変形型GCMのCommitting安全性解析

鳥羽 凌史[1,a)]　岩田 哲[2,b)]

概要：AES-GCM は NIST 標準の認証暗号方式であり，最も広く使用されている方式の 1 つである．しかし，AES-GCM には，1 つの鍵で暗号化できるデータ量における制限や，Key-Committing 安全性を満たさないといった問題がある．これを解決する方式として，GCM を利用し，192 ビットのランダムナンスをサポートすることでデータ量の制限をなくし，Key-Committing 安全性の有無を選択できる方式が提案された．本論文ではこれを変形型 GCM と呼ぶ．この方式について証明されている Committing 安全性は FROB 安全性と呼ばれる安全性のみであり，他の Committing 安全性解析は未解決である．本論文では，変形型 GCM の Committing 安全性を解析し，ナンス長が 192 ビットの変形型 GCM に対して Committing 安全性の一つである CMT-2 攻撃を示す．

キーワード：変形型 GCM，AES-GCM，Committing 安全性

Committing Security Analysis of Modified GCM

Ryoji Toba[1,a)]　Tetsu Iwata[2,b)]

Abstract: AES-GCM is an authenticated encryption algorithm standardized by NIST and is one of the most widely used schemes. However, it has a limitation on the amount of data that can be encrypted under a single key, and does not provide the key-committing security. To address these issues, a variant of the scheme has been proposed that supports 192-bit random nonces, thereby removing the limitation on the amount of data, and allows the choice of whether or not to satisfy the key-committing security. In this paper, we refer to this scheme as Modified GCM. To date, the only committing security property that has been proven for this scheme is a notion called FROB security, while other properties remain unproven. In this work, we analyze the committing security of Modified GCM and demonstrate a CMT-2 attack, which is a type of the committing security attack, against Modified GCM with a 192-bit random nonce.

Keywords: Modified GCM, AES-GCM, Committing Security

1. 背景

現在，最も広く使用されている認証暗号方式として，AES-GCM [1] が知られている．AES-GCM は NIST (National Institute of Standards and Technology) 標準暗号である一方で，ナンスが短く 1 つの鍵で処理できるデータ量に制限がある，Key-Committing 安全性を満たさないといった問題がある．

AES-GCM が効率的に処理できるナンス長は 96 ビットである．ナンス N をランダムナンス，ナンス長を $|N|$ ビット，クエリ回数を Q とすると，ナンス衝突確率が十分低くなる ($\leq 2^{-32}$) ためには，次の式が成立する必要がある．

$$\frac{Q^2}{2^{|N|+1}} \leq 2^{-32}$$

$|N| = 96$ を代入すると，$Q \leq 2^{32.5}$ が得られる．したがって，$2^{32.5}$ 回程度のクエリまではナンス衝突確率が十分低いことがわかる．しかし，現状は 2^{64} のクエリ回数に耐える方式が求められている [2]．また，Key-Committing 安全性を満たさない共通鍵暗号方式は，メッセージフランキング

[1] 名古屋大学 大学院工学研究科 情報・通信工学専攻
Graduate School of Engineering, Nagoya University
[2] 名古屋大学 未来材料・システム研究所
IMaSS, Nagoya University
[a)] toba.ryoji.m5@s.mail.nagoya-u.ac.jp
[b)] tetsu.iwata@nagoya-u.jp

表 1: 変形型 GCM0, 1 に対する Committing 安全性を破る攻撃計算量

方式＼安全性	FROB	CMT-1	CMT-2
変形型 GCM0	–	$2^{256/3}$ [7]	–
変形型 GCM1	2^{128}	–	定数（本論文）

システムで利用されたとき，攻撃者によって健全な利用者が悪者にされてしまう問題が知られている [3]．以上から，AES-GCM の問題点を解決することは重要である．

これらの AES-GCM の問題点を解決する方式として，AES-GCM を利用し，192 ビットのランダムナンスをサポートすることでデータ量の制限をなくし，Key-Committing 安全性の有無を選択できる方式が提案された．本論文ではこの方式を変形型 GCM と呼ぶ．変形型 GCM は，もともと NIST Accordion Cipher Mod Workshop 2024 で提案された方式であり [4], [5]，2025 年 3 月にアップデートされた [6]．前者を変形型 GCM0，後者を変形型 GCM1 と呼ぶことにする．変形型 GCM0 については，Key-Committing (CMT-1) 安全性を計算量約 $2^{256/3}$ で破る攻撃が存在する [7]．本論文では，変形型 GCM1 に注目する．

変形型 GCM1 における Key-Committing 安全性は証明されているが [8]，文献 [8] で示されているのは Committing 安全性における FROB 安全性と呼ばれる安全性のみであり，他の Committing 安全性解析は未解決である．本論文では，変形型 GCM1 の Committing 安全性を解析し，ナンス長が 192 ビットの変形型 GCM1 に対して，Committing 安全性の一つである CMT-2 を定数の計算量で破る攻撃を示す．また，攻撃の入出力の実例についても示す．

