher for constrained environments, 2005.

[HJM07] Martin Hell, Thomas Johansson, and Willi Meier. Grain: a stream cipher for constrained environments. *Int. J. Wirel. Mob. Comput.*, 2(1):86–93, 2007.

[HJMM06] Martin Hell, Thomas Johansson, Alexander Maximov, and Willi Meier. A stream cipher proposal: Grain-128. In *Proceedings 2006 IEEE International Symposium on Information Theory, ISIT 2006, The Westin Seattle, Seattle, Washington, USA, July 9-14, 2006*, pages 1614–1618. IEEE, 2006.

[HLLT20] Phil Hebborn, Baptiste Lambin, Gregor Leander, and Yosuke Todo. Lower bounds on the degree of block ciphers. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part I*, volume 12491 of *Lecture Notes in Computer Science*, pages 537–566. Springer, 2020.

[HLM+20] Yonglin Hao, Gregor Leander, Willi Meier, Yosuke Todo, and Qingju Wang. Modeling for three-subset division pr operty without unknown subset - improved cube attacks against trivium and grain-128aead. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part I*, volume 12105 of *Lecture Notes in Computer Science*, pages 466–495. Springer, 2020.

[HLM+21] Yonglin Hao, Gregor Leander, Willi Meier, Yosuke Todo, and Qingju Wang. Modeling for three-subset division property without unknown subset. *J. Cryptol.*, 34(3):22, 2021.

[HMMD22] Solane El Hirch, Silvia Mella, Alireza Mehrdad, and Joan Daemen. Improved differential and linear trail bounds for ASCON. *IACR Cryptol. ePrint Arch.*, page 1377, 2022.

[JLM14] Philipp Jovanovic, Atul Luykx, and Bart Mennink. Beyond 2 c/2 security in sponge-based authenticated encryption modes. 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 I*, volume 8873 of *Lecture Notes in Computer Science*, pages 85–104. Springer, 2014.

[KW02] Lars R. Knudsen and David A. Wagner. Integral cryptanalysis. In Joan Daemen and Vincent Rijmen, editors, *Fast Software Encryption, 9th International Workshop, FSE 2002, Leuven, Belgium, February 4-6, 2002, Revised Papers*, volume 2365 of *Lecture Notes in Computer Science*, pages 112–127. Springer, 2002.

[Lai94] Xuejia Lai. *Higher Order Derivatives and Differential Cryptanalysis*, pages 227–233. Springer US, 1994.

[LDW17] Zheng Li, Xiaoyang Dong, and Xiaoyun Wang. Conditional cube attack on round-reduced ASCON. *IACR Trans. Symmetric Cryptol.*, 2017(1):175–202, 2017.

[LH94] Susan K. Langford and Martin E. Hellman. Differential-linear cryptanalysis. In Yvo Desmedt, editor, *Advances in Cryptology - CRYPTO '94, 14th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1994, Proceedings*, volume 839 of *Lecture Notes in Computer Science*, pages 17–25. Springer, 1994.

[LLL21] Meicheng Liu, Xiaojuan Lu, and Dongdai Lin. Differential-linear cryptanalysis from an algebraic perspective. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part III*, volume 12827 of *Lecture Notes in Computer Science*, pages 247–277. Springer, 2021.

[LM12] Michael Lehmann and Willi Meier. Conditional differential cryptanalysis of grain-128a. In Josef Pieprzyk, Ahmad-Reza Sadeghi, and Mark Manulis, editors, *Cryptology and Network Security, 11th International Conference, CANS 2012, Darmstadt, Germany, December 12-14, 2012. Proceedings*, volume 7712, pages 1–11. Springer, 2012.

[LMM91] Xuejia Lai, James L. Massey, and Sean Murphy. Markov ciphers and differential cryptanalysis. In Donald W. Davies, editor, *Advances in Cryptology - EUROCRYPT '91, Workshop on the Theory and Application of of Cryptographic Techniques, Brighton, UK, April 8-11, 1991, Proceedings*, volume 547 of *Lecture Notes in Computer Science*, pages 17–38. Springer, 1991.

[LMSW22] Muzhou Li, Nicky Mouha, Ling Sun, and Meiqin Wang. Revisiting the extension of matsui's algorithm 1 to linear hulls: Application to tinyjambu. *IACR Trans. Symmetric Cryptol.*, 2022(2):161–200, 2022.

[LZWW17] Yanbin Li, Guoyan Zhang, Wei Wang, and Meiqin Wang. Cryptanalysis of round-reduced ASCON. *Sci. China Inf. Sci.*, 60(3):38102, 2017.

[Mat93] Mitsuru Matsui. Linear cryptanalysis method for DES cipher. In Tor Helleseth, editor, *Advances in Cryptology - EUROCRYPT '93, Workshop on the Theory and Application of of Cryptographic Techniques, Lofthus, Norway, May 23-27, 1993, Proceedings*, volume 765 of *Lecture Notes in Computer Science*, pages 386–397. Springer, 1993.

