Vol.2017-DPS-170 No.3
Vol.2017-CSEC-76 No.3
2017/3/2

ホワイトボックス AES 実装の改良

堀田 智彦[1]　双紙 正和[1]

概要：近年，センサーネットワークや IoT (Internet of Things) などが注目されているが，このような状況では，攻撃者が暗号化デバイスを不正に入手し，物理的攻撃を加えることで，秘密鍵を得てしまうという危険性が高くなる．これらの対策のために考案されたのが，ホワイトボックス暗号方式である．ホワイトボックス暗号方式は，攻撃者が暗号化デバイスを完全に制御できるような状況でも，秘密鍵が漏洩しないことを目的としている．しかしながら，これまで発表された多くのホワイトボックス暗号方式に対して，暗号解析が成功している．そこで本研究では，現時点では明示的な攻撃が知られていない Luo らによるホワイトボックス AES 暗号方式を対象として，それらのセキュリティを向上させる技術を提案する．

Some Improvements on a White-Box AES Implementation

Tomohiko Hotta[1]　Masakazu Soshi[1]

Abstract: White-box cryptography provides a method to conceal a secret key embedded in a hardware device for encryption even if an attacker obtains the device in a possibly illegal manner. Unfortunately, many of the proposed techniques for white-box cryptography have been broken by sophisticated and elaborated cryptanalyses. Therefore, in this paper, we propose some techniques to improve security of a white-box AES implementation proposed by Luo et al., which has not yet been successfully attacked, as far as we know.

1. 序論

近年，センサーネットワークや，様々な物やデバイスをインターネットに接続する IoT (Internet of Things) などが注目を浴びている．これにより，ユーザに高度なサービスを提供できる一方で，攻撃者がそれらのネットワークやデバイスにアクセスできる機会が増加することにもなっている．そこで，攻撃者が，(多くは不正な手段により) デバイスを入手できる可能性が高くなっている．

このような状況では，攻撃者は，デバイスの実行を制御してメモリを検査するなど，さまざまな物理的攻撃を実行できるようになる．その結果，デバイスに埋め込まれた秘密鍵が暴露される危険性が高まってきた．その対策として考案されたのが，ホワイトボックス暗号方式である [4], [6].

ホワイトボックス暗号方式は，信頼されないプラットフォーム上で実行されることを想定した，暗号アルゴリズムの実装方式である．たとえそのような環境で実行され攻撃を受けたとしても，秘密鍵が漏洩しないことを目的としている．

しかしながら，ホワイトボックス暗号方式のモデルでは，攻撃者があまりにも有利であり，一般的に，ホワイトボックス実装は非常に困難である．たとえば，AES [8] を対象とした，Chow らによるホワイトボックス実装 [3] は，Billet らにより暗号解析され，2^{30} の時間計算量で鍵を抽出された [2]．その後，Xiao らによって提案されたホワイトボックス AES 暗号化方式 [9] も，Mulder らによって解読されている [7].

そこで本研究では，まず共通鍵暗号方式として AES を対象とし，Luo らによって提案された，ホワイトボックス方式 [5] に着目する．この方式は，著者らの知る限り，本論文執筆時点で明示的に暗号解析されていない．本論文では，Luo らによるホワイトボックス AES 実装方式のセキュリティをさらに向上させる技術を提案する．

本論文は，以下のように構成される．まず，2 節において，AES の概要を紹介する．次に，3 節において，特に Luo らのホワイトボックス AES 実装方式に重点を置きな

[1] 広島市立大学 情報科学部

state ← plaintext
for $r = 1, ..., 9$
    ShiftRows(state)
    AddRoundKey(state, $\hat{k}_{r-1}$)
    SubBytes(state)
    MixColumns(state)
ShiftRows(state)
AddRoundKey(state, $\hat{k}_9$)
SubBytes(state)
AddRoundKey(state, $\hat{k}_{10}$)
ciphertext ← state