従来知られている，変形型 GCM0, 1 に対する Committing 安全性を破る攻撃計算量と本論文の結果を表 1 にまとめる．変形型 GCM1 の FROB 安全性は文献 [8] で証明されているが，FROB 安全の根拠となっているパラメータが 256 ビットのため，計算量約 2^{128} で動作する自明なバースデー攻撃が存在する．

2. 準備

2.1 記法

非負整数 n に対し，$\{0,1\}^n$ はすべての n ビット列の集合を表す．また，すべてのビット列の集合を $\{0,1\}^*$ と表す．n ビット列 $a, b \in \{0,1\}^n$ の連結を $a \| b$ と表す．n ビットの 0 を 0^n と表す．a, b の排他的論理和 (XOR) を $a \oplus b$ と表す．a, b が入力，出力等のデータ組であるとき，まとめて (a, b) と表す．a のビット長を $|a|$ と表す．非負整数 i, j に対し，a の上位 i バイト目から j バイト目までを $a[i:j]$ と表す．このとき，a の先頭を 0 バイトとして数える．a のビット長をできるだけ少ない 0 パディングで 128 ビットの倍数にする関数を $\mathrm{Pad}(a)$ と表す．有限体 $\mathrm{GF}(2^n)$ の要素 c の逆元を c^{-1} と表す．非負整数 x, s に対し，x の s ビット 2 進数表現を $[x]_s$ と表す．ただし，$x < 2^s$ である．x 以上かつ最小の整数を $\lceil x \rceil$ と表す．x 以下かつ最大の整数を $\lfloor x \rfloor$ と表す．a の上位 x ビットを取り出す関数を $\mathrm{MSB}_x(a)$ と表す．

鍵長 k ビット，ブロック長 n ビットのブロック暗号 E の暗号化関数 $E : (K, M) \mapsto C$ を $E_K(M) = C$ と表す．また，ブロック暗号の復号関数 $E^{-1} : (K, C) \mapsto M$ を $E_K^{-1}(C) = M$ と表す．ただし，$K \in \{0,1\}^k$ は鍵，$M \in \{0,1\}^n$ は平文，$C \in \{0,1\}^n$ は暗号文である．

鍵長 k ビット，タグ長 t ビットの認証暗号 (AEAD: Authenticated Encryption with Associated Data) の暗号化関数 AEAD : $(K, N, A, M) \mapsto (C, T)$ を $\mathrm{AEAD}_K(N, A, M) = (C, T)$ と表す．また，認証暗号の復号関数 $\mathrm{AEAD}^{-1} : (K, N, A, C, T) \mapsto M/\perp$ を $\mathrm{AEAD}_K^{-1}(N, A, C, T) = M/\perp$ と表す．ただし，$K \in \{0,1\}^k$ は鍵，$N \in \{0,1\}^*$ はナンス，A は Associated Data，M は平文，C は暗号文，$T \in \{0,1\}^t$ はタグであり，$|M| = |C|$ である．$\perp$ は拒否を表す記号である．

2.2 Committing 安全性

認証暗号 AEAD : $(K, N, A, M) \mapsto (C, T)$ を考える．Committing (CMT) 安全性とは，敵が

$$\mathrm{AEAD}_K(N, A, M) = \mathrm{AEAD}_{K'}(N', A', M')$$

を満たす入力 $(K, N, A, M), (K', N', A', M')$ の組を求めることが計算量的に困難な性質である．ここで，K, K' は鍵，N, N' はナンス，A, A' は Associated Data，M, M' は平文，C は暗号文，T はタグである．CMT 安全性では，任意の (C, T) と，(C, T) の復号に使う入力が計算量的に対応（コミット）していることを要求する．すなわち，CMT 安全性を満たす認証暗号において，(C, T) にコミットしていない入力のとき，出力が (C, T) となる確率は十分低い．

CMT 安全性には，コミットしている入力の制約に応じたいくつかの定義 (FROB, CMT-1, CMT-2, CMT-3, CMT-4) が存在する．FROB [9] は，AEAD の入力 $(K, N, A, M), (K', N', A', M')$ において，$K \neq K', N = N'$ を制約とする安全性定義である．CMT-1 [10] は，FROB より制限の緩い定義で，$K \neq K'$ を制約としている．CMT-1 は Key-Committing 安全性と同義である．CMT-2 [10], [11], CMT-3 [10], CMT-4 [10] は，CMT-1 を拡張した定義で，それぞれ $(K, N) \neq (K', N'), (K, N, A) \neq (K', N', A'), (K, N, A, M) \neq (K', N', A', M')$ を制約としている．CMT-3 と CMT-4 は等価であることが知られている [10].

