軽量暗号の安全性に関する調査及び評価
(Photon-Beetle, Sparkle, Tsudik's keymode)

岩田　哲

名古屋大学大学院工学研究科

2022 年 12 月

エグゼクティブサマリ

本報告書では，PHOTON-Beetle，Sparkle，Tsudik's keymode の安全性に関する調査及び評価を報告する．

PHOTON-Beetle． PHOTON-Beetle は暗号学的置換 P_{256} に基づく認証暗号 PHOTON-Beetle-AEAD とハッシュ関数 PHOTON-Beetle-Hash からなる．

- P_{256} は 12 ラウンドの繰り返し構造を有する入出力長 256 ビットの暗号学的置換である．仕様段数である 12 ラウンドの P_{256} に対し，[WGR18, CT-RSA 2018] において，サイズ 2^{184} の zero-sum 分割を用い，時間計算量 2^{183} の識別攻撃が示されている．ただし汎用的攻撃からの利得は小さく，またハッシュ関数などのモードに組み込んだ際の安全性を直ちに脅かすものではない．
- PHOTON-Beetle-AEAD は PHOTON-Beetle-AEAD[128] と PHOTON-Beetle-AEAD[32] からなる．設計者による安全性の主張は次の表の通りである．安全性はビットで表現されている．

モード	安全性モデル	データ計算量	時間計算量
PHOTON-Beetle-AEAD[128]	IND-CPA	121	121
PHOTON-Beetle-AEAD[128]	INT-CTXT	121	121
PHOTON-Beetle-AEAD[32]	IND-CPA	128	128
PHOTON-Beetle-AEAD[32]	INT-CTXT	128	128

  - 表にあるビット安全性の主張を覆す解析結果は知られていない．
  - 表の一部（赤字部分）は理論的根拠がない数字が挙げられている．
- PHOTON-Beetle-Hash は PHOTON-Beetle-Hash[32] が唯一の推奨方式であり，設計者による安全性の主張は次の表の通りである．安全性はビットで表現されている．

モード	安全性モデル	時間計算量
PHOTON-Beetle-Hash[32]	衝突	112（データ計算量 $2^{111.5}$）
PHOTON-Beetle-Hash[32]	原像	128

  - 表にあるビット安全性の主張を覆す解析結果は知られていない．

Sparkle． Sparkle は暗号学的置換の族であり，Schwaemm は Sparkle を暗号学的置換として用いた Sponge 構造に基づく認証暗号であり，Esch は Sparkle を暗号学的置換として用いた Sponge 構造に基づくハッシュ関数である．

- Sparkle には big instances と slim instances があり，Sparkle の big instances の安全性の主張は，入出力長 n ビットに対し，時間計算量とデータ計算量が $2^{n/2}$ を下回る識別攻撃が存在しないことである．slim instances は Sponge 構造に組み込んだ場合の安全性のみを想定し，レート部分に対応する入力のみを制御できる敵に対する識別不可能性を主張している．これらの安全性の主張を覆す解析結果は知られていない．

- Schwaemm は合計 4 通りのパラメータがあり，それぞれの安全性の主張は次の表のとおりである．安全性はビット単位で表現されており，データ制限はバイト単位である．

方式	安全性	データ制限（バイト）
Schwaemm256-128	120	2^{68}
Schwaemm192-192	184	2^{68}
Schwaemm128-128	120	2^{68}
Schwaemm256-256	248	2^{133}

  – 表にあるビット安全性の主張を覆す解析結果は知られていない．

- Esch は Esch256 と Esch384 の 2 通りが定義されており，次の表の安全性が主張されている．安全性はビット単位で表現されており，データ制限はバイト単位である．

方式	衝突	第 2 原像	原像	データ制限（バイト）
Esch256	128	128	128	2^{132}
Esch384	192	192	192	2^{196}

  – 表にあるビット安全性の主張を覆す解析結果は知られていない．

Tsudik's keymode． Tsudik's keymode は軽量ハッシュ関数を構成要素として用いる MAC である．

- ハッシュ関数が length-extension 攻撃を許す場合には Tsudik's keymode に対する偽造攻撃が可能であり，明らかな脆弱性を有している．Merkle-Damgård 変換に基づくハッシュ関数では length-extension 攻撃が可能であり，SHA-256 等を Tsudik's keymode で利用することはできない．

- 一方，ハッシュ関数が（可変長入力の）ランダムオラクルである場合には Tsudik's keymode は理想的に安全な擬似ランダム関数になる．したがって，ランダムオラクルからの強識別不可能性が証明できるようなハッシュ関数を用いれば，Tsudik's keymode の利用に安全性上の問題は見受けられない．

目次

1 はじめに **5**
- 1.1 目的 . . . 5
- 1.2 本報告書の構成 . . . 5

2 PHOTON-Beetle の安全性に関する調査及び評価 **6**
- 2.1 方式概要 . . . 6
  - 2.1.1 P_{256} の仕様の概要 . . . 6
  - 2.1.2 PHOTON-Beetle-AEAD の仕様の概要 . . . 7
  - 2.1.3 PHOTON-Beetle-Hash の仕様の概要 . . . 9
- 2.2 安全性調査 . . . 9
  - 2.2.1 安全性の主張 . . . 9
  - 2.2.2 P_{256} の安全性調査 . . . 10
  - 2.2.3 PHOTON-Beetle-AEAD の安全性調査 . . . 11
  - 2.2.4 PHOTON-Beetle-Hash の安全性調査 . . . 14
- 2.3 安全性評価 . . . 15
  - 2.3.1 P_{256} の安全性評価 . . . 15
  - 2.3.2 PHOTON-Beetle-AEAD の安全性評価 . . . 15
  - 2.3.3 PHOTON-Beetle-Hash の安全性評価 . . . 16

3 Sparkle の安全性に関する調査及び評価 **17**
- 3.1 方式概要 . . . 17
  - 3.1.1 Sparkle の仕様の概要 . . . 17
  - 3.1.2 Schwaemm の仕様の概要 . . . 19
  - 3.1.3 Esch の仕様の概要 . . . 20
- 3.2 安全性調査 . . . 21
  - 3.2.1 安全性の主張 . . . 21
  - 3.2.2 Sparkle の安全性の主張 . . . 21
  - 3.2.3 Schwaemm の安全性の主張 . . . 22
  - 3.2.4 Esch の安全性の主張 . . . 22
  - 3.2.5 Sparkle の安全性調査 . . . 22
  - 3.2.6 Schwaemm の安全性調査 . . . 23
  - 3.2.7 Esch の安全性調査 . . . 23
- 3.3 安全性評価 . . . 24
  - 3.3.1 Sparkle の安全性評価 . . . 24
  - 3.3.2 Schwaemm の安全性評価 . . . 24

3.3.3 Esch の安全性評価 25

4 Tsudik's keymode の安全性に関する調査及び評価 32
4.1 方式概要 32
4.2 安全性調査 32
4.2.1 [Tsu92] の安全性解析 32
4.2.2 [ISO19] の安全性解析 34
4.2.3 文献調査 34
4.3 安全性評価 36

参考文献 38

Chapter 1

はじめに

1.1 目的

2019 年度量子コンピュータ時代に向けた暗号の在り方検討タスクフォースにおいて，軽量暗号は CRYPTREC 暗号リストに組み込まず，別途ガイドラインという形で取り扱うことが決定された．これを受け，2020 年度第 2 回暗号技術検討会において，2016 年度に作成した「CRYPTREC 暗号技術ガイドライン（軽量暗号）」を 2023 年度中を目処に更新することが承認された．また，2021 年度暗号技術検討会において，ガイドライン更新に向けた 2022 年度の活動として，NIST 軽量暗号コンペティションでファイナリストに選定された 10 方式に加え，軽量メッセージ認証コードの 1 つである Tsudik's keymode を対象とした安全性に関する調査・評価を実施することが承認された．

本報告書では NIST 軽量暗号コンペティションでファイナリストに選定された 10 方式のうち Photon-Beetle と Sparkle，及び軽量メッセージ認証コードの Tsudik's keymode を対象とした安全性に関する調査・評価を行い，執筆時点のこれらの方式の安全性を明らかにすることを目的とする．

1.2 本報告書の構成

本報告書は全 4 章からなり，2章では PHOTON-Beetle を，3章では Sparkle を，4章では Tsudik's keymode を扱う．各章において，方式の概要を述べ，安全性調査と安全性評価を述べる．

Chapter 2

PHOTON-Beetleの安全性に関する調査及び評価

PHOTON-Beetle は Christof Beierle, Zhenzhen Bao, Avik Chakraborti, Nilanjan Datta, Jian Guo, Mridul Nandi, Thomas Peyrin, Kan Yasuda により設計された認証暗号，およびハッシュ関数の族である [BCD+21]．認証暗号を PHOTON-Beetle-AEAD といい，ハッシュ関数を PHOTON-Beetle-Hash という．これらは，ハッシュ関数 PHOTON [GPP11] に用いられている暗号学的置換 P_{256} を構成要素として用い，認証暗号 PHOTON-Beetle-AEAD は Duplex Sponge [BDPA11] を改良した Beetle [CDNY18a] に基づき設計されており，また，ハッシュ関数 PHOTON-Beetle-Hash は Sponge 構造 [BDPA08] に基づいている．

本章では，PHOTON-Beetle の方式概要，安全性調査，安全性評価について述べる．

2.1 方式概要

本章では，P_{256}，PHOTON-Beetle-AEAD，PHOTON-Beetle-Hash の仕様の概要をまとめる．より詳細な仕様は [BCD+21] に記載されている．

以下，ビット列 X のビット長を $|X|$ と書き，ビット列 X, Y の連結を $X \| Y$ と書く．

2.1.1 P_{256} の仕様の概要

PHOTON-Beetle-AEAD，PHOTON-Beetle-Hash では P_{256} を PHOTON_{256} と呼んでいる．P_{256} は入出力長 256 ビットの暗号学的置換であり，AES の設計に基づく SPN 構造（Substitution-Permutation Network 構造）を有している．内部状態は各要素が 4 ビットである 8×8 の行列として表現され，各 4 ビットの要素をセルという．この内部状態に対し，12 ラウンドの繰り返し処理を行う．各ラウンドは 4 つのレイヤ AddConstant，SubCells，ShiftRows，MixColumnSerial からなる．AddConstant は内部状態にラウンド依存の定数を XOR する．SubCells は内部状態の各要素に 4 ビットの S-box を適用する．ShiftRows はセルの位置を行方向に巡回置換により移動させ，MixColumnSerial は内部状態の各列に行列を乗算する．

図 2.1に擬似コードの形でそれぞれを記載する．入力は 8×8 の行列 X であり，各 $X[i, j]$（$0 \leq i, j \leq 7$）は 4 ビットである．

- 図 2.1において，AddConstant の $RC[12]$ と $IC[8]$ は各要素が 4 ビットの配列である．
- SubCells の S-box は次のように定義される．

$P_{256}(X)$

1. for $i = 0$ to 11
2. $\quad X \leftarrow \mathsf{AddConstant}(X, i)$
3. $\quad X \leftarrow \mathsf{SubCells}(X)$
4. $\quad X \leftarrow \mathsf{ShiftRows}(X, i)$
5. $\quad X \leftarrow \mathsf{MixColumnSerial}(X, i)$
6. return X

$\mathsf{AddConstant}(X, k)$

1. $RC[12] \leftarrow \{1, 3, 7, 14, 13, 11, 6, 12, 9, 2, 5, 10\}$
2. $IC[8] \leftarrow \{0, 1, 3, 7, 15, 14, 12, 8\}$
3. for $i = 0$ to 7
4. $\quad X[i, 0] \leftarrow X[i, 0] \oplus RC[k] \oplus IC[i]$
5. return X

$\mathsf{SubCells}(X)$

1. for $i = 0$ to 7, $j = 0$ to 7
2. $\quad X[i, j] \leftarrow \mathsf{S\text{-}box}(X[i, j])$
3. return X

$\mathsf{ShiftRows}(X)$

1. for $i = 0$ to 7, $j = 0$ to 7
2. $\quad X'[i, j] \leftarrow X[i, (j + i) \bmod 8]$
3. return X'

$\mathsf{MixColumnSerial}(X)$

1. $M \leftarrow \mathsf{Serial}[2, 4, 2, 11, 2, 8, 5, 6]$
2. $X \leftarrow M^8 \cdot X$
3. return X

図 2.1: 暗号学的置換 P_{256} の擬似コード.

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$\mathsf{S\text{-}box}(x)$	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

ただし，16 進数表記である.

- MixColumnSerial の $M = \mathsf{Serial}[2, 4, 2, 11, 2, 8, 5, 6]$ は次の 8×8 行列である.

$$M = \mathsf{Serial}[2, 4, 2, 11, 2, 8, 5, 6] = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 2 & 4 & 2 & 11 & 2 & 8 & 5 & 6 \end{pmatrix}$$

各要素は有限体 $\mathrm{GF}(2^4)$ の元であり，行列の演算は既約多項式 $x^4 + x + 1$ で定まる有限体 $\mathrm{GF}(2^4)$ 上で定義される.

2.1.2 PHOTON-Beetle-AEAD の仕様の概要

認証暗号 PHOTON-Beetle-AEAD はレート r をパラメータとし，PHOTON-Beetle-AEAD[r] と表される．$r = 32$ あるいは $r = 128$ であり，1 回の P_{256} の呼び出しで処理する入力のビット長を表している．$r = 128$ である PHOTON-Beetle-AEAD[128] が primary recommended version である.

PHOTON-Beetle-AEAD は Duplex Sponge [BDPA11] を改良した Beetle [CDNY18a] に基づいており，線形関数 ρ を利用することが特徴である．これにより，キャパシティの birthday bound を超える安全性を有する．線形関数 ρ は現在の内部状態 $S \in \{0, 1\}^r$ とビット列 $U \in \{0, 1\}^r$ を入力と

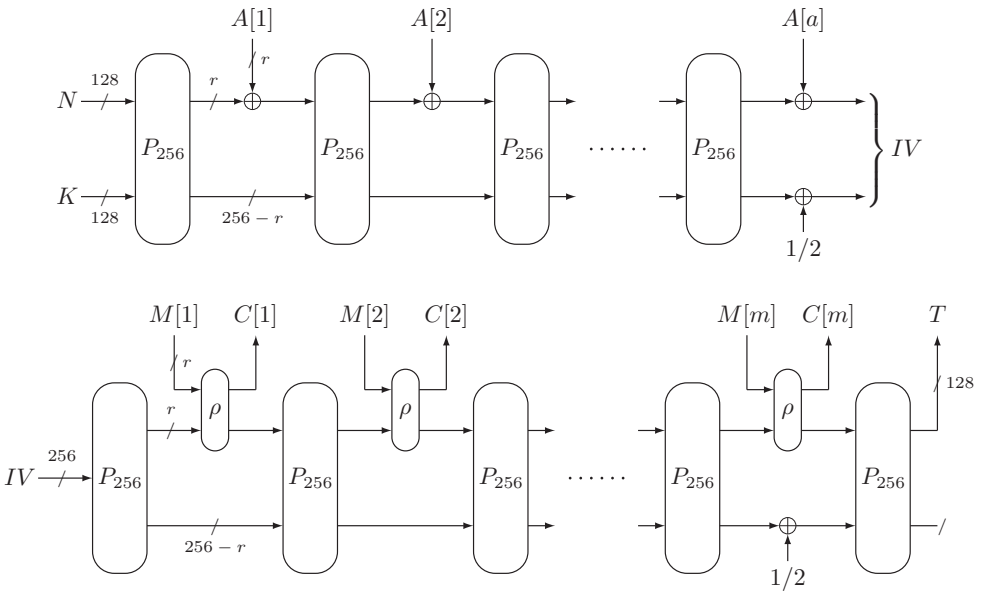

図 2.2: $|A|, |M| > 0$ の場合の PHOTON-Beetle-AEAD の暗号化．$A = (A[1] \| \cdots \| A[a])$，$M = (M[1] \| \cdots \| M[m])$，$a, m \geq 1$ である．