図 1 修正 AES 暗号アルゴリズム
Fig. 1 Modified AES Encryption Algorithm

がら，関連研究について議論する．そして，4 節で，Luo らの方式の改良技術を提案する．その後，5 節で提案手法を評価し，最後に 6 節で本論文の結論を述べる．

2. AES の概要

本研究を含めて，ホワイトボックス暗号方式は，AES を対象としていることが多い．そこでまず，この節では AES の概要について述べる．

Advanced Encryption Standard（AES）[8] は，今日最も広く使用されている共通鍵暗号方式である．鍵長として 128, 192, 256 ビットを選択できるが，本論文では，鍵長が 128 ビットの AES-128 を AES として考える．AES はブロック暗号で，ブロック長は 128 ビットである．また，AES-128 には 10 個のラウンドがあり，各ラウンドでは，128 ビットの鍵から作成された部分鍵により暗号化がなされ，128 ビットの**ステート**変数が更新されていく．

Chow らは，AES をホワイトボックス化しやすいように，暗号化変換自体には変更を加えず，その実行順などに修正を加えた [3]．Chow らによる AES 暗号化アルゴリズムを**図 1** に与える．

以下に，図 1 における各操作の概要を述べる．より詳しくは [8] を参照せよ．

AES においては，16 バイトのステートの各バイトは，$GF(2^8)$ の元とみなされ，乗算・加算は，その有限体上で行われる．

ShiftRows は，所定の置換により，ステートの各バイトを入れ替える．*AddRoundKey* は，128 ビットのラウンド鍵 (鍵スケジュールにより，128 ビットの秘密鍵から各ラウンドごとに作成される) の各鍵バイトをステートの各バイトに加算 (排他的論理和) する．*SubBytes* は，非線形変換である．ある代入テーブル (S とする) に基づき，ステートの各バイトを変換する．*MixColumns* は，式 (1) によって定義される行列 MC により，ステートの各 4 バイトを線形変換する．

$$MC := \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \quad (1)$$

MC の要素は，$GF(2^8)$ の元である．

Chow らによって与えられた修正 AES アルゴリズム (図 1) は，その後のホワイトボックス AES 方式において採用されており [5], [9]，本研究でも，AES のアルゴリズムは，図 1 を意味するものとする。

3. 関連研究

この節では，従来のホワイトボックス AES 実装について述べる．

まず，従来のホワイトボックス AES 実装として重要な，Chow らの研究 [3] について述べる．次に，従来のホワイトボックス AES 実装について概観し，最後に，我々の研究が対象とする Luo らによるホワイトボックス AES 実装 [5] について述べる．

3.1 Chow らによるホワイトボックス AES 実装

暗号化は，秘密の鍵に基づいて，平文を暗号文に変換するものである．したがって，理想的には，ホワイトボックス暗号は，ある秘密鍵における平文と暗号文の対応を記録しておくテーブル (配列) を用意することで，実現できる．このようなテーブルを，**ルックアップテーブル**と呼ぶ．

しかしながら，このような理想的なテーブルは，通常，現実には実装できないほど巨大なものになる．たとえば，AES-128 では，理想的なルックアップテーブルのサイズは，$2^{128} \times 128$ ビットになり，現実的には実現不可能なサイズである．

そこで，Chow らは，より小さなルックアップテーブルを用意し，一部のテーブルを秘密鍵依存として構成することにより，ホワイトボックス AES 実装方式を提案した [3]．

Chow らはさらに，ルックアップテーブルの実装の際に，入出力エンコーディングを付加することを提案した．Chow らによる**エンコーディング**とは，可逆な全単射のことである．それを説明するために，ある関数 f について考える．このとき，Chow らは，f をそのままルックアップテーブルとして実装するのではなく，入力エンコーディング f_i，出力エンコーディング f_o を導入し，$f' = f_o \circ f \circ {f_i}^{-1}$ として，ルックアップテーブルを実装することを考えた．こうして，f のルックアップテーブルのセキュリティを向上させることができる．また，ホワイトボックス実装の際には，複数の関数を順に実行することがよくある．この場合，入力に対して関数 f, g を順に実行するとすると，$g \circ f$ という合成関数を実行することになる．このとき，Chow らは，$f' = f_o \circ f \circ {f_i}^{-1}$，$g' = g_o \circ g \circ {g_i}^{-1}$ のように，入出