2.3 変形型 GCM

変形型 GCM1 の構成と暗号化の擬似コードをそれぞ

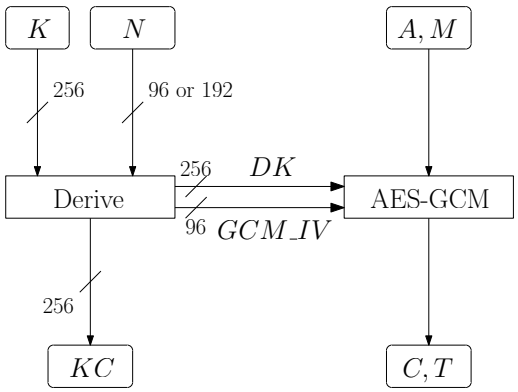

図 1: 変形型 GCM1 の構成（暗号化）

Algorithm 1 変形型 GCM1

Require: K, N, A, M
Ensure: KC, C, T
1: $(DK, KC, GCM\_IV) \leftarrow \text{Derive}(K, N)$
2: $(C, T) \leftarrow \text{AES-GCM}(DK, GCM\_IV, A, M)$
3: **return** (KC, C, T)

れ図 1，アルゴリズム 1 に示す． 暗号化の入力はルート鍵 $K \in \{0,1\}^{256}$，ランダムナンス $N \in \{0,1\}^{96}$ または $N \in \{0,1\}^{192}$，Associated Data A，平文 M である．本論文では $|N| = 192$ に限定する．また，出力は Key Commit 値 $KC \in \{0,1\}^{256}$，暗号文 C ($|C| = |M|$)，タグ T である．入力データの処理関数は Derive 関数（2.4 節）と AES-GCM（2.5 節）の 2 つである．まず，(K, N) を Derive 関数に渡し，AES-GCM のナンスとして用いる GCM_IV，鍵として用いる Derive 鍵 $DK \in \{0,1\}^{256}$，$KC \in \{0,1\}^{256}$ を得る．次に，(DK, GCM_IV, M, A) を AES-GCM に渡し，C, T を得る．最後に，(KC, C, T) を出力する．

復号の入力は，K, N, A, C, T, KC である．また，出力は M または $\perp$ である．まず，暗号化と同様に，(K, N) を Derive 関数に渡し，AES-GCM のナンスとして用いる $GCM\_IV \in \{0,1\}^{96}$，鍵として用いる Derive 鍵 $DK \in \{0,1\}^{256}$，$KC' \in \{0,1\}^{256}$ を得る．次に，入力の KC と先程得た KC' を比較する．$KC = KC'$ であれば，入力を認証し，$M = \text{AES-GCM}^{-1}_{DK}(GCM\_IV, A, C, T)$ を出力する．$KC \neq KC'$ であれば，入力を拒否し，$\perp$ を出力する．

2.4 Derive 関数

Derive 関数 [6], [8] は，$w = 5$ の XORP の構造を利用している．XORP とは，ブロック暗号を用いて，Birthday Bound を超える安全性をもつ高速な擬似ランダム関数である [12]．Birthday Bound とは，暗号方式のセキュリティパラメータが n の場合，その暗号を攻撃するのに必要な計算量が $O(2^{n/2})$ 以上であることを指す．XORP の構成と擬似コードをそれぞれ図 2，アルゴリズム 2 に示す． ブロック暗号を E，鍵を K, フレーム幅を w とする．まず，入力

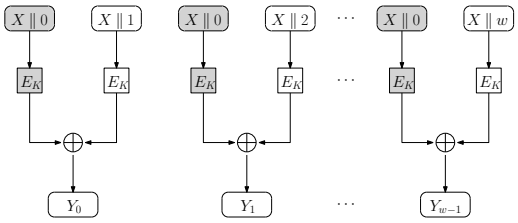

図 2: XORP の構成

Algorithm 2 XORP$[w](K, X)$

Require: w, K, X
Ensure: $Y_0, Y_1, \ldots, Y_{w-1}$
1: **for** $i \leftarrow 0$ to $w - 1$ **do**
2: $\quad Y_i \leftarrow E_K(X \,\|\, [0]_{1+\lfloor \log_2 w \rfloor}) \oplus E_K(X \,\|\, [i+1]_{1+\lfloor \log_2 w \rfloor})$
3: **end for**
4: **return** $Y_0 \,\|\, Y_1 \,\|\, \cdots \,\|\, Y_{w-1}$

Algorithm 3 Derive(K, N)

Require: K, N
Ensure: DK, KC, GCM_IV
1: $NPadded = N \,\|\, (0\text{x}00)^3$
2: $NHead \leftarrow NPadded[0:14]$
3: $NTail \leftarrow NPadded[15:26]$
4: **for** $i \leftarrow 0$ to 4 **do**
5: $\quad B_i \leftarrow NHead \,\|\, KC\_Choice \,\|\, [(LN - 12)]_4 \,\|\, [i]_3$
6: **end for**
7: $DK \leftarrow E_K(B_0) \oplus E_K(B_1) \,\|\, E_K(B_0) \oplus E_K(B_2)$
8: $KC \leftarrow E_K(B_0) \oplus E_K(B_3) \,\|\, E_K(B_0) \oplus E_K(B_4)$
9: $GCM\_IV \leftarrow NTail$
10: **return** (DK, KC, GCM_IV)