し，次の状態 $S \oplus U \in \{0,1\}^r$ とビット列 $\mathsf{Shuffle}(S) \oplus U \in \{0,1\}^r$ を出力する．すなわち，ρ は

$$\begin{aligned} \rho : \{0,1\}^r \times \{0,1\}^r &\to \{0,1\}^r \times \{0,1\}^r \\ (S, U) &\mapsto (S \oplus U, \mathsf{Shuffle}(S) \oplus U) \end{aligned}$$

と定義される．$\mathsf{Shuffle}(S)$ は S を $r/2$ ビットの S_1 と S_2 に $S_1 \| S_2 = S$ と分割し，$S_2 \| (S_1 \ggg 1)$ を出力する．$(S_1 \ggg 1)$ は S_1 の 1 ビット右巡回シフトである．

任意の $S \in \{0,1\}^r$ に対し，$\rho(S, \cdot)$ は $\{0,1\}^r$ 上の置換である．これを $\rho^{-1}(S, \cdot)$ と書き，$\rho^{-1}(S, V) = (S \oplus \mathsf{Shuffle}(S) \oplus V, \mathsf{Shuffle}(S) \oplus V)$ と定義される．なお [BCD+21] には $|U|, |V| < r$ の場合の ρ の動作が定義されているが，本報告書では省略する．

暗号化関数の入力は鍵 $K \in \{0,1\}^{128}$，ナンス $N \in \{0,1\}^{128}$，付加データ $A \in \{0,1\}^*$，平文 $M \in \{0,1\}^*$ であり，出力は暗号文 $C \in \{0,1\}^{|M|}$ とタグ $T \in \{0,1\}^{128}$ である．

復号関数は (K, N, A, C, T) を入力とし，平文 $M \in \{0,1\}^{|C|}$ か，あるいは reject を表す記号 $\perp$ を出力する．

暗号化関数を図 2.2–図 2.5に図示する．図 2.2は $|A|, |M| > 0$ の場合の動作を示している．$A = (A[1] \| \cdots \| A[a])$，$M = (M[1] \| \cdots \| M[m])$ と分割する．各 $A[i], M[j]$ は r ビットである．A が a ブロック，M が m ブロックの場合を図示しており，最終ブロック $A[a], M[m]$ が r ビットに満たない場合はパディングを施す．このパディングの有無により，A を処理した後の IV の直前の定数 XOR，M を処理した後の T 生成の P_{256} 呼び出し直前の定数 XOR を使い分ける．

図 2.3は $|A| = 0, |M| > 0$ の場合の動作を示しており，図 2.4は $|A| > 0, |M| = 0$ の場合，図 2.5は $|A| = |M| = 0$ の場合である．後述するように，図 2.5において，定数が鍵 K に直接 XOR されている点は注意を要する．

復号関数は (K, N, A, C, T) から，対応する M か，または reject を示す $\perp$ を出力する．IV の計算までは暗号化関数と同様であり，$C[1], \ldots, C[m]$ から $M[1], \ldots, M[m]$ とタグの候補である T' を ρ^{-1} を用いて計算し，$T = T'$ であれば $M[1], \ldots, M[m]$ を返す．そうでなければ $\perp$ を返す．

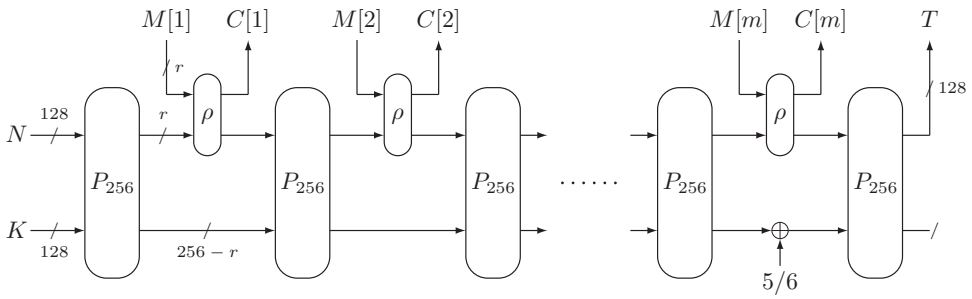

図 2.3: $|A| = 0, |M| > 0$ の場合の PHOTON-Beetle-AEAD の暗号化．$M = (M[1] \| \cdots \| M[m])$，$m \geq 1$ である．

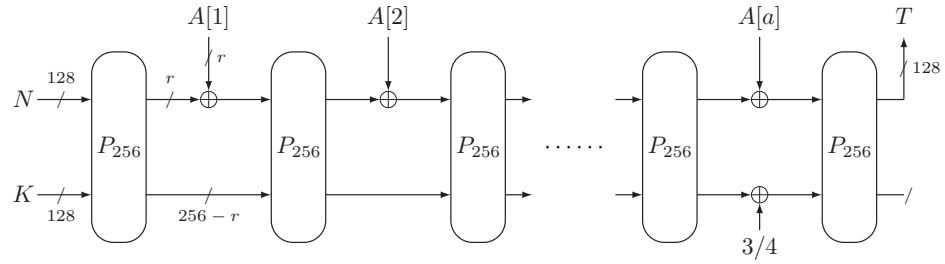

図 2.4: $|A| > 0, |M| = 0$ の場合の PHOTON-Beetle-AEAD の暗号化．$A = (A[1] \| \cdots \| A[a])$，$a \geq 1$ である．

2.1.3 PHOTON-Beetle-Hash の仕様の概要

ハッシュ関数 PHOTON-Beetle-Hash は $M \in \{0,1\}^*$ を入力とし，ハッシュ値 $T \in \{0,1\}^{256}$ を出力する．レート r は $r = 32$ の場合が唯一の推奨パラメータであり，これに沿って仕様をまとめる．まず入力 $M \in \{0,1\}^*$ を $|M| > 128$ の場合は

$$M = M[1] \| M[2] \| \cdots \| M[m]$$

と分割する．$|M[1]| = 128$，$|M[2]| = \cdots = |M[m-1]| = 32$，$|M[m]| \leq 32$ である．この分割された M に対し，図 2.6に従って動作し，ハッシュ値を計算する．$M[m]$ が 32 ビットに満たない場合にはパディングを使用し，パディングの有無が $M[m]$ 処理時に XOR する定数に反映される．$|M| \leq 128$ の場合は以下のように動作する．

- $|M| = 0$ であれば，$T = P_{256}(0^{255} \| 1)$ 出力する．
- $0 < |M| < 128$ であれば，$T = P_{256}(M \| 1 \| 0 \cdots 0 \| 1)$ 出力する．
- $|M| = 128$ であれば，$T = P_{256}(M \| 0 \cdots 0 \| 10)$ 出力する．

2.2 安全性調査

2.2.1 安全性の主張

設計者による PHOTON-Beetle-AEAD 並びに PHOTON-Beetle-Hash の安全性の主張は [BCD+21, Table 4.1, Table 4.2] に記載されており，これらを表 2.1，表 2.2に再掲する．表 2.1における IND-CPA

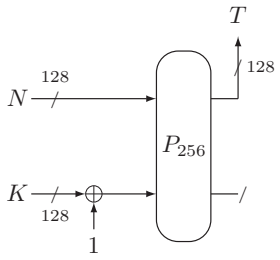

図 2.5: $|A| = |M| = 0$ の場合の PHOTON-Beetle-AEAD の暗号化.

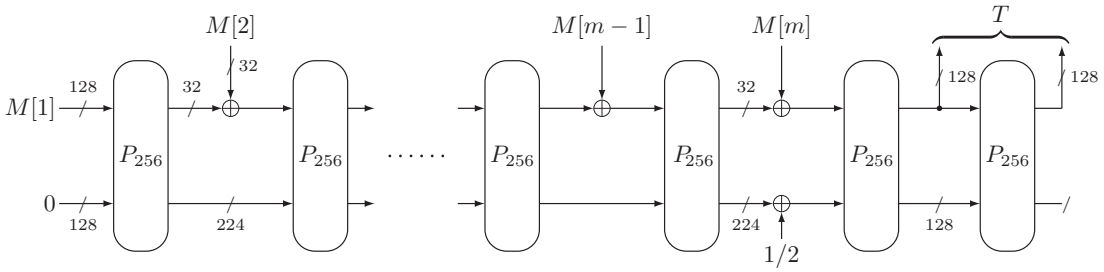

図 2.6: $|M| > 128$ の場合の PHOTON-Beetle-Hash の動作.

はナンスを繰り返さない敵に対する暗号化の安全性を表しており，INT-CTXT は暗号化オラクルに対してナンスを繰り返さない敵に対する認証の安全性を表している．表 2.2における衝突は衝突攻撃に対する安全性を，原像は原像攻撃に対する安全性を表している．単位は全てビットで表記されている．

2.2.2 P_{256} の安全性調査

暗号学的置換 P_{256} はハッシュ関数 PHOTON での利用のために [GPP11] で提案された．256 ビット入出力の P_{256} を含め，入出力長に応じ $P_{100}, P_{144}, P_{196}, P_{256}, P_{288}$ の 5 種類の暗号学的置換が定義されており，このうち PHOTON-Beetle では P_{256} を PHOTON_{256} と呼び，これを要素技術として用いる．これらの 5 種類の暗号学的置換の仕様段数は全て 12 ラウンドである．P_{256} の安全性解析は [BCD+21, Sect. 4.5] に記載されおり，以下その記載事項をまとめる．

- 設計者による解析 [GPP11] では，rebound 攻撃により，8 ラウンドの P_{256} とランダム置換とを時間計算量 2^{16}，メモリ計算量 2^8 で識別できることが示されている．
- [JNP13] では，上記の攻撃を改善し，時間計算量を 2^{16} から $2^{10.8}$ に削減できることが示されている．
- [JNP12] では，9 ラウンドの P_{256} とランダム置換とを時間計算量 2^{184}，メモリ計算量 2^{32} で識別できることが示されている．
- [CSCW17] では，統計的 integral 識別攻撃により，10 ラウンドの P_{256} とランダム置換とを時間計算量 $2^{96.59}$，メモリ計算量 $2^{70.46}$ で識別できることが示されている．
- [WGR18] では，仕様段数である 12 ラウンドの P_{256} に対し，サイズ 2^{184} の zero-sum 分割を用い，時間計算量 2^{183} の識別攻撃が示されている．

表 2.1: PHOTON-Beetle-AEAD の安全性 [BCD+21, Table 4.1].

モード	安全性モデル	データ計算量	時間計算量
PHOTON-Beetle-AEAD[128]	IND-CPA	121	121
PHOTON-Beetle-AEAD[128]	INT-CTXT	121	121
PHOTON-Beetle-AEAD[32]	IND-CPA	128	128
PHOTON-Beetle-AEAD[32]	INT-CTXT	128	128

表 2.2: PHOTON-Beetle-Hash の安全性 [BCD+21, Table 4.2].

モード	安全性モデル	時間計算量
PHOTON-Beetle-Hash[32]	衝突	112（データ計算量 $2^{111.5}$）
PHOTON-Beetle-Hash[32]	原像	128

ウェブサイト [GPP, Security] では [JNP12] の 9 ラウンド識別攻撃が最良の攻撃法とされている．また，[GPP11] の引用数は Google Scholar によれば 631 件であり，[BCD+21, Sect. 4.5] は 2021 年 5 月時点で書かれている．これ以降の [GPP11] を引用する発表文献を調査した結果，本報告書執筆時点で上記以外に挙げるべき解析結果は発表されていないことを確認した．

2.2.3 PHOTON-Beetle-AEAD の安全性調査

証明可能安全性

PHOTON-Beetle-AEAD は [CDNY18a] で提案された利用モードである Beetle に修正を施した方式である．NIST 提案文書 [BCD+21] において，PHOTON-Beetle-AEAD の証明可能安全性が主張されている．主張されている安全性バウンドは，暗号化に関する安全性（IND-CPA）が

$$O\left(\frac{\sigma^2}{2^{256}} + \frac{q_p}{2^{256-r}} + \frac{q \cdot q_p}{2^{256}} + \frac{rq_p}{2^{128}} + \frac{\sigma_e^r}{2^{128(r-1)}}\right) \tag{2.1}$$

である．ただし，σ は暗号化クエリの総ブロック数，q_p はオフラインクエリ回数，r はレート（$r = 32$ もしくは 128），q は暗号化クエリの回数，σ_e は暗号化クエリの総ブロック数である [BCD+21, Sect. 4.1][1]．認証に関する安全性（INT-CTXT）は

$$O\left(\frac{q_p(q+q')}{2^{256}} + \frac{rq_p}{2^{128}} + \frac{q_p^r}{2^{128(r-1)}} + \frac{r\sigma'}{2^{256-r}}\right) \tag{2.2}$$

である．ただし，q_p はオフラインクエリ回数，q は暗号化クエリの回数，q' は復号クエリの回数，r はレート（$r = 32$ もしくは 128），σ' は復号クエリの総ブロック数である [BCD+21, Sect. 4.2].

また PHOTON-Beetle-AEAD に関連するその他の証明可能安全性の結果として，[CDNY18a, CDNY18b, CJN20, CJN19] が挙げられる．[CDNY18a, CDNY18b] は PHOTON-Beetle-AEAD と方式の違いがあり，本報告書では考えない．

PHOTON-Beetle-AEAD に対し，敵 $\mathcal{A}$ の combined AE 識別利得を $\mathbf{Adv}^{\mathrm{ae}}_{\text{PHOTON-Beetle}}(\mathcal{A})$ と書く．これは，敵 $\mathcal{A}$ が

暗号化オラクル，復号オラクル，P_{256} オラクル，P_{256}^{-1} オラクル

[1] σ と σ_e の違いは不明である．

の 4 つ組と，

$$\$ \text{オラクル}，\perp \text{オラクル}，P_{256} \text{オラクル}，P_{256}^{-1} \text{オラクル}$$

の 4 つ組の識別に成功する確率である．ただし，$ オラクルは暗号化オラクルの出力と同じ長さの乱数を返すオラクル，⊥ オラクルは reject を表す ⊥ を返すオラクルであり，P_{256} はランダム置換，P_{256}^{-1} はその逆置換とモデル化される．

[CJN20] では前述の安全性バウンドを改良した結果が示されている．[CJN20] の Corollary 1 では，敵の攻撃成功確率が $r = 128$ の場合は高々

$$\begin{aligned}\mathbf{Adv}^{\mathrm{ae}}_{\text{PHOTON-Beetle}}(\mathcal{A}) \leq\ & \frac{4\tau\sigma_d}{2^c} + \frac{4r\sigma_d}{2^c} + \frac{4b\sigma_d}{2^c} + \frac{q_p}{2^\kappa} + \frac{2q_d}{2^\tau} + \frac{2\sigma_d(\sigma+q_p)}{2^b} \\ & + \frac{6\sigma_e q_p}{2^b} + \frac{8rq_p}{2^c} + \frac{4\tau q_p}{2^{b-\tau}} + \frac{\sigma_e+q_p}{2^b} + \frac{4rq_p\sigma_d}{2^{2c}} \end{aligned} \tag{2.3}$$

であることが示されている．ただし，τ はタグのビット長，c はキャパシティ，r はレート，$b = r + c$ であり，κ は鍵のビット長，q_e は暗号化クエリの回数，q_d は復号クエリの回数，σ_e は暗号化クエリの総ブロック数，σ_d は復号クエリの総ブロック数，q_p はオフラインクエリ数，$\sigma = \sigma_e + \sigma_d$ である．

攻撃手法（鍵回復攻撃）

Dobraunig と Mennink は，付加データと平文が空列の場合（図 2.5に対応）の PHOTON-Beetle-AEAD に対する鍵回復攻撃を示した [DM20]．$(q_e, q_p) \approx (2^{122.8}, 2^{124})$ で鍵回復に成功する．証明可能安全性の結果に矛盾しない攻撃であり，表 2.1はこの結果を反映した数字になっている．また，[DM20] を送信したメールでは下記のように述べられている．

> We point out that our observation does not seriously threaten PHOTON-Beetle and that the problem is easily resolved by updating the security claims.