力エンコーディングを考え，f', g' をそれぞれルックアップテーブルとして実装することを考えた．この際，$g_i = f_o$ のように入出力エンコーディングを採用すれば，

$$g' \circ f' = (g_o \circ g \circ {g_i}^{-1}) \circ (f_o \circ f \circ {f_i}^{-1}) = g_o \circ g \circ f \circ {f_i}^{-1}$$

となり，f の出力エンコーディングと g の入力エンコーディングがキャンセルされ，$g \circ f$ が (それらの入出力エンコーディングはいずれキャンセルさせなければならないものの) 計算できることになる．

以上のように，Chow らによるホワイトボックス AES 実装方式は斬新なアイディアに満ちており，後の研究に大きな影響を与えた．しかし，Chow らの方式に対しては，Billet らによる洗練された代数的な暗号解析が行われ，2^{30} の時間計算量で鍵を抽出されてしまった [2].

3.2 その他のホワイトボックス AES 実装

ホワイトボックス暗号方式のモデルでは，攻撃者があまりにも有利であり，一般的に，ホワイトボックス実装は非常に困難である．たとえば，Xiao らによって提案されたホワイトボックス AES 暗号化方式 [9] は，Mulder らによって解読されている [7].

3.3 Luo らによるホワイトボックス AES 実装

この節では，Luo らによって提案されたホワイトボックス AES 実装 [5] について述べる．

Luo らによるホワイトボックス AES 実装は，共同著者らによって提案された手法 [9] に基づく．[9] は，2012 年に Mulder らによって解読されたため [7]，Luo らは [9] を改良した手法を提案した [5]．以下では，Luo らによるホワイトボックス AES 実装 [5] を，Luo らの方式と呼ぶことにする．

Luo らの方式の概要を，**図 2** に示す．Luo らの方式は，ラウンド 1 からラウンド 10 までは，それぞれ 3 個のステージから構成される．それらのラウンドで，ラウンド 1 のみが異なる構造を持っている．さらに，最後に付加されるラウンド (ラウンド 11 とする) は，2 個のステージを持つ．

各ラウンドにおけるステージは，TSR, TXOR, TXOR3, nTMC という，4 種類のルックアップテーブルから構成される．それぞれの概略を，以下に説明する．詳しくは [5] を参照せよ．

最初に，ステージ 3 における nTMC について述べる．ラウンド r への 128×1 ビット入力を，x と書く．すなわち，$x^\top \in GF(2)^{128}$ である．また，$x^\top = (x_0^\top\, x_1^\top \cdots x_{15}^\top)$ と書く．つまり，$x_i^\top \in GF(2)^8$ $(0 \le i \le 15)$ である．すると，MixColumns の操作は，式 (2) のように書くことができる．

$$\begin{pmatrix} x'_{4i} \\ x'_{4i+1} \\ x'_{4i+2} \\ x'_{4i+3} \end{pmatrix} = MC_0 \begin{pmatrix} x_{4i} \\ x_{4i+1} \end{pmatrix} \oplus MC_1 \begin{pmatrix} x_{4i+2} \\ x_{4i+3} \end{pmatrix} \quad (2)$$

式 (2) においては，$0 \le i \le 3$ である．また，MC_0, MC_1 は，それぞれ MC の最初の 2 個の列，MC の最後の 2 個の列を表す．すなわち，

$$MC_0 := \begin{pmatrix} 02 & 03 \\ 01 & 02 \\ 01 & 01 \\ 03 & 01 \end{pmatrix}, \quad MC_1 := \begin{pmatrix} 01 & 01 \\ 03 & 01 \\ 02 & 03 \\ 01 & 02 \end{pmatrix} \quad (3)$$