X に $1 + \lfloor \log_2 w \rfloor$ ビットのカウンタを連結して得た w 個のデータを E_K に入力する．次に，得られた w 個の出力それぞれに対して $E_K(X \,\|\, [0]_{1+\lfloor \log_2 w \rfloor})$ を足し合わせる．最後に，生成された $Y_0 \,\|\, Y_1 \,\|\, \cdots \,\|\, Y_{w-1}$ を出力する．

Derive 関数の構成と擬似コードをそれぞれ図 3, アルゴリズム 3 に示す． 入力はルート鍵 $K \in \{0,1\}^{256}$，ランダムナンス $N \in \{0,1\}^{192}$ である．また, 出力は AES-GCM のナンスとして用いる $GCM\_IV \in \{0,1\}^{96}$，鍵として用いる Derive 鍵 $DK \in \{0,1\}^{256}$，Key Commit 値 $KC' \in \{0,1\}^{256}$ である．Derive 関数には $LN, KC\_Choice, ConfigByte$ の 3 つのパラメータがある．LN は N のバイト数を表す．$|N| = 192$ なので，$LN = 24$ となる．$KC\_Choice \in \{0,1\}$ は，変形型 GCM1 が Key-Committing 安全性を満たすか否かを指定するパラメータである．$ConfigByte = KC\_Choice \,\|\, [LN - 12]_4 \,\|\, [0]_3$ は，変形型 GCM1 の設定を示す 1 バイトのパラメータである．

まず, 27 バイト（216 ビット）の $NPadded = N \,\|\, (0\text{x}00)^3$ を生成する．$NPadded[0:14]$ を $NHead$，$NPadded[15:26]$ を $NTail$ とする．続いて，$NHead$ を XORP に入力する．このとき，XORP のカウンタは $ConfigByte + i = KC\_Choice \,\|\, [LN-12]_4 \,\|\, [i]_3$ とする．つまり, XORP 内部のブロック暗号 E への入力 B_i は $B_i = NHead \,\|\, KC\_Choice \,\|$

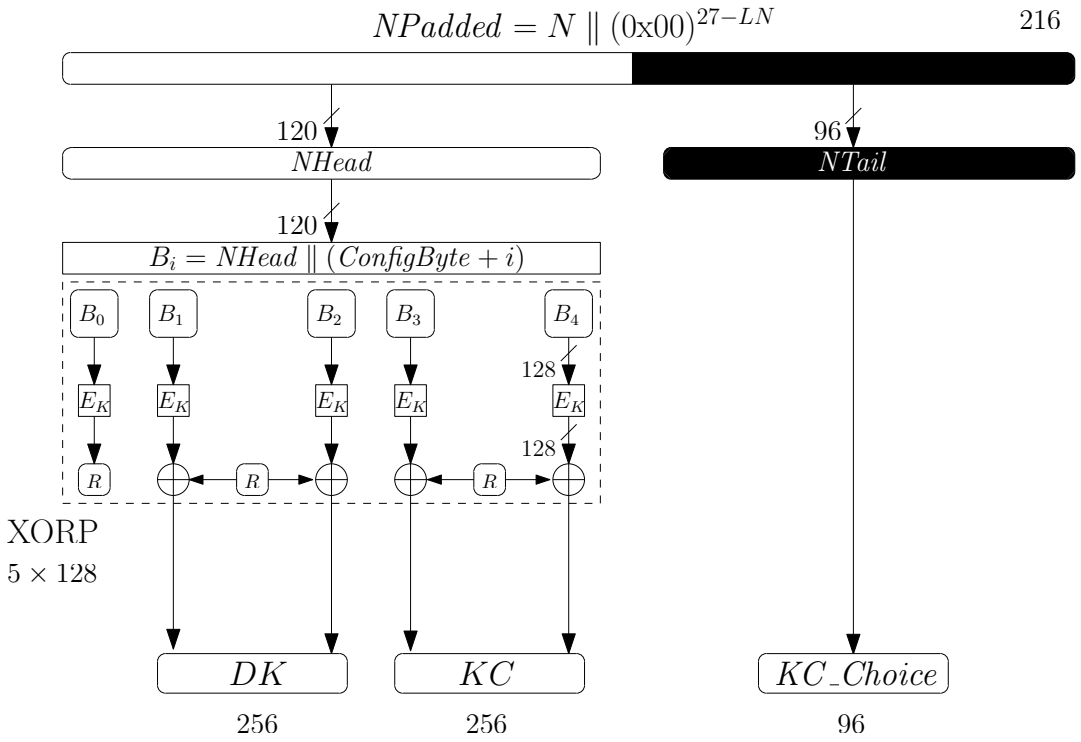

図 3: Derive 関数の構成

Algorithm 4 $\mathrm{GHASH}_L(X)$

Require: X
Ensure: Y
1: $L \leftarrow E_K(0^{128})$
2: $Y \leftarrow 0^{128}$
3: **for** $i \leftarrow 0$ to $m-1$ **do**
4: $\quad Y \leftarrow (Y \oplus X_i) \cdot L$
5: **end for**
6: **return** Y