攻撃手法（偽造攻撃，識別攻撃）

[IIM22] では PHOTON-Beetle-AEAD の証明可能安全性バウンドに関する解析がされており，以下その概要をまとめる．[IIM22] では式 (2.2) に矛盾する攻撃手法が 2 通り示されている．オフラインクエリ回数 q_p が 0 の場合，式 (2.2) は

$$O\left(\frac{r\sigma'}{2^{256-r}}\right) \tag{2.4}$$

と整理できる．この式は暗号化オラクルへのクエリ回数と独立であり，暗号化クエリの回数に関わらず安全性が保たれることを主張している．

また，$r = 32$ の場合，敵の偽造成功確率が $\sigma'/2^{128}$ よりも小さいことを主張している．しかし，どちらの主張も一般には成り立たない．

[IIM22] では $q = 2^{b/2}$ 回の暗号化オラクルへのクエリにより，256 ビットの内部状態全体が衝突し，これを利用して高い確率で偽造に成功することが示されている．また，PHOTON-Beetle-AEAD のタグ長は 128 ビットであり，単にランダムなタグを復号オラクルに q' 回クエリする敵を考えると，その成功確率は $q'/2^{128}$ である．これは，偽造成功確率が $\sigma'/2^{128}$ よりも小さいことと矛盾する．

ただし，どちらの攻撃も 2^{128} の計算量を必要とし，表 2.1にあるビット安全性の主張を破るものではない．

[IIM22] では [CJN20] の式 (2.3) についても解析している．$q_p = 0$ で，$q_d = \sigma_d = 0$ の場合（オフラインクエリと復号オラクルへのクエリをしない場合），式 (2.3) は

$$\mathbf{Adv}^{\mathrm{ae}}_{\text{PHOTON-Beetle}}(\mathcal{A}) \leq \frac{\sigma_e}{2^b}$$

と整理できる．すなわち，暗号化に関する IND-CPA 安全性が $\sigma_e = 2^b$ 回の暗号化オラクルへクエリする敵に対して保たれることを主張している．しかし，$q_e = 2^{b/2}$ 回の暗号化オラクルへのクエリにより 256 ビットの内部状態全体の衝突が期待でき，このとき IND-CPA 安全性を保つことはできない．したがって，式 (2.3) は一般に成り立たない．

なお式 (2.3) の問題点は，[CJN20] の ePrint 版である [CJN19] において解決されている．$r = 128$ の場合の [CJN19] に記載の安全性バウンドは

$$\begin{aligned}\mathbf{Adv}^{\mathrm{ae}}_{\text{PHOTON-Beetle}}(\mathcal{A}) \leq\ & \frac{8r\sigma_d}{2^c} + \frac{8b^3 q_p^2 \sigma_d}{2^{b+c}} + \frac{q_p}{2^\kappa} + \frac{2q_d}{2^r} + \frac{2\sigma(2\sigma + q_p)}{2^b} \\ & + \frac{q_p^2}{2^b} + \frac{6\sigma_e q_p}{2^b} + \frac{12 r q_p}{2^c} + \frac{\sigma_e + q_p}{2^b} + \frac{4 r q_p \sigma_d}{2^{2c}} \end{aligned} \tag{2.5}$$

である．この安全性バウンドは $\sigma^2/2^b$ の項が追加されており，暗号化クエリにおける内部状態全体の衝突が考慮された式になっている．したがって，前述の攻撃に用いた事象を含んだ安全性バウンドになっている．

なおこの問題は，[CJN20] にある Theorem 2 の証明の最後の部分で，各確率の項の和を取る際に $2\sigma_e^2/2^b$ の扱いを誤った計算ミスが原因だと思われる．

攻撃手法（関連鍵攻撃）

次に，[IIM22] で示されている PHOTON-Beetle-AEAD に対する関連鍵攻撃 [Bih93, BK03, Luc04] についてまとめる．一般に，暗号化オラクル $\mathcal{E}_K(\cdot,\cdot,\cdot)$ はナンス，付加データ，平文の組 (N, A, M) を入力とし，暗号文とタグの組 $(C, T) = \mathcal{E}_K(N, A, M)$ を返す．ここで考える関連鍵攻撃では，(N, A, M) に加えて $\Delta \in \{0,1\}^k$ を入力としてとる関連鍵暗号化オラクルを考える．k は鍵 K のビット長である．関連鍵暗号化オラクルはクエリ (Δ, N, A, M) に対し，$(C, T) = \mathcal{E}_{K\oplus\Delta}(N, A, M)$ を返す．このとき，以下のように効率的な偽造が可能である．

1. (Δ, N, A, M) を固定する．ただし，$\Delta = 1$，N は任意のナンス，A は空列，M は $|M| \geq r$ である任意の平文である．
2. 関連鍵暗号化オラクルに (Δ, N, A, M) をクエリし，対応する (C, T) を得る．$M[1]$ を M の先頭 r ビットとし，$C[1]$ を C の先頭 r ビットとする．
3. (N, A', C', T') を偽造文として出力する．ただし，A' と C' は空列であり，$T' = \mathsf{Shuffle}^{-1}(M[1] \oplus C[1])$ である．

図 2.7にこの攻撃を図示する．関連鍵 $K \oplus 1$ の関連鍵暗号化クエリは，付加データと平文が空列である場合の暗号化の処理手順をシミュレートしており，上記攻撃手順のステップ 3 は確率 1 で受理され，偽造に成功する．

なおこの関連鍵攻撃は付加データと平文が空列である場合のみであり，その適用範囲は極めて狭い．また，関連鍵攻撃に関する安全性は NIST 軽量暗号標準化プロジェクトの安全性要件ではないとともに，設計者によって主張もされていない．

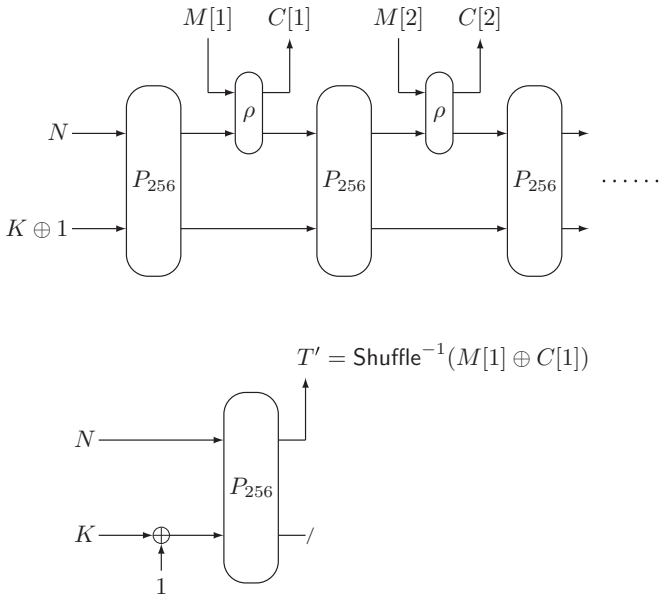

図 2.7: PHOTON-Beetle-AEAD に対する関連鍵攻撃による偽造.

攻撃手法（サイドチャネル攻撃）

[JP22] において，PHOTON-Beetle-AEAD に対する DFA（Differential Fault Attack）の解析がされている．ランダムフォールトモデルにおいて，$2^{37.15}$ 回のフォールトクエリにより鍵回復が可能であることが示されている．このモデルでは，内部状態のうちの一つのセルの値にランダムなフォールトを注入できるモデルを考えている．また，既知フォールトモデルを考えており，$2^{11.05}$ 回のフォールトクエリにより，鍵回復が可能であることが示されている．このモデルでは，内部状態のうちの一つのセルの値にランダムであるが値が既知のフォールトを注入できるモデルを考えている．

2.2.4 PHOTON-Beetle-Hash の安全性調査

PHOTON-Beetle-Hash の安全性は衝突攻撃については [BCD+21, Sect. 4.3] で，原像計算について [BCD+21, Sect. 4.4] で解析されている．q 回の P_{256} の計算をする敵が衝突を見つける確率が高々 $q^2/2^{256-r-1}$ であることと，原像を計算する確率が高々$q/2^{128}$ であることが述べられている．

Sponge 構造 [BDPA08] によりハッシュ関数を定義でき，ハッシュ関数 PHOTON [GPP11] はこれに沿っており，[BDPA08] にある強識別不可能性（indifferentiability [MRH04, CDMP05]）の安全性証明の結果を用いることができる．一方で PHOTON-Beetle-Hash はハッシュ関数 PHOTON や Sponge 構造に基づくハッシュ関数とは以下の点で異なっており，直接的には [BDPA08] の結果には含まれない．

- 入力 M を分割したビット列 $M[1]\|M[2]\|\cdots\|M[m]$ について，$M[1]$ とそれ以降の $M[2]\|\cdots\|M[m]$ とは長さが異なる．
- パディングの有無を定数 1/2 をキャパシティに XOR することで区別をしている．

なお，[LM22] において，PHOTON-Beetle-Hash の原像攻撃の安全性が 2^{128} であることが示されている．

攻撃手法

Mége は，[Még21] において，パディングの有無を示す定数 1/2 を利用し，データ計算量が $2^{111.5}$ の衝突攻撃が可能であることを指摘した．表 2.2はこの結果を反映した数字になっている．

2.3 安全性評価

2.3.1 P_{256} の安全性評価

ハッシュ関数 PHOTON は ISO/IEC 29192-5:2016 にて国際標準化されており，すでに多数の論文にて安全性が解析されている．2.2.2章で述べた通り，本報告書執筆時点ですでに記載した事項以外に挙げるべき解析結果は発表されていないことを確認した．また，報告者は，ハッシュ関数 PHOTON あるいは P_{256} の安全性に関して報告すべき懸念事項を発見していない．

2.3.2 PHOTON-Beetle-AEAD の安全性評価

[DM20] の攻撃はすでに表 2.1に反映されており，表 2.1にあるビット安全性の主張を覆す解析結果は知られていない．

NIST 提案文書 [BCD+21] に記載の証明可能安全性のバウンドは一般には成り立たず，[CDNY18a, CDNY18b] を改良したとする [CJN20] のバウンドも正確ではない．しかし，後者に関しては [CJN19] において修正がされている．

留意点として，式 (2.1) と式 (2.2) と，表 2.1の関連について記載する．式 (2.1) と式 (2.2) にはいずれも

$$O\left(\frac{rq_p}{2^{128}}\right) \tag{2.6}$$

という項がある．$r = 128$ の場合，これは

$$O\left(\frac{128q_p}{2^{128}}\right) = O\left(\frac{q_p}{2^{121}}\right)$$

を意味し，q_p はオフラインクエリの回数で時間計算量に対応する．表 2.1の $r = 128$ の場合の IND-CPA 並びに INT-CTXT の時間計算量はこれに相当する数字である 121 になっている．

一方で，$r = 32$ の場合，式 (2.6) は

$$O\left(\frac{32q_p}{2^{128}}\right) = O\left(\frac{q_p}{2^{123}}\right)$$

である．この式に従う限り，表 2.1の $r = 32$ の場合の IND-CPA 並びに INT-CTXT の時間計算量は 123 とすべきであり，これを 128 とする根拠は見受けられない．また，これを議論している NIST 提案文書 [BCD+21, Sect. 4.2] には

> Details of the security claim can be found in [3].

と記載されている（[3] は本報告書の [CJN19] である）が，[CJN19] と式 (2.1) と式 (2.2) では対象としている方式，考慮している安全性モデル，主張している安全性バウンドが異なっていることを付記する．

以上から，表 2.1に関して，次のように結論する．

- 表 2.1にあるビット安全性の主張を覆す解析結果は知られていない．
- 表 2.1にあるビット安全性に関わる理論的解析の議論には不備がある．また，一部（$r = 32$ の場合の IND-CPA 並びに INT-CTXT の時間計算量）は理論的根拠がない数字が挙げられている．

関連鍵攻撃は限定的な攻撃シナリオでのみ成立する攻撃であり，またその適用範囲も極めて限定的である．攻撃シナリオが成立しないような使用，実装とすれば問題にはならない．注意すべき事例として，付加データと平文を空列とし，適当なナンスに対するタグを KCV（Key Check Value）とするような使用が挙げられる．KCV は鍵の ID としたり，あるいは整合性のチェックのために用いる値であり，鍵管理用のデータとして，例えば [GSM16, Glo18] を含め，様々な場面で用いられる．

[JP22] にあるサイドチャネル攻撃は，実装面での対策が有効であると考えられる．

2.3.3 PHOTON-Beetle-Hash の安全性評価

表 2.2にあるビット安全性の主張を覆す解析結果は知られていない．また，[BCD+21, Sect. 4.4] にある原像を計算する確率が高々$q/2^{128}$ であることの議論に不備は見受けられず，[LM22] でも同様の安全性が述べられている．一方で，[BCD+21, Sect. 4.3] にある衝突攻撃の成功確率が高々$q^2/2^{256-r-1}$ であることの議論は，可能性のある衝突攻撃シナリオのうち，一つを取り上げてその成功確率の上界を述べているのみであり，これを以って衝突攻撃の成功確率が一般的に $q^2/2^{256-r-1}$ で抑えられるという保証は与えていない．

しかし，ここで主張しているバウンド $q^2/2^{256-r-1}$ はキャパシティの birthday bound に相当し，これは [BDPA08] で示されている強識別不可能性の安全性バウンドに一致する．2.2.4章で述べた通り [BDPA08] の結果は直接的には PHOTON-Beetle-Hash を含んでいないが，これと同等の安全性は期待できると思われる．

また，パディングの有無を XOR する定数によって区別する手法は [Hir18] において解析されており，問題とはならないと考えられる．

以上を鑑みて，表 2.2に記載のビット安全性は本報告書執筆時点で問題ないと考えられる．なお，原像計算に関する安全性の主張は，強識別不可能性が示す安全性バウンドよりも強い主張をしていることを付記する．

Chapter 3

Sparkleの安全性に関する調査及び評価

Sparkle は Christof Beierle, Alex Biryukov, Luan Cardoso dos Santos, Johann Großschädl, Léo Perrin, Aleksei Udovenko, Vesselin Velichkov, Qingju Wang により設計された ARX に基づく暗号学的置換の族である [BBdS+20b, BBdS+21]．入出力サイズに応じて，Sparkle256，Sparkle384，Sparkle512 の 3 通りの暗号学的置換が定義されている．これらはブロック暗号 Sparx [DPU+16] の設計に基づいており，Sparx の鍵を固定し，入出力長を大きくした置換に対応する．Schwaemm は Sparkle を暗号学的置換として用いた Sponge 構造に基づく AEAD であり，Esch は Sparkle を暗号学的置換として用いた Sponge 構造に基づくハッシュ関数である．

本章では，Sparkle，Schwaemm，Esch の方式概要，安全性調査，安全性評価について述べる．なお [BBdS+21] では XOF（eXtendable-Output Function）も定義されているが，本報告書では扱わない．

3.1 方式概要

本章では，Sparkle，Schwaemm，Esch の仕様の概要をまとめる．より詳細な仕様は [BBdS+21] に記載されている．

3.1.1 Sparkle の仕様の概要

Sparkle は入出力長により Sparkle256，Sparkle384，Sparkle512 の 3 通りが定義されており，それぞれ入出力が 256 ビット，384 ビット，512 ビットの暗号学的置換である．

いずれもブロック長 64 ビット，鍵長 32 ビットのブロック暗号 Alzette [BBdS+20a] を構成要素として用いる．定数（鍵）$c \in \{0,1\}^{32}$ に対し，Alzette は 64 ビット上の置換であり，これを A_c と書く．A_c は ARX（Addition-Rotation-XOR）による Feistel 構造であり，これを 4 ラウンドと定義している．この動作を図 3.1に示す．入力は $(x, y) \in \{0,1\}^{32} \times \{0,1\}^{32}$，出力は $(u, v) \in \{0,1\}^{32} \times \{0,1\}^{32}$ である．$\boxplus$ は算術加算，$\ggg i$ は i ビット巡回シフトを表す．

Sparkle256，Sparkle384，Sparkle512 の動作を図 3.2，図 3.3，図 3.4にそれぞれ示す．鍵を定数とした Alzette を S-box とした SPN 構造であり，Alzette の鍵として使用する定数 $c_0, \ldots, c_7$ は入出力