[MR22] Rusydi H. Makarim and Raghvendra Rohit. Towards tight differential bounds of ascon A hybrid usage of SMT and MILP. *IACR Trans. Symmetric Cryptol.*, 2022(3):303–340, 2022.

[MTQ17] Zhen Ma, Tian Tian, and Wen-Feng Qi. Conditional differential attacks on grain-128a stream cipher. *IET Inf. Secur.*, 11(3):139–145, 2017.

[RHSS21] Raghvendra Rohit, Kai Hu, Sumanta Sarkar, and Siwei Sun. Misuse-free key-recovery and distinguishing attacks on 7-round ascon. *IACR Trans. Symmetric Cryptol.*, 2021(1):130–155, 2021.

[RS21] Raghvendra Rohit and Santanu Sarkar. Diving deep into the weak keys of round reduced ascon. *IACR Trans. Symmetric Cryptol.*, 2021(4):74–99, 2021.

[sha15] Sha-3 standard: Permutation-based hash and extendable-output functions, 2015.

[SSS+20] Dhiman Saha, Yu Sasaki, Danping Shi, Ferdinand Sibleyras, Siwei Sun, and Yingjie Zhang. On the security margin of tinyjambu with refined differential and linear cryptanalysis. *IACR Trans. Symmetric Cryptol.*, 2020(3):152–174, 2020.

[SST+22] Ferdinand Sibleyras, Yu Sasaki, Yosuke Todo, Akinori Hosoyamada, and Kan Yasuda. Birthday-bound slide attacks on tinyjambu's keyed-permutations for all key sizes. In Chen-Mou Cheng and Mitsuaki Akiyama, editors, *Advances in Information and Computer Security - 17th International Workshop on Security, IWSEC 2022, Tokyo, Japan, August 31 - September 2, 2022, Proceedings*, volume 13504 of *Lecture Notes in Computer Science*, pages 107–127. Springer, 2022.

[TIHM17] Yosuke Todo, Takanori Isobe, Yonglin Hao, and Willi Meier. Cube attacks on non-blackbox polynomials based on division property. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part III*, volume 10403 of *Lecture Notes in Computer Science*, pages 250–279. Springer, 2017.

[TIM+18] Yosuke Todo, Takanori Isobe, Willi Meier, Kazumaro Aoki, and Bin Zhang. Fast correlation attack revisited - cryptanalysis on full grain-128a, grain-128, and grain-v1. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part II*, volume 10992 of *Lecture Notes in Computer Science*, pages 129–159. Springer, 2018.

[Tod15] Yosuke Todo. Structural evaluation by generalized integral property. In Elisabeth Oswald and Marc 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.

[TSY+21] Wil Liam Teng, Md. Iftekhar Salam, Wei-Chuen Yau, Josef Pieprzyk, and Raphaël C.-W. Phan. Cube attacks on round-reduced tinyjambu. *IACR Cryptol. ePrint Arch.*, page 1164, 2021.

[WH16] Hongjun Wu and Tao Huang. The jambu lightweight authentication encryption mode (v2.1), 2016.

[WH19] Hongjun Wu and Tao Huang. Tinyjambu: A family of lightweight authenticated encryption algorithms, 2019.

[WH21] Hongjun Wu and Tao Huang. Tinyjambu: A family of lightweight authenticated encryption algorithms (version 2), 2021.

[WHG+19] Senpeng Wang, Bin Hu, Jie Guan, Kai Zhang, and Tairong Shi. Milp-aided method of searching division property using three subsets and applications. In Steven D. Galbraith and Shiho Moriai, editors, *Advances in Cryptology - ASIACRYPT 2019 - 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, December 8-12, 2019, Proceedings, Part III*, volume 11923 of *Lecture Notes in Computer Science*, pages 398–427. Springer, 2019.

[WHT+18] Qingju Wang, Yonglin Hao, Yosuke Todo, Chaoyun Li, Takanori Isobe, and Willi Meier. Improved division property based cube attacks exploiting algebraic properties of superpoly. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part I*, volume 10991 of *Lecture Notes in Computer Science*, pages 275–305. Springer, 2018.

[YLW+19] Hailun Yan, Xuejia Lai, Lei Wang, Yu Yu, and Yiran Xing. New zero-sum distinguishers on full 24-round keccak-f using the division property. *IET Inf. Secur.*, 13(5):469–478, 2019.

[ZDW19] Rui Zong, Xiaoyang Dong, and Xiaoyun Wang. Collision attacks on round-reduced gimli-hash/ascon-xof/ascon-hash. *IACR Cryptol. ePrint Arch.*, page 1115, 2019.