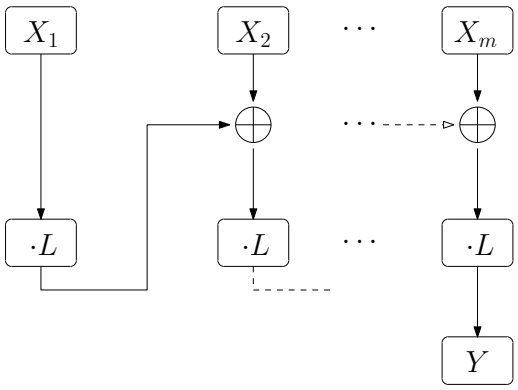

図 4: GHASH の構成

$[LN-12]_4 \parallel [i]_3$ となる．最後に，$(DK, KC, GCM\_IV) = E_K(B_0) \oplus E_K(B_1) \| E_K(B_0) \oplus E_K(B_2), E_K(B_0) \oplus E_K(B_3) \| E_K(B_0) \oplus E_K(B_4), NTail)$ を出力する．$KC\_Choice = 1$ のとき，KC が出力されることを根拠として，変形型 GCM1 は Key-Committing 安全性を満たすとしている [6], [8]．$KC\_Choice = 0$ のとき，KC は空配列として出力されるため，変形型 GCM1 は Key-Committing 安全性を満たさない．本論文では，$KC\_Choice = 1$ の場合に注目する．

2.5 AES-GCM

AES-GCM は，ブロック長 n のブロック暗号 (AES) E，GCTR，GHASH により構成される．GCTR とは，GCM で用いられるカウンタモードである．また，GHASH とは，$\mathrm{GF}(2^n)$ 上で定義される演算を用いた鍵付きの多項式ハッシュ関数である．GHASH の構成と擬似コードをそれぞれ図 4，アルゴリズム 4 に示す． E の鍵を $K \in \{0,1\}^{256}$，入力を $X = X_1 \parallel X_2 \parallel \cdots \parallel X_m$ とおく．ただし，X は m ブロックであり，$X_1, X_2, \ldots, X_m$ はそれぞれ 1 ブロックである．GHASH の鍵は $L = E_K(0^{128})$ と定める．このとき，GHASH の出力 $Y = \mathrm{GHASH}_L(X)$ は式 (1) で与えられる．Y は 1 ブロックである．

$$Y = \sum_{i=1}^{m} X_i \cdot L^{m+1-i} \tag{1}$$

AES-GCM の構成と擬似コードをそれぞれ図 5，アルゴリズム 5 に示す． 暗号化の入力はルート鍵 $K \in \{0,1\}^{256}$，ナンス N，Associated Data A，平文 M である．また，出力は暗号文 C ($|C| = |M|$)，τ (≤ 128) ビットタグ T である．まず，$|N| = 96$ のとき $I = N \parallel 0^{31} \parallel 1$，$|N| \neq 96$ のとき $I = \mathrm{GHASH}_L(N \parallel \mathrm{Pad}(N) \parallel 0^{64} \parallel [N]_{64})$ とする．いずれの場合も $|I| = 128$ である．次に，暗号文 $C = M \oplus \mathrm{MSB}_{|M|}(\mathrm{GCTR}_K(I))$ を生成する．最後に，タグ $T = \mathrm{MSB}_\tau(E_K(I) \oplus \mathrm{GHASH}_L(X))$ を生成し，(C, T) を出力する．ただし，AES-GCM における GHASH の入力 X

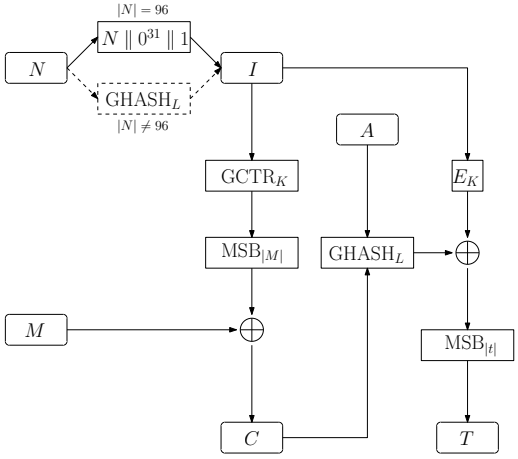

図 5: AES-GCM の構成（暗号化）

Algorithm 5 AES-GCM(K, N, A, M)

Require: K, N, A, M
Ensure: C, T
1: $L \leftarrow E_K(0^{128})$
2: **if** $|N| = 96$ **then**
3: $\quad I \leftarrow N \parallel 0^{31} \parallel 1$
4: **else**
5: $\quad I \leftarrow \mathrm{GHASH}_H(N \parallel \mathrm{Pad}(N) \parallel 0^{64} \parallel [|N|]_{64})$
6: **end if**
7: $C \leftarrow M \oplus \mathrm{MSB}_{|M|}(\mathrm{GCTR}_K(I))$
8: $X \leftarrow A \parallel \mathrm{Pad}(A) \parallel C \parallel \mathrm{Pad}(C) \parallel [|A|]_{64} \parallel [|C|]_{64}$
9: $T \leftarrow \mathrm{MSB}_\tau(E_K(I) \oplus \mathrm{GHASH}_L(X))$
10: **return** (C, T)