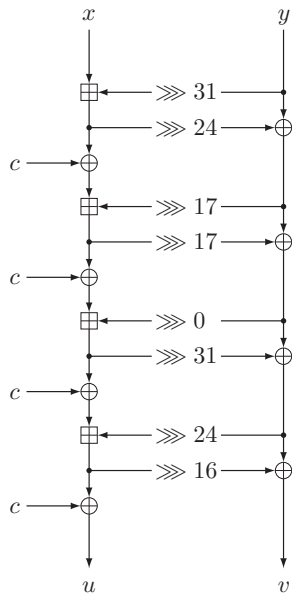

図 3.1: Alzette の動作．$A_c : (x, y) \mapsto (u, v)$，$c \in \{0,1\}^{32}$ は定数（鍵）であり，入力は $(x, y) \in \{0,1\}^{32} \times \{0,1\}^{32}$，出力は $(u, v) \in \{0,1\}^{32} \times \{0,1\}^{32}$ である．

サイズによらず，16 進数表記で

$$\begin{cases} c_0 = \texttt{0xB7E15162} \\ c_1 = \texttt{0xBF715880} \\ c_2 = \texttt{0x38B4DA56} \\ c_3 = \texttt{0x324E7738} \\ c_4 = \texttt{0xBB1185EB} \\ c_5 = \texttt{0x4F7C7B57} \\ c_6 = \texttt{0xCFBFA1C8} \\ c_7 = \texttt{0xC2B3293D} \end{cases} \tag{3.1}$$

と定義されている．各 z_i は 64 ビットであり，Sparkle256，Sparkle384，Sparkle512 の入出力はそれぞれ $(z_0, \ldots, z_3) \in \{0,1\}^{256}$，$(z_0, \ldots, z_5) \in \{0,1\}^{384}$，$(z_0, \ldots, z_8) \in \{0,1\}^{512}$ である．Sparkle では各繰り返し処理をステップと呼んでおり，各ステップは定数 XOR，S-box 層，線形層からなる．図 3.2，図 3.3，図 3.4では 4 ステップのみ図示している．

- ステップ s において（$s = 0, 1, \ldots$），定数 XOR は $c_{s \bmod 8}$ と $s \bmod 8$ をそれぞれ 64 ビット定数と見なし，これらを z_0 と z_1 に XOR する．より詳細には，$0^{32} \,\|\, c_{s \bmod 8}$ と $s \bmod 8$ の 64 ビット 2 進数表現を XOR する．
- S-box 層は Alzette を S-box として用い，これを並列に適用する．使用する定数は式 (3.1) に従う．
- 線形層は Feistel 構造に基づいており，64 ビット単位の置換と XOR，ℓ' からなる．図 3.2，図 3.3，図 3.4の ℓ' は入出力 64 ビットの線形変換であり，次のように定義される．入力 $t \in \{0,1\}^{64}$

表 3.1: Sparkle256，Sparkle384，Sparkle512 のステップ数.

方式	n	slim のステップ数	big のステップ数
Sparkle256	256	7	10
Sparkle384	384	7	11
Sparkle512	512	8	12

表 3.2: Schwaemm のパラメータと安全性.

方式	n	r	c	$\|K\|$	$\|N\|$	$\|T\|$	安全性	データ制限（バイト）
Schwaemm256-128	384	256	128	128	256	128	120	2^{68}
Schwaemm192-192	384	192	192	192	192	192	184	2^{68}
Schwaemm128-128	256	128	128	128	128	128	120	2^{68}
Schwaemm256-256	512	256	256	256	256	256	248	2^{133}

を $t = t_0 \,\|\, t_1 \,\|\, t_2 \,\|\, t_3$，$t_0, \ldots, t_3 \in \{0,1\}^{16}$ と分割する．ℓ' の出力は次の式で定義される．

$$\ell'(t) = t_3 \,\|\, t_3 \oplus t_2 \,\|\, t_1 \,\|\, t_1 \oplus t_0 \tag{3.2}$$

Sparkle256，Sparkle384，Sparkle512 はいずれも任意のステップ数に対して定義されているが，[BBdS+21] では slim instance と big instance が定義されており，それぞれ表 3.1にあるステップ数として定義される．

後述の Schwaemm，Esch では，表 3.1にあるステップ数の Sparkle を用いる．それぞれ，Sparkle256_7，Sparkle256_{10}，Sparkle384_7，Sparkle384_{11}，Sparkle512_8，Sparkle512_{12} と書く．ステップ数が 8 より大きい場合には，各ステップの始めの定数 XOR は $c_{s \bmod 8}$ と $s \bmod 8$ を用いる．

3.1.2 Schwaemm の仕様の概要

Schwaemm はナンスベースの認証暗号であり，4 通りのパラメータ Schwaemm128-128，Schwaemm256-128，Schwaemm192-192，Schwaemm256-256 が定義されている．Schwaemm256-128 が primal member である．Schwaemm$r$-c はレートが r ビット，キャパシティが c ビットの Sponge 構造に基づく認証暗号であり，入出力長 $n = c + r$ ビットの Sparkle を暗号学的置換として用いる．表 3.2にパラメータと安全性の主張をまとめる．表 3.2において，n は Sparkle の入出力長，r はレート，c はキャパシティ，$|K|$ は鍵長，$|N|$ はナンス長，$|T|$ はタグ長をビットで表している．安全性はビットで表現されており，データ制限はバイトである．

Schwaemm の暗号化関数の入力は鍵 $K \in \{0,1\}^c$，ナンス $N \in \{0,1\}^r$，付加データ $A \in \{0,1\}^*$，平文 $M \in \{0,1\}^*$ であり，出力は暗号文 $C \in \{0,1\}^{|M|}$ とタグ $T \in \{0,1\}^c$ である．

復号関数は (K, N, A, C, T) を入力とし，平文 $M \in \{0,1\}^{|C|}$ か，あるいは reject を表す記号 $\perp$ を出力する．

図 3.5に primal member である Schwaemm256-128 を示す．また，図 3.6に Schwaemm$r$-c，$r = c$ を示す．これは Schwaemm128-128，Schwaemm192-192，Schwaemm256-256 に対応する．いずれも全体構造としては PHOTON-Beetle-AEAD との共通点が多く，これにいくつかの修正を施した方式である．

表 3.3: Esch のパラメータと安全性.

方式	n	r	c	衝突	第 2 原像	原像	データ制限（バイト）
Esch256	384	128	256	128	128	128	2^{132}
Esch384	512	128	384	192	192	192	2^{196}

図 3.5の Schwaemm256-128 では，データ処理に Sparkle384_7 を用い，初期処理，定義域分離，タグ生成に Sparkle384_{11} を用いる．Sparkle384_{11} は図 3.5においてグレーで示している．また，図 3.6の Schwaemm$r$-c，$r = c$ の場合はデータ処理に Sparklen_{slim} を用い，初期処理，定義域分離，タグ生成に Sparklen_{big} を用いる．ここで，$n = r + c$ である．Sparklen_{big} は図 3.6においてグレーで示している．

- Schwaemm では，ρ の定義が PHOTON-Beetle-AEAD とは異なる．入力 $(S, D) \in \{0,1\}^r \times \{0,1\}^r$ に対し，

$$\begin{aligned} \rho : \{0,1\}^r \times \{0,1\}^r &\to \{0,1\}^r \times \{0,1\}^r \\ (S, D) &\mapsto (S' \oplus D, S \oplus D) \end{aligned}$$

  と定義される．ただし，$S' = S_2 \,\|\, (S_2 \oplus S_1)$，$S = S_1 \,\|\, S_2$，$S_1, S_2 \in \{0,1\}^{r/2}$ である．任意の $S \in \{0,1\}^r$ に対し $\rho(S, \cdot)$ は $\{0,1\}^r$ 上の置換であり，これを ρ^{-1} と書く．

  なお Schwaemm で利用する ρ は，PHOTON-Beetle-AEAD の基となった Beetle [CDNY18a, CDNY18b] と同一である．

- 図 3.5の Schwaemm256-128 では，c ビットのキャパシティから r ビットのレート部分に対して，$\mathcal{W}_{c,r}$ を用いて XOR を行う．$c = 128$，$r = 256$ であり，入力 $S \in \{0,1\}^r$ に対し，$\mathcal{W}_{c,r}$ は $S \,\|\, S \in \{0,1\}^c$ を出力する．$c = r$ の場合，$\mathcal{W}_{c,r}$ は恒等関数と定義され，図 3.6では $\mathcal{W}_{c,r}$ を記載していない．

- Const_A と Const_M は定数であり，付加データ A と平文 M を r ビットに分割したビット列 $A = A[1] \,\|\, \cdots \,\|\, A[a]$ と $M = M[1] \,\|\, \cdots \,\|\, M[m]$ の最終ブロック $A[a]$，$M[m]$ が r ビットに満たない場合はパディングを用い，これの有無により定数を使い分ける．

- タグ生成の直前に K の XOR がある点が PHOTON-Beetle-AEAD とは異なる．

復号関数は (K, N, A, C, T) から，対応する M か，または reject を示す $\perp$ を出力する．$C[1], \ldots, C[m]$ から $M[1], \ldots, M[m]$ とタグの候補である T' を ρ^{-1} を用いて計算し，$T = T'$ であれば $M[1], \ldots, M[m]$ を返す．そうでなければ $\perp$ を返す．

3.1.3 Esch の仕様の概要

ハッシュ関数 Esch は Esch256 と Esch384 の 2 通りが定義されており，$M \in \{0,1\}^*$ を入力とし，Esch256 はハッシュ値 $T \in \{0,1\}^{256}$ を出力し，Esch384 はハッシュ値 $T \in \{0,1\}^{384}$ を出力する．Esch256 は Sparkle384_7 と Sparkle384_{11} を用い，Esch384 では Sparkle512_8 と Sparkle512_{12} を用いる．レート r はいずれも $r = 128$ である．表 3.3にパラメータと安全性の主張をまとめる．表 3.3において，n は Sparkle の入出力長，r はレート，c はキャパシティをビットで表している．安全性（衝突，第 2 原像，原像）もビット単位で表現されており，データ制限のみバイト単位である．

まず入力 $M \in \{0,1\}^*$ を

$$M = M[1] \parallel M[2] \parallel \cdots \parallel M[m]$$

と分割する．$|M[1]| = \cdots = |M[m-1]| = 128$，$|M[m]| \leq 128$ である．この分割された M に対し，Esch256 は図 3.7に従って動作し，ハッシュ値を計算する．出力は $T = D[1] \parallel D[2] \in \{0,1\}^{256}$ である．Esch384 は図 3.8に従って動作し，$T = D[1] \parallel D[2] \parallel D[3] \in \{0,1\}^{384}$ を出力する．いずれの場合も，最終ブロックである $M[m]$ が 128 ビットに満たない場合にはパディングを使用し，パディングの有無が $M[m]$ 処理時に XOR する定数 c_M に反映される．

- 図 3.7の Esch256 では，パディングをする場合は $c_M = 0^{191}1 \in \{0,1\}^{192}$ であり，パディングがない場合は $c_M = 0^{190}10 \in \{0,1\}^{192}$ である．$\mathcal{M}_3$ は Sparkle384 の線形層の一部に相当し，入力 $M \parallel M' \parallel 0^{64} \in \{0,1\}^{192}$ に対し，以下のように 192 ビットの出力を計算する．

$$\begin{aligned} \mathcal{M}_3 : \{0,1\}^{192} &\to \{0,1\}^{192} \\ M \parallel M' \parallel 0^{64} &\mapsto M \oplus S \parallel M' \oplus S \parallel S \end{aligned}$$

  ただし，$M, M', S \in \{0,1\}^{64}$ である．また，$S = \ell'(M \oplus M' \oplus 0^{64})$ であり，$\ell' : \{0,1\}^{64} \to \{0,1\}^{64}$ は式 (3.2) で定義される．

- 図 3.8の Esch384 では，パディングがある場合は $c_M = 0^{255}1 \in \{0,1\}^{256}$，パディングがない場合は $c_M = 0^{254}10 \in \{0,1\}^{256}$ である．$\mathcal{M}_4$ は Sparkle512 の線形層の一部に相当する．入力 $M \parallel M' \parallel 0^{64} \parallel 0^{64} \in \{0,1\}^{256}$ に対し，

$$\begin{aligned} \mathcal{M}_4 : \{0,1\}^{256} &\to \{0,1\}^{256} \\ M \parallel M' \parallel 0^{64} \parallel 0^{64} &\mapsto M \oplus S \parallel M' \oplus S \parallel S \parallel S \end{aligned}$$

  と定義される．ただし，$M, M', S \in \{0,1\}^{64}$，$S = \ell'(M \oplus M' \oplus 0^{64} \oplus 0^{64})$ である．$\ell' : \{0,1\}^{64} \to \{0,1\}^{64}$ は式 (3.2) で定義される．

3.2 安全性調査

3.2.1 安全性の主張

NIST 提案文書 [BBdS+21] に従い，Sparkle，Schwaemm，Esch の安全性の主張をまとめる．

3.2.2 Sparkle の安全性の主張

Sparkle の big instances の安全性の主張は，入出力長 n ビットに対し，時間計算量とデータ計算量が $2^{n/2}$ を下回る識別攻撃が存在しないことである．big instances では入出力全体を敵が制御することを想定しているのに対し，slim instances は Sponge 構造に組み込んだ場合の安全性のみを想定している．Sponge 構造のレート部分に対応する入力のみを制御できる敵に対する識別不可能性を主張している．

表 3.4: Sparkle の安全性解析結果 [BBdS+21, Table 4.1]（抜粋）．各数字は攻撃可能ステップ数を表している．ここでは，入出力サイズ n ビットに対し，計算量 $2^{n/2}$ までの攻撃を考えている．最終 2 行は slim instances と big instances のステップ数を書いている．

攻撃手法	Sparkle256	Sparkle384	Sparkle512
差分攻撃	4	5	6
線形攻撃	5	6	6
ブーメラン攻撃	3	4	5
truncated 差分攻撃	2	2	3
Yoyo game	4	4	4
不可能差分攻撃	4	4	4
Zero-correlation 攻撃	4	4	4
Integral 攻撃，division property	4	4	4
slim のステップ数	7	7	8
big のステップ数	10	11	12

3.2.3 Schwaemm の安全性の主張

Schwaemm の安全性の主張は表 3.2の通りである．これは暗号化に関する安全性と認証に関する安全性を両方含んでいる．なお，暗号化オラクルに対してナンスを繰り返さない敵のみを考えている．これは（PHOTON-Beetle-AEAD ではなく）Beetle の証明可能安全性のバウンド [CDNY18a] が

$$\min\left\{r, \frac{r+c}{2}, c - \log_2(r)\right\} \tag{3.3}$$

ビットの安全性を保証していることを根拠としている．ここで，r はレート，c はキャパシティである．

NIST 提案文書 [BBdS+21, Sect. 4.1.2] では，ナンスが繰り返す場合の安全性について言及している．この場合であっても，内部状態全体を回復する攻撃，並びに鍵回復攻撃は現実的な計算量では不可能であることを主張している．また，ナンスが繰り返す場合でも付加データが繰り返さなければ，暗号化に関する安全性，認証に関する安全性が保たれることを主張している．

なお，これらの主張は正式な安全性の主張ではない旨が記載されており，設計者はナンスが繰り返さない状況での利用を強く推奨している．

3.2.4 Esch の安全性の主張

Esch の安全性の主張は表 3.3の通りである．キャパシティ c ビットに対し，衝突，原像計算，第 2 原像計算いずれも $c/2$ ビットの安全性を主張している．データ制限は $2^{c/2}$ ブロックに相当する．

3.2.5 Sparkle の安全性調査

[BBdS+21, BBdS+20b] において，広範な攻撃手法に対する安全性解析がなされている．[BBdS+21, Table 4.1] にその結果がまとめられており，これを表 3.4に再掲する．なお Sparkle の構成要素である Alzette は [BBdS+20a] で提案され，安全性解析がされている．

[BBdS+21, Table 4.1] は 2021 年 5 月時点で書かれており，2022 年 9 月の文書 [BBdS+22] にて，第三者による解析の進展が議論されている．以下，その論点をまとめる．

- [SS22] において，Sparkle に対する guess-and-determine 識別攻撃が示されている．4 ステップの Sparkle256，4 ステップの Sparkle384，5 ステップの Sparkle512 が現実的な計算量で攻撃できる．

- [HXW22] において，Alzette に対する差分攻撃と線形攻撃のバウンドを改善した結果が示されている．これらは [HW19, HW20] にて提案されたツールを用いて得られた結果であり，設計者による解析を改善し，Alzette がより安全であることを示している．また，[HXW22] では Alzette の rotational-XOR differential 確率についても解析している．