である．

以上の MC_0, MC_1 の定義のもとに，ラウンド r $(0 \le r \le 9)$ での 16×32 ビットルックアップテーブル[*1] nTMC_i^r は，**図 3** のように示される．

図 3 において，in, out は，それぞれ 4 ビットの入力・出力エンコーディングである．また，L_i^r $(0 \le i \le 7)$, R_i^r $(0 \le i \le 3)$ は，それぞれ 16 ビット，32 ビットの線型全単射である．また，S は S-box を表し，$\hat{k}_i^r$ は，ラウンド r におけるラウンド鍵の $(i+1)$ バイト目を表す $(0 \le i \le 15)$.

ステージ 1 における TSR は，ShiftRows を実現するための，8×128 ビットのルックアップテーブル $\mathrm{TSR}_{i,j}^r$ 16 個から構成される ($r = 1$ のとき，$0 \le i \le 3$, $0 \le j \le 3$. また，$2 \le r \le 11$ のとき，$0 \le i \le 3$, $0 \le j \le 7$)．ラウンド r $(2 \le r \le 10)$ における $\mathrm{TSR}_{i,j}^r$ を，**図 4** に示す．

図 4 において，$(R_i^r)_j^{-1}$ $(0 \le j \le 7)$ は，$(R_i^r)^{-1}$ における 32×4 ビットの部分行列である．すなわち，$(R_i^r)^{-1} = ((R_i^r)_0^{-1} \cdots (R_i^r)_7^{-1})$ と書ける．さらに，$LL^r := \mathrm{diag}(L_0^r, ..., L_7^r)$ である．また，SR を，ShiftRows を実現する，128×128 ビット行列とする．すると，SR_i $(0 \le i \le 3)$ は，SR における 128×32 ビット行列である．すなわち，$SR = (SR_0 \cdots SR_3)$ と書ける．

最後に，ステージ 2 におけるルックアップテーブル TXOR, TXOR3 について述べる．ステージ 1 には 32 個の TSR があり (ラウンド 1 のみ TSR は 16 個)，それぞれ 128 ビットずつ出力する．これらの出力を加算して 128 ビットの値とするのがステージ 2 の TXOR (416 個，それぞれ 8 ビット入力，4 ビット出力) と TXOR3 (288 個，それぞれ 12 ビット入力，4 ビット出力) である．TXOR, TXOR3 については構造が単純なので，図は省略する．

Luo らの方式は，著者らの知る限り，本論文執筆時点では，明示的な攻撃法は知られていない．そこで本研究では，Luo らの方式を改良するいくつかの技術について提案する．