の形式は式 (2) で定義される．

$$X = A \parallel \mathrm{Pad}(A) \parallel C \parallel \mathrm{Pad}(C) \parallel [|A|]_{64} \parallel [|C|]_{64} \quad (2)$$

復号の入力は，K, N, A, C, T である．また，出力は M または $\perp$ である．まず，暗号化と同様に I を計算する．次に，入力 K, A, C と I を用いて，タグ $T' = \mathrm{MSB}_\tau(E_K(I) \oplus \mathrm{GHASH}(X))$ を計算する．ここで，入力の T と先程得た T' を比較する．$T = T'$ であれば，入力を認証し，$M = C \oplus \mathrm{MSB}_{|C|}(\mathrm{GCTR}_K(I))$ を出力する．$T \neq T'$ であれば，入力を拒否し，$\perp$ を出力する．

3. 変形型 GCM に対する CMT-2 攻撃

変形型 GCM1 に対する CMT-2 攻撃では，$(K, N) \neq (K', N')$ の制約下で暗号化出力 $(KC, C, T), (KC', C', T')$ を衝突させるような入力を見つけることを目指す．本章では，$K = K'$ であり，N, N' の一部に差分を入力したとき，Derive 関数の出力 KC が必ず衝突することと，AES-GCM は Key-Committing 安全性を満たさないことを利用した CMT-2 攻撃アルゴリズムについて述べる．また，CMT-2 攻撃の入出力の実例を計算機実験により示す．

3.1 アルゴリズム

変形型 GCM1 に対する CMT-2 攻撃をアルゴリズム 6 に示す．まず，KC, KC' を衝突させる．手順 1 では $(DK, KC, GCM\_IV) = \mathrm{Derive}(K, N)$，手順 2 では $(DK', KC', GCM\_IV) = \mathrm{Derive}(K, N')$ を計算する．ただし，$N[0:14] = N'[0:14], N[15:23] \neq N'[15:23]$ とする．アルゴリズム 3 より，先述のように N, N' を設定した場合，$NHead = N'Head, NTail \neq N'Tail$ となる．つまり，$KC = KC', DK = DK', GCM\_IV \neq GCM\_IV'$ となる．

Algorithm 6 変形型 GCM1 に対する CMT-2 攻撃

Require: K, N, N'
Ensure: A, A', M, M'
1: $(DK, KC, GCM\_IV) \leftarrow \mathrm{Derive}(K, N)$ $\{KC, KC'\}$
2: $(DK', KC', GCM\_IV') \leftarrow \mathrm{Derive}(K, N')$
3: $M \xleftarrow{\$} \{0,1\}^a$ $\{C, C'\}$
4: $I \leftarrow GCM\_IV \parallel 0^{31} \parallel 1$
5: $I' \leftarrow GCM\_IV' \parallel 0^{31} \parallel 1$
6: $M' = M \oplus \mathrm{MSB}_{|M|}(\mathrm{GCTR}_{DK}(I)) \oplus \mathrm{MSB}_{|M|}(\mathrm{GCTR}_{DK}(I'))$
7: $A \xleftarrow{\$} \{0,1\}^*$ $\{T, T'\}$
8: $A'_2 \xleftarrow{\$} \{0,1\}^{128}$
9: $C \leftarrow M \oplus \mathrm{GCTR}_{DK}(I)$
10: $C' \leftarrow M' \oplus \mathrm{GCTR}_{DK}(I')$
11: $X' \leftarrow A \parallel \mathrm{Pad}(A) \parallel C \parallel \mathrm{Pad}(C) \parallel [A]_{64} \parallel [C]_{64}$
12: $X'_1 \parallel X'_2 \parallel X'_3 \parallel X'_4 \leftarrow A'_2 \parallel C' \parallel \mathrm{Pad}(C') \parallel [256]_{64} \parallel [C']_{64}$
13: $L \leftarrow E_K(0^{128})$
14: $A'_1 \leftarrow \left(\mathrm{GHASH}_L(X) \oplus A'_2 \cdot L^3 \oplus \sum_{i=1}^{m'-2} X'_i \cdot L^{m'-i}\right) \cdot (L^4)$
15: $A' \leftarrow A'_1 \parallel A'_2$
16: **return** (N, N', A, A', M, M')

次に，C, C' を衝突させる．アルゴリズム 5 より，式 (3) を満たすとき $C = C'$ となる．

$$M \oplus \mathrm{MSB}_{|M|}(\mathrm{GCTR}_K(I)) = M' \oplus \mathrm{MSB}_{|M'|}(\mathrm{GCTR}(I')) \quad (3)$$

K は既に与えられていて，I, I' はそれぞれ GCM_IV, GCM_IV' によって決定されている．よって，a (≤ 128) ビットの M を任意に定め，M' を式 (4) で計算することで，C, C' を衝突させることができる．

$$M' = M \oplus \mathrm{MSB}_{|M|}(\mathrm{GCTR}_K(I)) \oplus \mathrm{MSB}_{|M|}(\mathrm{GCTR}_K(I')) \quad (4)$$