- [LSL21, NSLL22, LNS+23] において，Alzette に対する rotational differential-linear 攻撃の解析が示されている．[NSLL22] では，Alzette に対し，correlation が $2^{-5.57}$（理論値．計算機実験では $2^{-3.14}$）である rotational differential-linear 識別攻撃が示されており，Alzette を 2 回繰り返した方式に対し，correlation が $-2^{-8.24}$（理論値．計算機実験では $-2^{-5.50}$）である rotational differential-linear 識別攻撃が示されている．

- [XLJ+22] では，ARX の差分特性確率の計算について，マルコフ性の仮定に基づかないより精密な計算法を示しており，Alzette に適用した結果が示されている．

- [Spe22] では，1 ラウンドの Sparkle を鍵付きハッシュ関数として利用した際の差分衝突攻撃に関する安全性を解析している．

なお Alzette は Sparkle において S-box として機能しており，Alzette への識別攻撃が Sparkle の安全性を直ちに脅かすものではない．また，Sparkle の解析結果は計算量の面では改善がみられるが，攻撃可能ラウンド数（ステップ数）の面では，更新がない．

[BBdS+21] を引用する発表文献を調査した結果，本報告書執筆時点で上記以外に挙げるべき解析結果は発表されていないことを確認した．

3.2.6 Schwaemm の安全性調査

Schwaemm は PHOTON-Beetle-AEAD ではなく，Beetle [CDNY18a, CDNY18b] に基づき設計されており，式 (3.3) にある安全性定理を安全性の根拠としている．

- [BBdS+21, Sect. 3.1] において，設計指針について議論されており，[BBdS+21, Sect. 4.3] において攻撃者の視点からの Schwaemm の安全性解析が議論がされている．差分攻撃，線形攻撃，不可能差分攻撃，Zero-correlation 攻撃の適用ができないことが議論されている．

- ステップ数を削減した Schwaemm を使用した場合の guess-and-determine 攻撃が [BBdS+21, Sect. 4.4] で示されている．その結果を表 3.5にまとめる．

- [JK22] では Gorver のアルゴリズム [Gro96] を用いる場合の鍵全数探索にかかる量子回路のサイズの見積もりが示されている．

3.2.7 Esch の安全性調査

Esch は Sparkle を暗号学的置換として用いた Sponge 構造に基づくハッシュ関数である．

[BBdS+21, Sect. 3.1] において，設計指針について議論されており，[Hir18] においてキャパシティの birthday bound の証明可能安全性が示されており，この結果に基づいていること述べられている．また，データ入力の際に $\mathcal{M}_3$ や $\mathcal{M}_4$ を用いる手法を indirect injection と呼び，これの合理性について説明している．

表 3.5: Schwaemm に対する guess-and-determine 攻撃の結果 [BBdS+21, Table 4.9]（抜粋）．ϵ は任意の正のパラメータであり，ホワイトニングはキャパシティからレートへの XOR の有無を，0.5 ステップは追加の Alzette 層を意味する．

方式	ステップ数	ホワイトニング	時間計算量	データ計算量
Schwaemm128-128	3.5	no	2^{64}	2^{64}
Schwaemm192-192	3.5	no	2^{128}	2^{64}
Schwaemm256-256	3.5	no	2^{192}	2^{64}
Schwaemm256-256	3.5	no	2^{192}	1
Schwaemm128-128	4.5	no	$2^{96+\epsilon}$	$2^{96-\epsilon}$
Schwaemm192-192	4.5	no	$2^{128+\epsilon}$	$2^{128-\epsilon}$
Schwaemm256-256	4.5	no	$2^{192}+2^{160+\epsilon}$	$2^{160-\epsilon}$
Schwaemm256-256	3.5	yes	$2^{224+\epsilon}$	$2^{224-\epsilon}$

また，[BBdS+21, Sect. 4.3] において攻撃者の視点からの Esch の安全性解析が議論がされている．差分攻撃，不可能差分攻撃，Zero-correlation 攻撃の適用ができないことが議論されている．

3.3 安全性評価

3.3.1 Sparkle の安全性評価

3.2.5章で述べた通り，本報告書執筆時点ですでに記載した事項以外に挙げるべき解析結果は発表されていないことを確認した．また，報告者は，Sparkle あるいは Alzette の安全性に関して報告すべき懸念事項を発見していない．

3.3.2 Schwaemm の安全性評価

Schwaemm は PHOTON-Beetle-AEAD ではなく，Beetle [CDNY18a, CDNY18b] に基づき設計されており，その安全性定理を安全性の根拠とし，安全性定理に影響しないよう最適化を目指した設計である．Schwaemm と PHOTON-Beetle-AEAD とでは以下の違いがある．

- Beetle では単一の暗号学的置換を用いるが，Schwaemm では slim instance と big instance の 2 種類の暗号学的置換を用いる．
- キャパシティからレートへの XOR がある．Schwaemm ではこれをレートホワイトニングと呼んでいる．
- Schwaemm で利用する ρ は Beetle [CDNY18a, CDNY18b] と同一であり，PHOTON-Beetle-AEAD とは異なる．
- Schwaemm では，タグ生成の直前に K の XOR がある．これにより，付加データと平文が空列の場合に，PHOTON-Beetle-AEAD のように鍵に定数を XOR する処理がない．なお Beetle では付加データを空列とすることも平文を空列とすることも仕様上できない．

以下，上記の各点について述べる．

- 2 種類の暗号学的置換を用いる点について，Schwaemm の slim instance と big instance が 2 つの独立なランダム置換との識別不可能性を有するかの議論は十分にされていないと考えられる．一方で，[CDNY18a, CDNY18b] にある Beetle の安全性証明は単一のランダム置換を用いており，slim instance と big instance が 2 つの独立なランダム置換との識別不可能性を有するかどうかに関わらず，現実的に Beetle よりも安全性が低下する要因とはならないと考えられる．したがって，式 (3.3) が正しいと仮定するならば，slim instance と big instance を使い分けることは，安全性を低下させる要因とはならないと考えられる．

- キャパシティからレートへの XOR がある点について，証明可能安全性の観点からは $\mathcal{W}_{c,r}$ を含めて暗号学的置換と見れば，これが安全性を低下させる要因とはならない．

- ρ の定義について，PHOTON-Beetle-AEAD とは定義が異なっており，Schwaemm では Beetle [CDNY18a, CDNY18b] と同一の ρ を用いている．Beetle の安全性を示す式 (3.3) はこの ρ についての解析である．[CDNY18a, Sect. 3.2.1] において，ρ に必要な条件として，

  – 写像 $I \mapsto \texttt{Shuffle}(I)$ が全単射である
  – 写像 $I \mapsto \texttt{Shuffle}(I) \oplus I$ が全単射である

  の 2 点を挙げており，Beetle 並びに Schwaemm で使用する `Shuffle` はこの定義を満たす．なお $\texttt{Shuffle} : \{0,1\}^r \to \{0,1\}^r$ は，$\texttt{Shuffle}(I) = I_2 \,\|\, (I_2 \oplus I_1)$，$I = I_1 \,\|\, I_2$，$I_1, I_2 \in \{0,1\}^{r/2}$ と定義される．

  この条件は [CDNY18a, CDNY18b] にある安全性証明に必要であり，パディングがない場合に必要な条件を記述している．一方で，平文の最終ブロックにおいては ρ を適用する際に出力の切り捨てが生じる場合がある．安全性証明では最終ブロックが r ビットに満たない場合を考慮しておらず，上記の 2 条件のみで安全性証明が完結するか不明であり，式 (3.3) が一般的に成立するか，精査が必要な箇所であると考えられる．

- タグ生成の直前に K の XOR がある点について，証明可能安全性の観点からは影響がない．K を保持する記憶領域が必要であるが，PHOTON-Beetle-AEAD では空列の入力の処理が関連鍵攻撃に繋がったのに対し，この問題を回避している．

3.3.3 Esch の安全性評価

Esch は Sparkle を暗号学的置換として用いた Sponge 構造に基づくハッシュ関数である．一般的な Sponge 構造のハッシュ関数 [BDPA08] とでは，以下の違いがある．

- Esch では slim instance と big instance の 2 種類の暗号学的置換を用いる．

- データをレートに入力する際に線形変換 $\mathcal{M}_3$ ないしは $\mathcal{M}_4$ を用いる．

- パディングの有無を，XOR する定数によって区別する．

以下，上記の各点について述べる．

- 2 種類の暗号学的置換を用いる点について，Schwaemm と同様に，slim instance と big instance が 2 つの独立なランダム置換との識別不可能性を有するかの議論は十分にされていないと考えられるが，Sponge 構造の強識別不可能性は単一の暗号学的置換で示されており [BDPA08]，slim instance と big instance を使うことが安全性を低下させる要因とはならないと考えられる．

- データをレートに入力する際に線形変換 $\mathcal{M}_3$ ないしは $\mathcal{M}_4$ を用いる点について，次のように評価する．
  - この点は [BBdS+21, Sect. 3.1.3.2] において議論されており，$\mathcal{M}_3$ ないしは $\mathcal{M}_4$ の線形性から，$\mathcal{M}_3^{-1}$ ないしは $\mathcal{M}_4^{-1}$ を暗号学的置換のレート部分の入力に，$\mathcal{M}_3$ ないしは $\mathcal{M}_4$ を暗号学的置換のレート部分の出力に追加し，$(\mathcal{M}_3, \mathcal{M}_3^{-1})$ あるいは $(\mathcal{M}_4, \mathcal{M}_4^{-1})$ を入出力に加えた暗号学的置換を改めて暗号学的置換と考え，これを理想化するモデルを考えることで，通常の Sponge 構造のモデル化と同等の構造となり，証明可能安全性の観点からは問題とはならない．
  - パディングの有無を区別する定数 c_M が入力を制御することによってキャンセルされないことが重要である．PHOTON-Beetle-Hash ではこの定数はキャパシティに XOR されており，入力を制御してもキャンセルできないことが明らかである．Esch ではこのことは仕様からは直ちに明らかではないが，[BBdS+21, Sect. 3.1.3.2] において $\mathcal{M}_3$ と $\mathcal{M}_4$ の分岐数を用いて解析されており，問題とはならないと結論できる．
- PHOTON-Beetle-Hash 同様，パディングの有無を XOR する定数によって区別する手法は [Hir18] において解析されており，問題とはならないと考えられる．

  ただし，証明可能安全性の観点からは，前項のように $(\mathcal{M}_3, \mathcal{M}_3^{-1})$ あるいは $(\mathcal{M}_4, \mathcal{M}_4^{-1})$ を入出力に加えた暗号学的置換を考えた場合，パディングの有無を区別する定数 c_M が暗号学的置換の内部に XOR されることとなる．このような方式に対しての理論的なモデル化は直ちに明らかでなく，[Hir18] の証明可能安全性の結果をそのまま用いることができるかは明らかではないことを付記する．

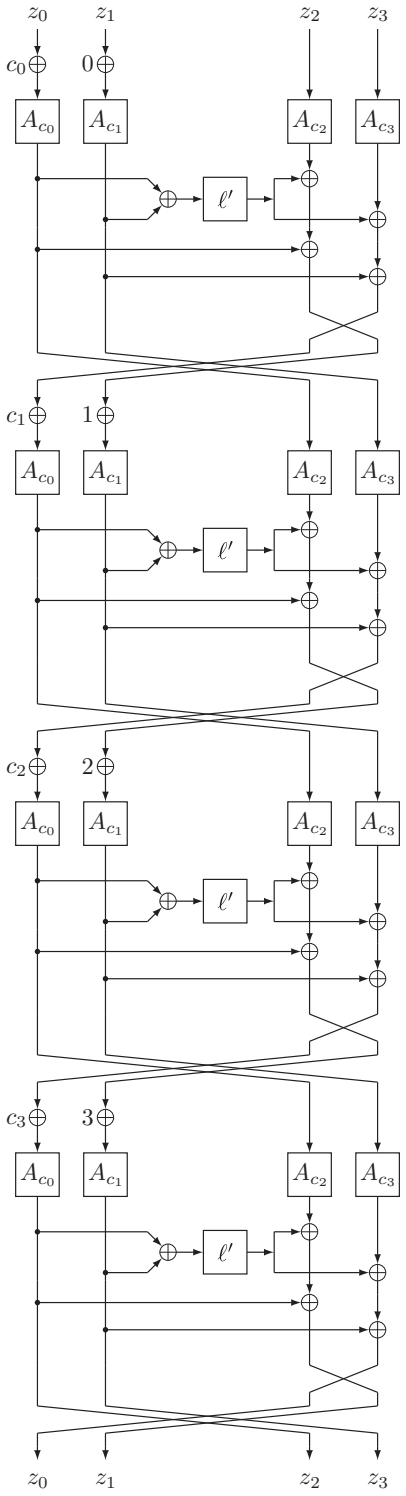

図 3.2: Sparkle256 の動作．A_{c_i} は定数（鍵）を c_i とした Alzette であり，入出力は $(z_0, \ldots, z_3) \in \{0,1\}^{256}$ である．