ここで $|M| \leq 128$ とした理由は，T, T' の衝突を効率的に求めるために C, C' を 1 ブロック以内に収めたいからである．

最後に，T, T' を衝突させる．アルゴリズム 5 より，式 (5) を満たすとき $T = T'$ となる．

$$E_K(I) \oplus \mathrm{GHASH}_L(X) = E_K(I') \oplus \mathrm{GHASH}_L(X') \quad (5)$$

ルート鍵は K で固定しているため，GHASH の鍵も $L = E_K(0^{128})$ で固定である．また，I, I' はそれぞれ GCM_IV, GCM_IV' によって決定されている．X は式 (2) のように A, C を用いて決定される．したがって，T, T' を衝突させるためには，A を任意に定めた後，式 (5) を満

たすような X' を求めればよい．X' の形式は式 (6) で定義される．

$$X' = A' \| \mathrm{Pad}(A') \| C' \| \mathrm{Pad}(C') \| [A']_{64} \| [C']_{64} \quad (6)$$

ここで，A' の長さを 2 ブロックとする．A'_1, A'_2 をそれぞれ 1 ブロックとして，$A' = A'_1 \| A'_2$ のように表す．このとき，C' $(= C)$ は既に長さ 1 ブロック以内で決定されているため，式 (6) は 4 ブロックで式 (7) のように表現できる．式 (7) を GHASH に入力したときの出力 Y' は式 (8) のようになる．

$$X' = A'_1 \| A'_2 \| (C' \| \mathrm{Pad}(C')) \| ([256]_{64} \| [C']_{64}) \quad (7)$$

$$\begin{aligned} Y' &= \mathrm{GHASH}_L(X') \\ &= A'_1 \cdot L^4 \oplus A'_2 \cdot L^3 \oplus (C' \| \mathrm{Pad}(C')) \cdot L^2 \\ &\quad \oplus ([256]_{64} \| [C']_{64}) \cdot L \end{aligned} \quad (8)$$

A'_2 を任意に定めることで，式 (8) における未知のパラメータは A'_1 のみとなる．式 (8) を式 (5) に代入して，A'_1 について整理することで式 (9) が得られる．

$$\begin{aligned} A'_1 = \big(&E_K(I) \oplus E_K(I') \oplus \mathrm{GHASH}_L(X) \oplus A' \cdot L^3 \\ &\oplus (C' \| \mathrm{Pad}(C')) \cdot L^2 \oplus ([256]_{64} \| [C']_{64}) \cdot L\big) \cdot \big(L^{-4}\big)^{-1} \end{aligned} \quad (9)$$

以上のように $K, N, N', M, M', A, A' = A'_1 \| A'_2$ を定めることで，変形型 GCM1 に $(K, N, A, M), (K, N', A', M')$ を入力させたときの出力 $(KC, C, T), (KC', C', T')$ を衝突させることができる．

このアルゴリズムでは，各手順で決まった回数の計算しか行っていない．したがって，求めたい $(K, N, A, M), (K, N', A', M')$ を定数の計算量で得られる．

3.2 実例

ルート鍵 $K \in \{0,1\}^{256}$，ナンス $N \in \{0,1\}^{192}$，Associated Data A，平文 M を次のように定める．

$K =$ 0x 0100 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000

$N =$ 0x 0001 0203 0405 0607 0809 0a0b 0c0d 0e0f
1011 1213 1415 1617

$A =$ 0x 01 0000 0011

$M =$ 0x 1100 0001

(K, N, A, M) を変形型 GCM1 に入力したとき，次のような Key Commit 値 KC，暗号文 C，タグ T が得られる．

$KC =$ 0x 2baf 00ef d298 de13 055c 9a6c 39e0 5aee
5715 8338 4357 635e 144f a214 4423 9968

$C =$ 0x 8eee 8a4b

$T =$ 0x 8a1c 8d0c eb7e 07e3 c834 cafe 75aa 001f

$N'Head = NHead, N'Tail \neq NTail$ であるナンス N', A'_1 を式 (9) により計算して A'_2 を任意に定めた $A' = A'_1 \| A'_2$，式 (4) により計算した M' を次に示す．

$N' =$ 0x 0001 0203 0405 0607 0809 0a0b 0c0d 0e0f
1011 1213 1415 1618

$A' =$ 0x 13fb 0a1c 9a13 9cc9 39c4 197e 703a 5116
3333 3333 3333 3333 3333 3333 3333 3333

$M' =$ 0x 728e c953

(K, N', A', M') を変形型 GCM1 に入力したとき，次のような Key Commit 値 KC'，暗号文 C'，タグ T' が得られる．

$KC' =$ 0x 2baf 00ef d298 de13 055c 9a6c 39e0 5aee
5715 8338 4357 635e 144f a214 4423 9968