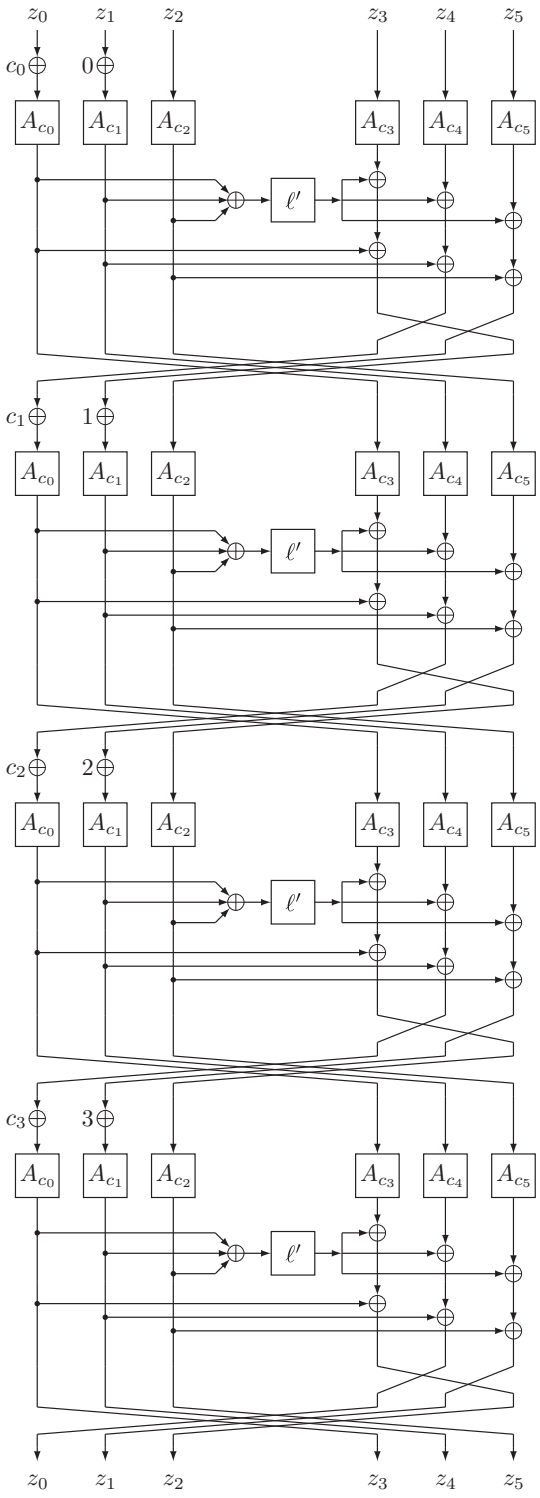

図 3.3: Sparkle384 の動作．入出力は $(z_0, \ldots, z_5) \in \{0,1\}^{384}$ である．

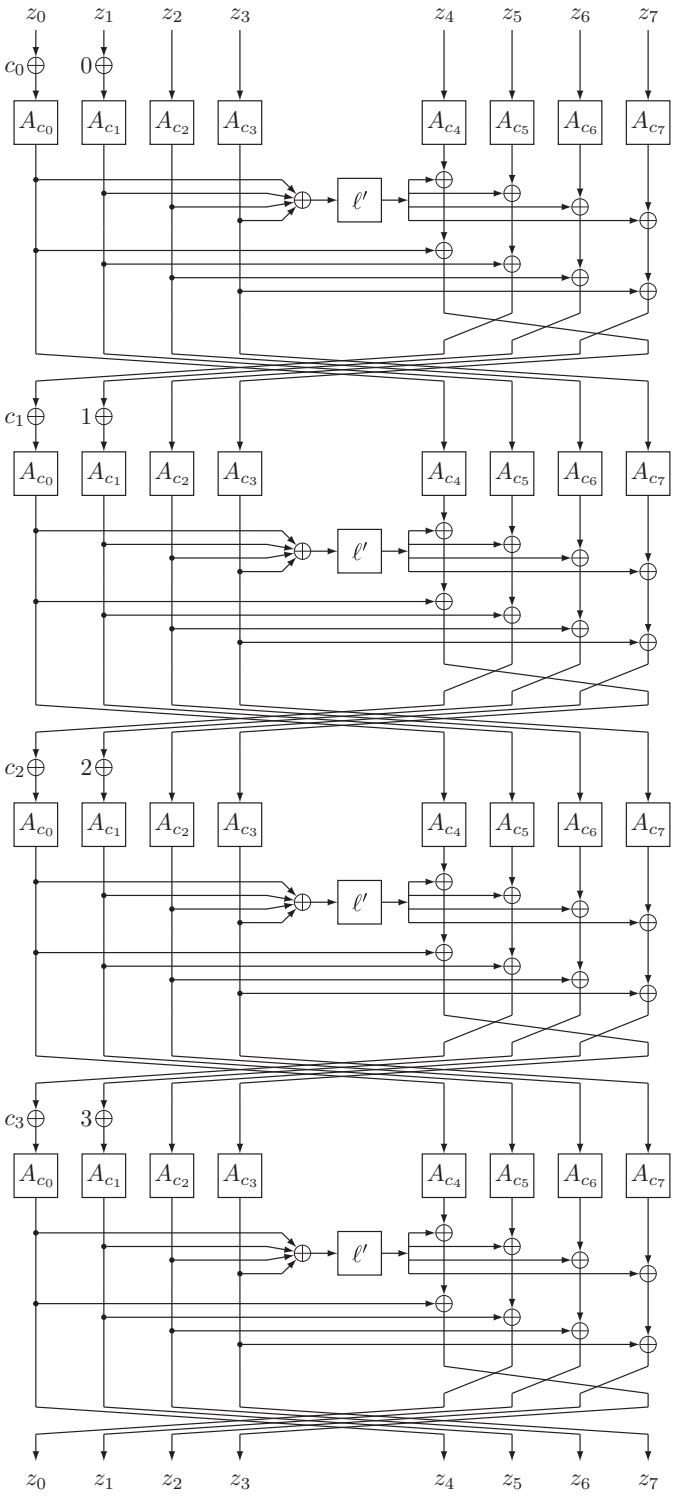

図 3.4: Sparkle512 の動作．入出力は $(z_0, \ldots, z_7) \in \{0,1\}^{512}$ である．

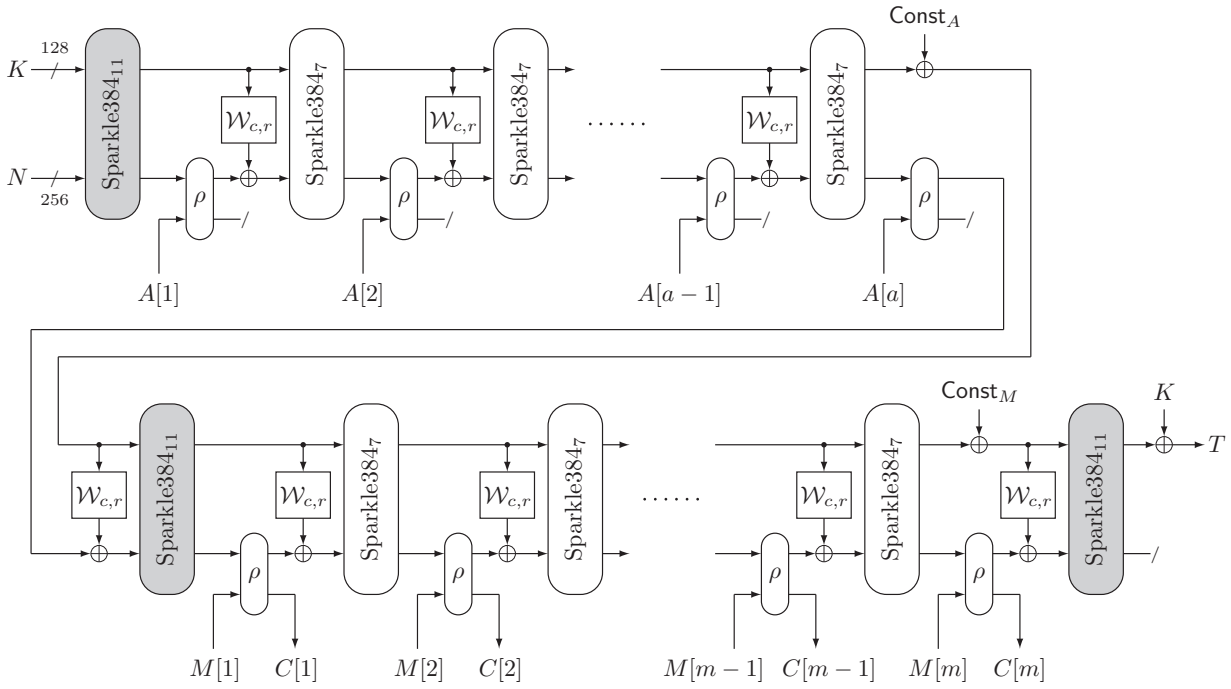

図 3.5: Schwaemm256-128 の暗号化関数．入力は (K, N, A, M) であり，出力は (C, T) である．

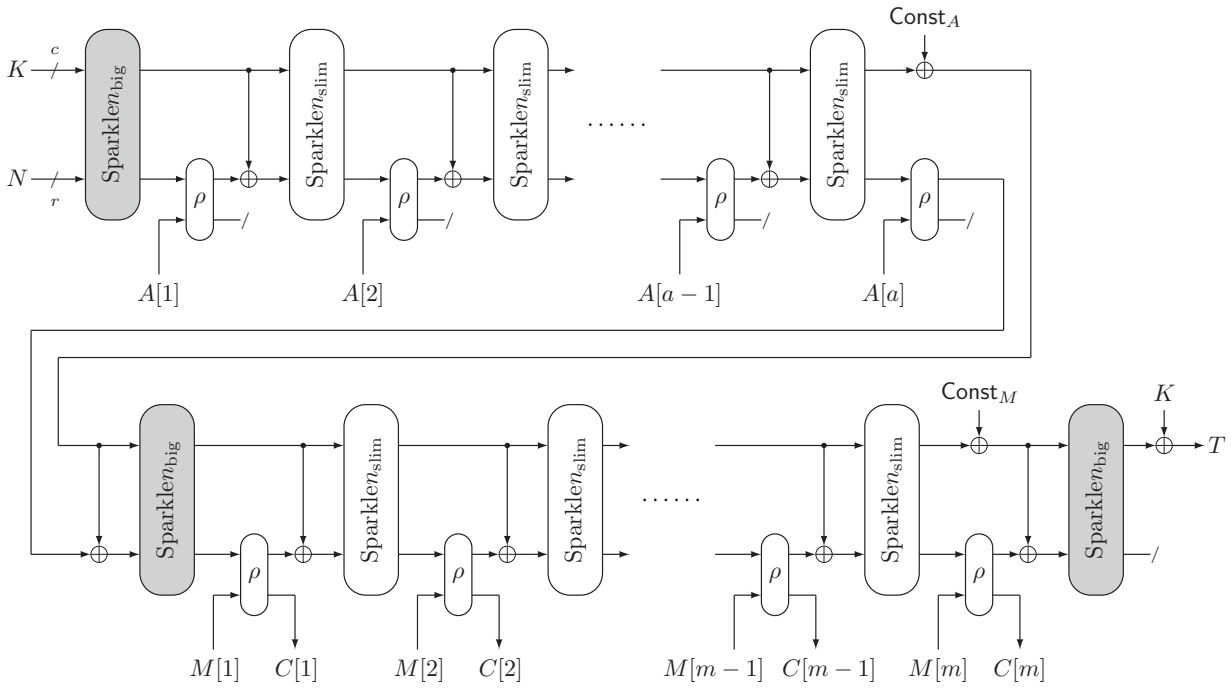

図 3.6: Schwaemm$r$-c，$r = c$ の暗号化関数．$r, c \in \{128, 192, 256\}$ であり，$n = r + c$ とした Sparkle$n$ を用いる．入力は (K, N, A, M) であり，出力は (C, T) である．

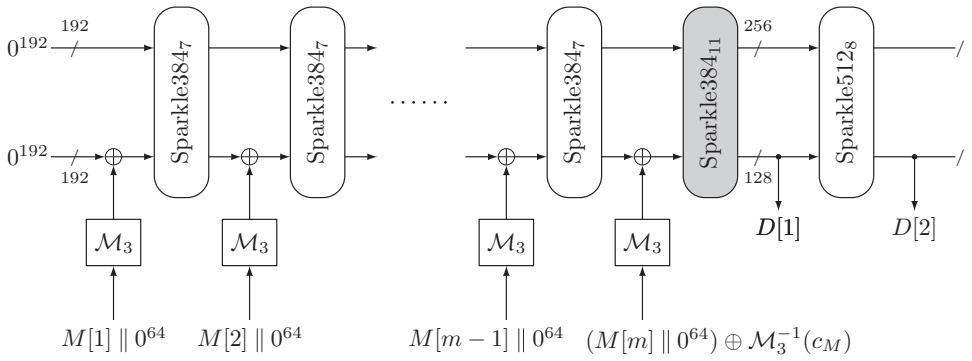

図 3.7: Esch256 の動作．$\mathcal{M}_3 : \{0,1\}^{192} \to \{0,1\}^{192}$ であり，各 $M[i]$（$1 \le i \le m-1$）は 128 ビットである．

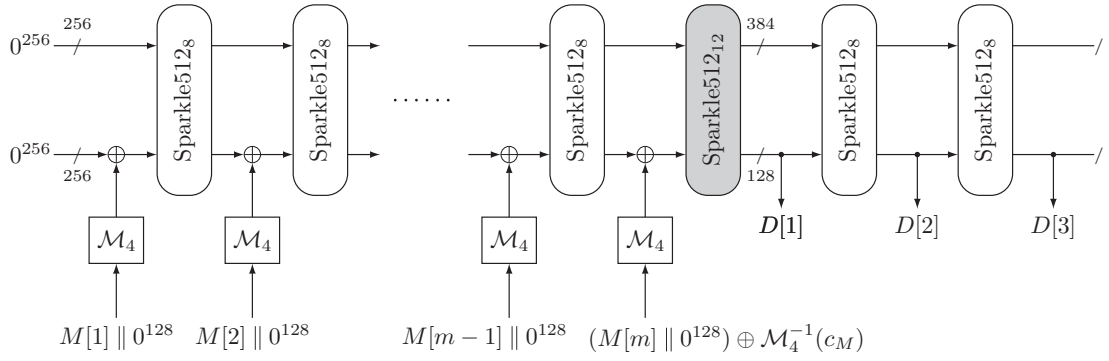

図 3.8: Esch384 の動作．$\mathcal{M}_4 : \{0,1\}^{256} \to \{0,1\}^{256}$ であり，各 $M[i]$（$1 \le i \le m-1$）は 128 ビットである．

Chapter 4

Tsudik's keymodeの安全性に関する調査及び評価

4.1 方式概要

Tsudik's keymode は Tsudik により設計された MAC であり [Tsu92]，ISO/IEC 29192-6:2019 Information technology — Lightweight cryptography — Part 6: Message authentication codes (MACs) にて国際標準化されている [ISO19]．Tsudik's keymode は軽量 MAC として国際標準化されており，軽量ハッシュ関数を構成要素として用いる．

$H : \{0,1\}^* \to \{0,1\}^n$ を出力長 n ビットのハッシュ関数とする．鍵長 k ビット，タグ長 t ビットの Tsudik's keymode は，$\mathrm{TKM} : \{0,1\}^k \times \{0,1\}^* \to \{0,1\}^t$,

$$\mathrm{TKM}_K(M) = \lfloor H(K \parallel M) \rfloor_t \tag{4.1}$$

と定義される．ただし，$K \in \{0,1\}^k$ は秘密鍵，$M \in \{0,1\}^*$ はメッセージ，$T \in \{0,1\}^t$ はタグである．また，$n \geq t$ とし，t ビット以上のビット列 X に対し，$\lfloor X \rfloor_t$ は X の最下位 t ビットを表す.

4.2 安全性調査

4.2.1 [Tsu92] の安全性解析

Tsudik により，[Tsu92] において Tsudik's keymode の安全性が解析されている．[Tsu92] では 2 通りの方式が提案されており，式 (4.1) はそのうちの secret prefix method と呼ばれる手法である．もう一方は secret suffix method であり，

$$\mathrm{TKM}_K(M) = \lfloor H(M \parallel K) \rfloor_t \tag{4.2}$$

と定義される．式 (4.1) と式 (4.2) とでは，K と M の連結の順番が異なる．また，式 (4.1) と式 (4.2) の両方を組み合わせた hybrid method として，

$$\mathrm{TKM}_{K,K'}(M) = \lfloor H(K \parallel M \parallel K') \rfloor_t \tag{4.3}$$

も提案されている．この方式は K, K' の 2 つの鍵を入力とする．ISO/IEC 29192-6:2019 で標準化されているのは式 (4.1) のみである．

[Tsu92] においてはハッシュ関数として MD4 を例として挙げており，ハッシュ関数について 2 つの conjecture と 2 つの claim を挙げている．

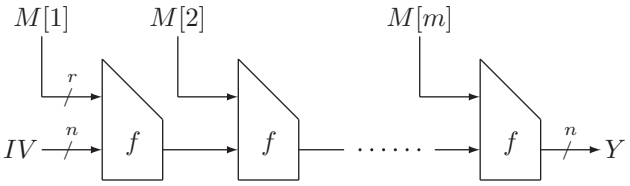

図 4.1: 圧縮関数 $f : \{0,1\}^{r+n} \to \{0,1\}^n$ を使用した MD 変換に基づくハッシュ関数 H^f．$IV \in \{0,1\}^n$ は初期値，入力は $M = (M[1] \,\|\, \cdots \,\|\, M[m])$ であり，簡単のため各 $M[i]$ は r ビットであるとする．出力はハッシュ値 $T \in \{0,1\}^n$ であり，$T = H^f(M)$ と書く．

Conjecture 1. $M \neq M'$，$H(M) = H(M')$ を満たす M, M' を見つける計算量は $O(2^{n/2})$ である．

Conjecture 2. ハッシュ値 $Y \in \{0,1\}^n$ が与えられ，$H(M) = Y$ を満たす M を見つける計算量は $O(2^n)$ である．

Claim 1. ハッシュ値 $Y \in \{0,1\}^n$ と M が与えられ，$H(M' \,\|\, M) = Y$ を満たす M' を見つける計算量は $O(2^n)$ である．

Claim 2. ハッシュ値 $Y \in \{0,1\}^n$ と M が与えられ，$H(M \,\|\, M') = Y$ を満たす M' を見つける計算量は $O(2^n)$ である．

これらの conjecture と claim の成立を仮定し，[Tsu92] において，式 (4.1) あるいは式 (4.2) の方式を攻撃するには全数探索攻撃以外に攻撃方法がないことが主張されている．ここでの全数探索は衝突の発見や原像の計算を含んでいる．