$C' =$ 0x 8eee 8a4b

$T' =$ 0x 8a1c 8d0c eb7e 07e3 c834 cafe 75aa 001f

$(KC, C, T) = (KC', C', T')$ より，アルゴリズム 6 が変形型 GCM1 の CMT-2 攻撃として動作することが計算機実験により確認できた．

4. まとめと今後の課題

本論文では，ナンス長が 192 ビットの変形型 GCM1 の CMT-2 安全性に対して，計算量 $O(1)$ で攻撃するアルゴリズムとその実例を示した．CMT-2 攻撃は CMT-3, 4 攻撃でもあるため，変形型 GCM1 は CMT-2, 3, 4 の安全性を満たさない．なお，変形型 GCM1 において鍵 K，ナンス N，平文 M を固定し，Associated Data のみに差分を入力することで出力を衝突させることができる．このときの Associated Data を A, A' とおくと，$(K, N, A) \neq (K, N, A')$ のため，これは CMT-3 攻撃である．CMT-3 と CMT-4 は等価であるため [10]，変形型 GCM1 には CMT-2 攻撃ではない CMT-3, 4 攻撃も存在する．

今後の課題として，ナンス長が 192 ビットの変形型 GCM1 の CMT-1 安全性解析や，ナンス長が 96 ビットの変形型 GCM1 に対する Committing 安全性解析が挙げられる．

謝辞．本研究の一部は JSPS 科研費 JP24K07489 の助成を受けたものです．

参考文献

[1] McGrew, D. A. and Viega, J.: The Security and Performance of the Galois/Counter Mode (GCM) of Operation, *Progress in Cryptology - INDOCRYPT 2004, 5th International Conference on Cryptology in India, Chennai, India, December 20-22, 2004, Proceedings* (Canteaut, A. and Viswanathan, K., eds.), Lecture Notes in Computer Science, Vol. 3348, Springer, pp. 343–355 (online), DOI: 10.1007/978-3-540-30556-9_27 (2004).

[2] Kampanakis, P., Halevi, S., Ebeid, N. and Campagna, M.: Blockcipher-Based Key Commitment for Nonce-Derived Schemes, Cryptology ePrint Archive, Paper 2025/758 (2025).
[3] Grubbs, P., Lu, J. and Ristenpart, T.: Message Franking via Committing Authenticated Encryption, *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part III* (Katz, J. and Shacham, H., eds.), Lecture Notes in Computer Science, Vol. 10403, Springer, pp. 66–97 (online), DOI: 10.1007/978-3-319-63697-9_3 (2017).
[4] Gueron, S.: Double-Nonce-Derive-Key-GCM (DNDK-GCM) General Design Paradigms and Application, NIST Workshop on the Requirements for an Accordion Cipher Mode 2024 (2024).
[5] Gueron, S.: Double Nonce Derive Key AES-GCM (DNDK-GCM), Internet-Draft draft-gueron-cfrg-dndkgcm-01, Internet Engineering Task Force (2024).
[6] Gueron, S.: Double Nonce Derive Key AES-GCM (DNDK-GCM), Internet-Draft draft-gueron-cfrg-dndkgcm-02, Internet Engineering Task Force (2025).
[7] Toba, R. and Iwata, T.: DNDK-GCM の Key-Committing 安全性解析, SCIS 2025, 3B2-5 (2025).
[8] Gueron, S. and Ristenpart, T.: DNDK: Combining Nonce and Key Derivation for Fast and Scalable AEAD, Cryptology ePrint Archive, Paper 2025/785 (2025).
[9] Farshim, P., Orlandi, C. and Rosie, R.: Security of Symmetric Primitives under Incorrect Usage of Keys, *IACR Trans. Symmetric Cryptol.*, Vol. 2017, No. 1, pp. 449–473 (online), DOI: 10.13154/TOSC.V2017.I1.449-473 (2017).
[10] Bellare, M. and Hoang, V. T.: Efficient Schemes for Committing Authenticated Encryption, *Advances in Cryptology - EUROCRYPT 2022 - 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, May 30 - June 3, 2022, Proceedings, Part II* (Dunkelman, O. and Dziembowski, S., eds.), Lecture Notes in Computer Science, Vol. 13276, Springer, pp. 845–875 (online), DOI: 10.1007/978-3-031-07085-3_29 (2022).
[11] Takeuchi, R., Todo, Y. and Iwata, T.: Key Recovery, Universal Forgery, and Committing Attacks against Revised Rocca: How Finalization Affects Security, *IACR Trans. Symmetric Cryptol.*, Vol. 2024, No. 2, pp. 85–117 (online), DOI: 10.46586/TOSC.V2024.I2.85-117 (2024).
[12] Iwata, T.: New Blockcipher Modes of Operation with Beyond the Birthday Bound Security, *Fast Software Encryption, 13th International Workshop, FSE 2006, Graz, Austria, March 15-17, 2006, Revised Selected Papers* (Robshaw, M. J. B., ed.), Lecture Notes in Computer Science, Vol. 4047, Springer, pp. 310–327 (online), DOI: 10.1007/11799313_20 (2006).