また，[Tsu92] では攻撃法についても議論されている．ハッシュ関数として，length-extension 攻撃が可能であるような方式は式 (4.1) で用いてはならないことが述べられている．ハッシュ関数に対する length-extension 攻撃は，ある M に対するハッシュ値 $H(M)$ が既知である場合に，(M を知らなくても) 任意の M' に対し，$M \,\|\, M'$ に対するハッシュ値 $H(M \,\|\, M')$ を効率的に計算する攻撃である．SHA-256 に代表されるハッシュ関数で採用されている MD 変換（Merkle-Damgård 変換 [Mer89, Dam89]）では，length-extension 攻撃が可能であり，これらを式 (4.1) で用いることはできない．なお本報告書では MD 変換に基づくハッシュ関数は，図 4.1の構成を指す．以下，攻撃の概要を述べる．

ハッシュ値のビット長とタグ長が一致する $n = t$ の場合を考える．メッセージとタグのペア (M, T) が既知であるとする．ただし，$T = \lfloor H(K \| M) \rfloor_t$ であり，K は未知の秘密鍵である．ハッシュ関数に対する length-extension 攻撃が可能である場合，敵は任意の M' を選択し，T から $T' = \lfloor H(K \| M \| M') \rfloor_t$ を計算でき，$(M \,\|\, M', T')$ は受信者によって正しいメッセージ，タグのペアであると判定され，偽造攻撃に成功する．

この攻撃の対処法として，[Tsu92] において，入力メッセージのビット長をハッシュ関数の入力の prefix の一部とすることが言及されている．ただし，4.2.3章で記載するように，この対策を施した方式に対しても様々な解析方法が知られている．

なお，ハッシュ関数に対する length-extension 攻撃は，式 (4.2) の方式では問題とはならないと [Tsu92] において述べられている．また，計算量が $2^{n/2}$ である birthday 攻撃により，式 (4.2) の方式は攻撃可能であるが，同様の攻撃は式 (4.1) の方式には適用できないことが主張されている．これは，MD 変換によるハッシュ関数 H において次の事実が成り立つことによる：

- $H(M) = H(M')$，$M \neq M'$ ならば，任意の K に対し，$H(M \,\|\, K) = H(M' \,\|\, K)$ が成り立つ．
- $H(M) = H(M')$，$M \neq M'$ であったとしても，$H(K \,\|\, M) = H(K \,\|\, M')$ が成り立つとは限らない．

式 (4.3) の hybrid method について，鍵回復攻撃にかかる計算量が $2^{|K|+|K'|}$ であることが期待されるとの記述があるが，4.2.3章で記載するように，これは一般には成り立たない．

4.2.2 [ISO19] の安全性解析

[ISO19] において，国際標準方式である式 (4.1) の安全性が述べられている．

ハッシュ関数は衝突困難性を有することと，length-extension 攻撃ができないことが要件として挙げられており，ISO/IEC 29192-5:2016 にある方式を使用することとされている．当該文書は軽量ハッシュ関数に関する標準であり，

- PHOTON [GPP11]
- SPONGENT [BKL+11]
- Lesamnta-LW [HIK+12]

の 3 方式が記載されている[1]．ISO/IEC 29192-5:2016 の Annex C にハッシュ関数の選択に関する言及があり，(可変長入力の）ランダムオラクル [BR93] からの強識別不可能性（indifferentiability [MRH04, CDMP05]）を有するハッシュ関数であれば，Tsudik's keymode での利用に適していることが記載されている．PHOTON と SPONGENT はこれに該当する．Lesamnta-LW に関しては，[HIK+12] において，Tsudik's keymode で利用した場合の擬似ランダム性の証明がされている．このため，ISO/IEC 29192-5:2016 にある上記の 3 方式の軽量ハッシュ関数はいずれも Tsudik's keymode での利用に適している．

一方で，ISO/IEC 10118-3 に記載のハッシュ関数を Tsudik's keymode で利用すべきではないことが記載されている．なお，執筆時点の最新版は ISO/IEC 10118-3:2018 であり，これには強識別不可能性を有するとされる SHA-3 が記載されており，ISO/IEC 10118-3 に記載のハッシュ関数全てを排除する必要はないことを付記する．

4.2.3 文献調査

本報告書執筆時点で [Tsu92] を引用する文献は Google Scholar によれば 488 件ある．本章では，このうち Tsudik's keymode の安全性に関わる主な文献をリストする．

式 (4.1)，式 (4.2)，式 (4.3) の 3 方式の安全性が [PvO95, PvO99] において解析されている．[PvO95, Table 1] にその結果がまとめられており，これを表 4.1に再掲する．

- 表 4.1の式 (4.1) の鍵回復攻撃について，$(\#\mathrm{MAC}, \#\mathrm{opn}) = (1, 0)$ とあるのは，秘密鍵に相当する情報が既知のメッセージとタグのペアから漏洩することによるものであり，秘密鍵自体を回復できるわけではない．
- 表 4.1の式 (4.3) の鍵回復に関する攻撃について，$(\#\mathrm{MAC}, \#\mathrm{opn}) = (2^{n/2}, 2^k + 2^{k'})$ の攻撃は，$(\#\mathrm{MAC}, \#\mathrm{opn}) = (\lceil \frac{k+k'}{n} \rceil, 2^{k+k'})$ よりもデータを必要とするが計算量が少なく，[Tsu92] にある鍵回復攻撃にかかる計算量の期待を覆している．

以下， 式 (4.1) の国際標準方式とは異なるが，関連する方式の安全性解析を挙げる．

- 式 (4.3) の修正版である $T = H(K \parallel P \parallel M \parallel K)$ とした方式の安全性が [PvO96] で解析されている．ここで，H は MD5，K は秘密鍵，P はパディング，M は入力メッセージ，T はタグで

[1] なおこれらのハッシュ関数自体の安全性解析は本報告書では扱わない．

表 4.1: Tsudik's keymode の安全性 [PvO95, Table 4.1]（一部改変）．各数字は攻撃に必要な計算量を表している．ハッシュ関数は MD 変換による方式を想定し，length-extension 攻撃が可能であるとする．#MAC は既知のメッセージ，タグのペアの数を表し，#opn は MD 変換の圧縮関数の計算回数を表す．C は選択メッセージとタグのペアの数を表す．$|K| = k, |K'| = k'$ である．n は MD 変換における chaining value のビット長であり，タグ長とも一致するとする．τ は既知のメッセージの数を表す．

攻撃手法	理想的な MAC		式 (4.1)		式 (4.2)		式 (4.3)	
	#MAC	#opn	#MAC	#opn	#MAC	#opn	#MAC	#opn
鍵回復	$\lceil \frac{k}{n} \rceil$	2^k	1	0	$\lceil \frac{k}{n} \rceil$	2^k	$\lceil \frac{k+k'}{n} \rceil$ $2^{n/2}$	$2^{k+k'}$ $2^k + 2^{k'}$
偽造	$\lceil \frac{k}{n} \rceil$	2^k	1	1	1C	$2^{n/2}$	$5\mathrm{C} + 2^{n/2}$ $2^{n/2}$	0 2^k
第 2 原像計算	2^n	0	τ	$2^n/\tau$	τ	$2^n/\tau$	2^n	0

ある．$O(2^{n/2})$ の選択メッセージと $O(2^{n/2})$ の既知メッセージにより，鍵回復ができることが示されている．

この方式に関連して，$T = H(K \parallel P \parallel M \parallel P' \parallel K)$ とした方式は sandwitch scheme と呼ばれ，圧縮関数の擬似ランダム性を仮定した証明可能安全性が [Yas07] で示されている．ここで，H は MD 変換に基づくハッシュ関数，K は秘密鍵，P, P' はパディング，M は入力メッセージ，T はタグである．また [KM14] では同方式の証明可能安全性を関連鍵攻撃に関する安全性を仮定し，構成的な帰着により示している．

- 式 (4.1) の修正版である $T = H(K \parallel \ell \parallel M)$ とした方式は [WWJW09] において，LPMAC と呼ばれている．ここで，H は MD 変換に基づくハッシュ関数，K は秘密鍵，ℓ は入力メッセージの長さ，M は入力メッセージ，T はタグである．
  - この方式は圧縮関数の擬似ランダム性を仮定し，全体が $O(2^{n/2})$ 回のクエリまで安全な擬似ランダム関数であることの証明がある [BCK96].
  - この方式は [PvO95] で解析されており，chaining value の衝突を発見する攻撃が可能であることが述べられている．
  - [Sas12, Sas14] では，$O(2^{n/2})$ のクエリによる distinguishing-H 攻撃と almost universal forgery が可能であることが示されている．
    なお distinguishing-H 攻撃とは $H_K^f(\cdot)$ と $H_K^r(\cdot)$ とを識別する攻撃である．ここで K は秘密鍵，f は圧縮関数，r は f と同じ入出力空間を有するランダム関数であり，$H_K^f(M) = H(K \parallel \ell \parallel M)$ は f を圧縮関数として用いた MD 変換に基づくハッシュ関数 H の入力 $K \parallel \ell \parallel M$ に対するハッシュ値であり，$H_K^r(M)$ は r を用いて同様に定義される．
    また，ここでの almost universal forgery とは，与えられた m ブロックのメッセージ $M = (M[1] \parallel \cdots \parallel M[m])$ に対し，先頭 $d = \lceil \log m \rceil$ ブロックを敵が選択した任意の $M'[1] \parallel \cdots \parallel M'[d]$ に改変し，メッセージ

    $$M'[1] \parallel \cdots \parallel M'[d] \parallel M[d+1] \parallel \cdots \parallel M[m]$$

    に対するタグを偽造する攻撃である．

表 4.2: LPMAC に対する distinguishing-H 攻撃 [Sas12, Table 1]（一部改変）．ラウンド数 x/y は攻撃可能段数が x ラウンド，仕様段数が y ラウンドであることを表す．

	攻撃対象	出力ビット長	ラウンド数	クエリ回数	文献
	SHA-1	160	43/80	$2^{124.5}$	[WWJW09]
	SHA-1	160	61/80	$2^{154.5}$	[WWJW09]
	SHA-1	160	65/80	$2^{80.9}$	[QWJ09]
	SHA-256	256	39/64	$2^{184.5}$	[YW09]
	RIPEMD	128	48/48 (full)	2^{66}	[Wan10]
	RIPEMD-256	256	58/64	$2^{163.5}$	[Wan10]
	RIPEMD-320	320	48/80	$2^{208.5}$	[Wan10]
MD 変換によるハッシュ関数		n	full	$3 \cdot 2^{n/2}$	[Sas12]

  - ハッシュ関数 H としてステップ数を削減した SHA-1 や SHA-256 を使用した LPMAC の distinguishing-H 攻撃に関する安全性が [WWJW09, QWJ09, YW09, Wan10] で解析されている．解析結果が [Sas12, Table 1] にまとめられており，表 4.2にそれを再掲する．なお [Sas12] の攻撃は MD 変換に基づくハッシュ関数に対する汎用的な攻撃であり，[WWJW09, QWJ09, YW09, Wan10] の結果を改善している．

- $T = H(H(K \,\|\, P \,\|\, M)) = H^2(K \,\|\, P \,\|\, M)$ とした方式は [Yas09] において，H^2-MAC と呼ばれ，証明可能安全性が示されている．ここで，H は MD 変換に基づくハッシュ関数，K は秘密鍵，P はパディング，M は入力メッセージ，T はタグである．

- [LXS11] では，MD 変換に基づくハッシュ関数を利用した H^2-MAC に対し，$O(2^{n/2})$ のオンラインクエリと $O(2^{n/2})$ のオフラインクエリを用いた古典モデルにおける online-offline 中間値一致攻撃が示されている．これに対し [HS18] では，古典オンラインクエリとオフライン量子計算機を仮定する Q1 モデルにおいて，[LXS11] の攻撃のオフラインフェーズが改善できることを指摘するとともに，同様の改善が LPMAC，式 (4.1) の方式に対して可能であることを指摘している．

4.3 安全性評価

4.2.1章に記載の 2 つの conjecture と 2 つの claim は，ランダムオラクル [BR93] について成立する．また，式 (4.1) と式 (4.2) ともに，H が可変長入力のランダムオラクルであれば，理想的に安全な k ビット鍵，t ビット出力の擬似ランダム関数になる．したがって，ランダムオラクルとの強識別不可能性を有するハッシュ関数を利用すれば，式 (4.1) の Tsudik's keymode は証明可能安全性を有する方式である．

4.2.2章に記載の通り，PHOTON [GPP11] と SPONGENT [BKL+11] はこれに該当する．一般に Sponge 構造の証明可能安全性の結果 [BDPA08] に含まれるようなハッシュ関数や，MDP（Merkle-Damgård with a permutation [HPY12]）であれば，Tsudik's keymode での利用に安全性上の問題は生じない．なお，MDP については式 (4.1) で利用した際の擬似ランダム関数としての証明が [HPY12] にて示されている．

Lesamnta-LW 自体は MD 変換によるハッシュ関数であり length-extension 攻撃が可能であるが，[HIK+12] で示されている通り，出力を適切に切り捨てることにより，擬似ランダム関数としての直接的な証明が可能となり，Tsudik's keymode での利用に問題がない．

式 (4.1) の Tsudik's keymode はハッシュ関数が length-extension 攻撃を許す場合は明らかな脆弱性を有している．このことは認識される必要があり，必要に応じて注意喚起が必要であると考えられるが，衝突を計算することが困難であり，強識別不可能性を有するハッシュ関数を利用すれば，式 (4.1) の Tsudik's keymode の安全性に懸念点は見受けられない．

参考文献

[BBdS+20a] Christof Beierle, Alex Biryukov, Luan Cardoso dos Santos, Johann Großschädl, Léo Perrin, Aleksei Udovenko, Vesselin Velichkov, and Qingju Wang. Alzette: A 64-bit arx-box - (feat. CRAX and TRAX). In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology - CRYPTO 2020 - 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17-21, 2020, Proceedings, Part III*, volume 12172 of *Lecture Notes in Computer Science*, pages 419–448. Springer, 2020.

[BBdS+20b] Christof Beierle, Alex Biryukov, Luan Cardoso dos Santos, Johann Großschädl, Léo Perrin, Aleksei Udovenko, Vesselin Velichkov, and Qingju Wang. Lightweight AEAD and hashing using the Sparkle permutation family. *IACR Trans. Symmetric Cryptol.*, 2020(S1):208–261, 2020.

[BBdS+21] Christof Beierle, Alex Biryukov, Luan Cardoso dos Santos, Johann Großschädl, Léo Perrin, Aleksei Udovenko, Vesselin Velichkov, and Qingju Wang. Schwaemm and Esch: Lightweight authenticated encryption and hashing using the Sparkle permutation family. NIST submission, 2021.

[BBdS+22] Christof Beierle, Alex Biryukov, Luan Cardoso dos Santos, Johann Großschädl, Amir Moradi, Léo Perrin, Aein Rezaei Shahmirzadi, Aleksei Udovenko, Vesselin Velichkov, and Qingju Wang. An update on the LWC finalist Sparkle. NIST submission, 2022.

[BCD+21] Zhenzhen Bao, Avik Chakraborti, Nilanjan Datta, Jian Guo, Mridul Nandi, Thomas Peyrin, and Kan Yasuda. PHOTON-Beetle authenticated encryption and hash family. NIST submission, 2021.

[BCK96] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. Pseudorandom functions revisited: The cascade construction and its concrete security. In *37th Annual Symposium on Foundations of Computer Science, FOCS '96, Burlington, Vermont, USA, 14-16 October, 1996*, pages 514–523. IEEE Computer Society, 1996.

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

[BDPA11] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Duplexing the Sponge: Single-pass authenticated encryption and other applications. In Ali Miri and Serge Vaudenay, editors, *Selected Areas in Cryptography - 18th International Workshop, SAC 2011, Toronto, ON, Canada, August 11-12, 2011, Revised Selected Papers*, volume 7118 of *Lecture Notes in Computer Science*, pages 320–337. Springer, 2011.

[Bih93] Eli Biham. New types of cryptanalytic attacks using related keys (extended abstract). 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 398–409. Springer, 1993.

[BK03] Mihir Bellare and Tadayoshi Kohno. A theoretical treatment of related-key attacks: RKA-PRPs, RKA-PRFs, and applications. In Eli Biham, editor, *Advances in Cryptology - EUROCRYPT 2003, International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4-8, 2003, Proceedings*, volume 2656 of *Lecture Notes in Computer Science*, pages 491–506. Springer, 2003.

[BKL+11] Andrey Bogdanov, Miroslav Knezevic, Gregor Leander, Deniz Toz, Kerem Varici, and Ingrid Verbauwhede. SPONGENT: A lightweight hash function. In Bart Preneel and Tsuyoshi Takagi, editors, *Cryptographic Hardware and Embedded Systems - CHES 2011 - 13th International Workshop, Nara, Japan, September 28 - October 1, 2011. Proceedings*, volume 6917 of *Lecture Notes in Computer Science*, pages 312–325. Springer, 2011.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In Dorothy E. Denning, Raymond Pyle, Ravi Ganesan, Ravi S. Sandhu, and Victoria Ashby, editors, *CCS '93, Proceedings of the 1st ACM Conference on Computer and Communications Security, Fairfax, Virginia, USA, November 3-5, 1993*, pages 62–73. ACM, 1993.

[CDMP05] Jean-Sébastien Coron, Yevgeniy Dodis, Cécile Malinaud, and Prashant Puniya. Merkle-damgård revisited: How to construct a hash function. In Victor Shoup, editor, *Advances in Cryptology - CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-18, 2005, Proceedings*, volume 3621 of *Lecture Notes in Computer Science*, pages 430–448. Springer, 2005.

[CDNY18a] Avik Chakraborti, Nilanjan Datta, Mridul Nandi, and Kan Yasuda. Beetle family of lightweight and secure authenticated encryption ciphers. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(2):218–241, 2018.

[CDNY18b] Avik Chakraborti, Nilanjan Datta, Mridul Nandi, and Kan Yasuda. Beetle family of lightweight and secure authenticated encryption ciphers. *IACR Cryptol. ePrint Arch.*, page 805, 2018.

[CJN19] Bishwajit Chakraborty, Ashwin Jha, and Mridul Nandi. On the security of Sponge-type authenticated encryption modes. *IACR Cryptol. ePrint Arch.*, page 1475, 2019.

[CJN20] Bishwajit Chakraborty, Ashwin Jha, and Mridul Nandi. On the security of Sponge-type authenticated encryption modes. *IACR Trans. Symmetric Cryptol.*, 2020(2):93–119, 2020.

[CSCW17] Tingting Cui, Ling Sun, Huaifeng Chen, and Meiqin Wang. Statistical integral distinguisher with multi-structure and its application on AES. In Josef Pieprzyk and Suriadi Suriadi, editors, *Information Security and Privacy - 22nd Australasian Conference, ACISP 2017, Auckland, New Zealand, July 3-5, 2017, Proceedings, Part I*, volume 10342 of *Lecture Notes in Computer Science*, pages 402–420. Springer, 2017.

[Dam89] Ivan Damgård. A design principle for hash functions. In Gilles Brassard, editor, *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*, volume 435 of *Lecture Notes in Computer Science*, pages 416–427. Springer, 1989.

[DM20] Christoph Dobraunig and Bart Mennink. Key recovery attack on PHOTON-Beetle. NIST lightweight-crypto mailing list, ROUND 2 OFFICIAL COMMENT: PHOTON-Beetle, 2020.

[DPU+16] Daniel Dinu, Léo Perrin, Aleksei Udovenko, Vesselin Velichkov, Johann Großschädl, and Alex Biryukov. Design strategies for ARX with provable bounds: Sparx and LAX. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I*, volume 10031 of *Lecture Notes in Computer Science*, pages 484–513, 2016.

[Glo18] GlobalPlatform Technology. Card specification version 2.3.1. Document Reference: GPC_SPE_034, 2018.

[GPP] Jian Guo, Thomas Peyrin, and Axel Poschmann. The PHOTON family of lightweight hash functions. `https://sites.google.com/site/photonhashfunction/security`.

[GPP11] Jian Guo, Thomas Peyrin, and Axel Poschmann. The PHOTON family of lightweight hash functions. In Phillip Rogaway, editor, *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*, volume 6841 of *Lecture Notes in Computer Science*, pages 222–239. Springer, 2011.

[Gro96] Lov K. Grover. A fast quantum mechanical algorithm for database search. In Gary L. Miller, editor, *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, pages 212–219. ACM, 1996.

[GSM16] GSM Association. Remote provisioning architecture for embedded uicc technical specification. version 3.1. Official Document SGP.02, 2016.

[HIK+12] Shoichi Hirose, Kota Ideguchi, Hidenori Kuwakado, Toru Owada, Bart Preneel, and Hirotaka Yoshida. An AES based 256-bit hash function for lightweight applications:

Lesamnta-LW. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, 95-A(1):89–99, 2012.

[Hir18] Shoichi Hirose. Sequential hashing with minimum padding. *Cryptogr.*, 2(2):11, 2018.

[HPY12] Shoichi Hirose, Je Hong Park, and Aaram Yun. A simple variant of the Merkle-Damgård scheme with a permutation. *J. Cryptol.*, 25(2):271–309, 2012.

[HS18] Akinori Hosoyamada and Yu Sasaki. Cryptanalysis against symmetric-key schemes with online classical queries and offline quantum computations. In Nigel P. Smart, editor, *Topics in Cryptology - CT-RSA 2018 - The Cryptographers' Track at the RSA Conference 2018, San Francisco, CA, USA, April 16-20, 2018, Proceedings*, volume 10808 of *Lecture Notes in Computer Science*, pages 198–218. Springer, 2018.

[HW19] Mingjiang Huang and Liming Wang. Automatic tool for searching for differential characteristics in ARX ciphers and applications. In Feng Hao, Sushmita Ruj, and Sourav Sen Gupta, editors, *Progress in Cryptology - INDOCRYPT 2019 - 20th International Conference on Cryptology in India, Hyderabad, India, December 15-18, 2019, Proceedings*, volume 11898 of *Lecture Notes in Computer Science*, pages 115–138. Springer, 2019.

[HW20] Mingjiang Huang and Liming Wang. Automatic search for the linear (hull) characteristics of ARX ciphers: Applied to SPECK, SPARX, Chaskey, and CHAM-64. *Secur. Commun. Networks*, 2020:4898612:1–4898612:14, 2020.

[HXW22] Mingjiang Huang, Zhen Xu, and Liming Wang. On the probability and automatic search of rotational-xor cryptanalysis on ARX ciphers. *Comput. J.*, 65(12):3062–3080, 2022.

[IIM22] Akiko Inoue, Tetsu Iwata, and Kazuhiko Minematsu. Analyzing the provable security bounds of GIFT-COFB and Photon-Beetle. In Giuseppe Ateniese and Daniele Venturi, editors, *Applied Cryptography and Network Security - 20th International Conference, ACNS 2022, Rome, Italy, June 20-23, 2022, Proceedings*, volume 13269 of *Lecture Notes in Computer Science*, pages 67–84. Springer, 2022.

[ISO19] ISO/IEC. Information technology — lightweight cryptography — part 6: Message authentication codes (MACs). ISO/IEC 29192-6:2019, 2019.

[JK22] Adam Jagielski and Krzysztof Kanciak. Grover on sparkle quantum resource estimation for a NIST LWC call finalist. *Quantum Inf. Comput.*, 22(13&14):1132–1143, 2022.

[JNP12] Jérémy Jean, María Naya-Plasencia, and Thomas Peyrin. Improved rebound attack on the finalist Grøstl. In Anne Canteaut, editor, *Fast Software Encryption - 19th International Workshop, FSE 2012, Washington, DC, USA, March 19-21, 2012. Revised Selected Papers*, volume 7549 of *Lecture Notes in Computer Science*, pages 110–126. Springer, 2012.

[JNP13] Jérémy Jean, María Naya-Plasencia, and Thomas Peyrin. Multiple limited-birthday distinguishers and applications. In Tanja Lange, Kristin E. Lauter, and Petr Lisonek, editors, *Selected Areas in Cryptography - SAC 2013 - 20th International Conference, Burnaby, BC, Canada, August 14-16, 2013, Revised Selected Papers*, volume 8282 of *Lecture Notes in Computer Science*, pages 533–550. Springer, 2013.

[JP22] Amit Jana and Goutam Paul. Differential fault attack on PHOTON-Beetle. In Chip-Hong Chang, Ulrich Rührmair, Debdeep Mukhopadhyay, and Domenic Forte, editors, *Proceedings of the 2022 Workshop on Attacks and Solutions in Hardware Security, ASHES 2022, Los Angeles, CA, USA, 11 November 2022*, pages 25–34. ACM, 2022.

[KM14] Neal Koblitz and Alfred Menezes. Another look at security theorems for 1-key nested MACs. In Çetin Kaya Koç, editor, *Open Problems in Mathematics and Computational Science*, pages 69–89. Springer, 2014.

[LM22] Charlotte Lefevre and Bart Mennink. Tight preimage resistance of the Sponge construction. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15-18, 2022, Proceedings, Part IV*, volume 13510 of *Lecture Notes in Computer Science*, pages 185–204. Springer, 2022.

[LNS+23] Yunwen Liu, Zhongfeng Niu, Siwei Sun, Chao Li, and Lei Hu. Rotational differential-linear cryptanalysis revisited. *J. Cryptol.*, 36(1):3, 2023.

[LSL21] Yunwen Liu, Siwei Sun, and Chao Li. Rotational cryptanalysis from a differential-linear perspective - practical distinguishers for round-reduced FRIET, Xoodoo, and Alzette. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology - EUROCRYPT 2021 - 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17-21, 2021, Proceedings, Part I*, volume 12696 of *Lecture Notes in Computer Science*, pages 741–770. Springer, 2021.

[Luc04] Stefan Lucks. Ciphers secure against related-key attacks. In Bimal K. Roy and Willi Meier, editors, *Fast Software Encryption, 11th International Workshop, FSE 2004, Delhi, India, February 5-7, 2004, Revised Papers*, volume 3017 of *Lecture Notes in Computer Science*, pages 359–370. Springer, 2004.

[LXS11] Fanbao Liu, Tao Xie, and Changxiang Shen. Breaking H^2-MAC using birthday paradox. *IACR Cryptol. ePrint Arch.*, page 647, 2011.

[Még21] Alexandre Mége. Official comment: PHOTON-Beetle. NIST lightweight-crypto mailing list, ROUND 2 OFFICIAL COMMENT: PHOTON-Beetle, 2021.

[Mer89] Ralph C. Merkle. One way hash functions and DES. In Gilles Brassard, editor, *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*, volume 435 of *Lecture Notes in Computer Science*, pages 428–446. Springer, 1989.

[MRH04] Ueli M. Maurer, Renato Renner, and Clemens Holenstein. Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In Moni Naor, editor, *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004, Cambridge, MA, USA, February 19-21, 2004, Proceedings*, volume 2951 of *Lecture Notes in Computer Science*, pages 21–39. Springer, 2004.

[NSLL22] Zhongfeng Niu, Siwei Sun, Yunwen Liu, and Chao Li. Rotational differential-linear distinguishers of ARX ciphers with arbitrary output linear masks. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15-18, 2022, Proceedings, Part I*, volume 13507 of *Lecture Notes in Computer Science*, pages 3–32. Springer, 2022.

[PvO95] Bart Preneel and Paul C. van Oorschot. MDx-MAC and building fast MACs from hash functions. In Don Coppersmith, editor, *Advances in Cryptology - CRYPTO '95, 15th Annual International Cryptology Conference, Santa Barbara, California, USA, August 27-31, 1995, Proceedings*, volume 963 of *Lecture Notes in Computer Science*, pages 1–14. Springer, 1995.

[PvO96] Bart Preneel and Paul C. van Oorschot. On the security of two MAC algorithms. In Ueli M. Maurer, editor, *Advances in Cryptology - EUROCRYPT '96, International Conference on the Theory and Application of Cryptographic Techniques, Saragossa, Spain, May 12-16, 1996, Proceeding*, volume 1070 of *Lecture Notes in Computer Science*, pages 19–32. Springer, 1996.

[PvO99] Bart Preneel and Paul C. van Oorschot. On the security of iterated message authentication codes. *IEEE Trans. Inf. Theory*, 45(1):188–199, 1999.

[QWJ09] Siyuan Qiao, Wei Wang, and Keting Jia. Distinguishing attack on secret prefix MAC instantiated with reduced SHA-1. In Dong Hoon Lee and Seokhie Hong, editors, *Information, Security and Cryptology - ICISC 2009, 12th International Conference, Seoul, Korea, December 2-4, 2009, Revised Selected Papers*, volume 5984 of *Lecture Notes in Computer Science*, pages 349–361. Springer, 2009.

[Sas12] Yu Sasaki. Cryptanalyses on a Merkle-Damgård based MAC - Almost universal forgery and distinguishing-h attacks. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*, volume 7237 of *Lecture Notes in Computer Science*, pages 411–427. Springer, 2012.

[Sas14] Yu Sasaki. Cryptanalyses on a Merkle-Damgård based MAC - Almost universal forgery and distinguishing-H attacks. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, 97-A(1):167–176, 2014.

[Spe22] Ties Speel. Cryptanalysis of SPARKLE's ARX-box Alzette. Bachelor Thesis, Radboud University, 2022.

[SS22] André Schrottenloher and Marc Stevens. Simplified MITM modeling for permutations: New (quantum) attacks. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15-18, 2022, Proceedings, Part III*, volume 13509 of *Lecture Notes in Computer Science*, pages 717–747. Springer, 2022.

[Tsu92] Gene Tsudik. Message authentication with one-way hash functions. *Comput. Commun. Rev.*, 22(5):29–38, 1992.

[Wan10] Gaoli Wang. Distinguishing attacks on LPMAC based on the full RIPEMD and reduced-step RIPEMD-{256, 320}. In Xuejia Lai, Moti Yung, and Dongdai Lin, editors, *Information Security and Cryptology - 6th International Conference, Inscrypt 2010, Shanghai, China, October 20-24, 2010, Revised Selected Papers*, volume 6584 of *Lecture Notes in Computer Science*, pages 199–217. Springer, 2010.

[WGR18] Qingju Wang, Lorenzo Grassi, and Christian Rechberger. Zero-sum partitions of PHOTON permutations. In Nigel P. Smart, editor, *Topics in Cryptology - CT-RSA 2018 - The Cryptographers' Track at the RSA Conference 2018, San Francisco, CA, USA, April 16-20, 2018, Proceedings*, volume 10808 of *Lecture Notes in Computer Science*, pages 279–299. Springer, 2018.

[WWJW09] Xiaoyun Wang, Wei Wang, Keting Jia, and Meiqin Wang. New distinguishing attack on MAC using secret-prefix method. In Orr Dunkelman, editor, *Fast Software Encryption, 16th International Workshop, FSE 2009, Leuven, Belgium, February 22-25, 2009, Revised Selected Papers*, volume 5665 of *Lecture Notes in Computer Science*, pages 363–374. Springer, 2009.

[XLJ+22] Zheng Xu, Yongqiang Li, Lin Jiao, Mingsheng Wang, and Willi Meier. Do NOT misuse the Markov cipher assumption - Automatic search for differential and impossible differential characteristics in ARX ciphers. *IACR Cryptol. ePrint Arch.*, page 135, 2022.

[Yas07] Kan Yasuda. "Sandwich" is indeed secure: How to authenticate a message with just one hashing. In Josef Pieprzyk, Hossein Ghodosi, and Ed Dawson, editors, *Information Security and Privacy, 12th Australasian Conference, ACISP 2007, Townsville, Australia, July 2-4, 2007, Proceedings*, volume 4586 of *Lecture Notes in Computer Science*, pages 355–369. Springer, 2007.

[Yas09] Kan Yasuda. HMAC without the "second" key. In Pierangela Samarati, Moti Yung, Fabio Martinelli, and Claudio A. Ardagna, editors, *Information Security, 12th International Conference, ISC 2009, Pisa, Italy, September 7-9, 2009. Proceedings*, volume 5735 of *Lecture Notes in Computer Science*, pages 443–458. Springer, 2009.

[YW09] Hongbo Yu and Xiaoyun Wang. Distinguishing attack on the secret-prefix MAC based on the 39-step SHA-256. In Colin Boyd and Juan Manuel González Nieto, editors, *Information Security and Privacy, 14th Australasian Conference, ACISP 2009, Brisbane, Australia, July 1-3, 2009, Proceedings*, volume 5594 of *Lecture Notes in Computer Science*, pages 185–201. Springer, 2009.