[*1] 16 ビット入力，32 ビット出力のテーブルのことである．その実際のサイズは，$2^{16} \times 32$ ビットである．

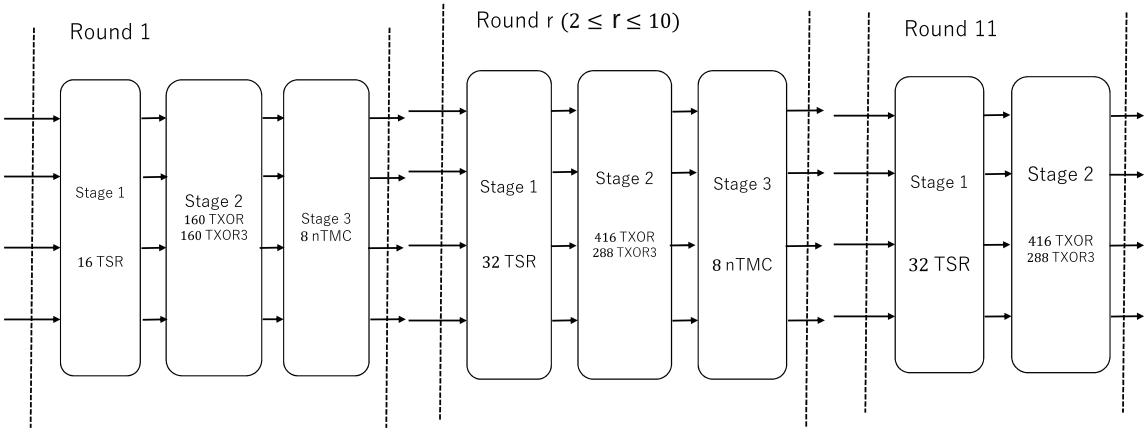

図 2 Luo らの方式の構成
Fig. 2 The Structure of the Approach of Luo et al.

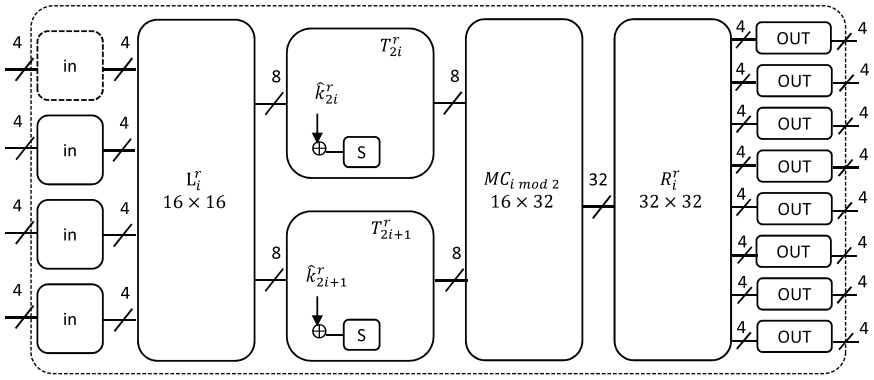

図 3 nTMC_i^r の構成 $(1 \le r \le 9, 0 \le i \le 7)$
Fig. 3 The Structure of nTMC_i^r $(1 \le r \le 9, 0 \le i \le 7)$

4. 提案手法

この節では，Luo らによるホワイトボックス AES 実装 [5] のセキュリティを向上させるための技術について提案する．

4.1 基本的なアイディア

Luo らの方式のセキュリティを向上させるため，まずラウンド r における nTMC_i^r の構成について考える (図 3, $1 \le r \le 9, 0 \le i \le 7$)．ここで，3.3 節と同様に x を MixColumns への 128 ビット入力とすると，MixColumns による乗算は，以下のように 4 個の 32 ビット値の加算に分解できることに注意せよ $(0 \le j \le 3)$.

$$MC \begin{pmatrix} x_{4j} \\ x_{4j+1} \\ x_{4j+2} \\ x_{4j+3} \end{pmatrix} = x_{4j} \begin{pmatrix} 02 \\ 01 \\ 01 \\ 03 \end{pmatrix} \oplus x_{4j+1} \begin{pmatrix} 03 \\ 02 \\ 01 \\ 01 \end{pmatrix} \oplus x_{4j+2} \begin{pmatrix} 01 \\ 03 \\ 02 \\ 01 \end{pmatrix} \oplus x_{4j+3} \begin{pmatrix} 01 \\ 01 \\ 03 \\ 02 \end{pmatrix} \quad (4)$$

加算は可換であり，式 (4) の右辺における 4 個の値を任意の順で加算できる．そこで，各ラウンドの nTMC で，この加算の順番を任意に変更することで，Luo らの方式のセキュリティを向上させることを試みる．

また，式 (4) においては，排他的論理和 (加算) の性質より，任意の数を偶数回加えても値が変わらないことに注意せよ．

以上の基本的なアイディアに基づいて，Luo らの方式を改良する．

4.2 MixColumns の再構成

4.1 節で述べたアイディアに基づき，提案方式では，各

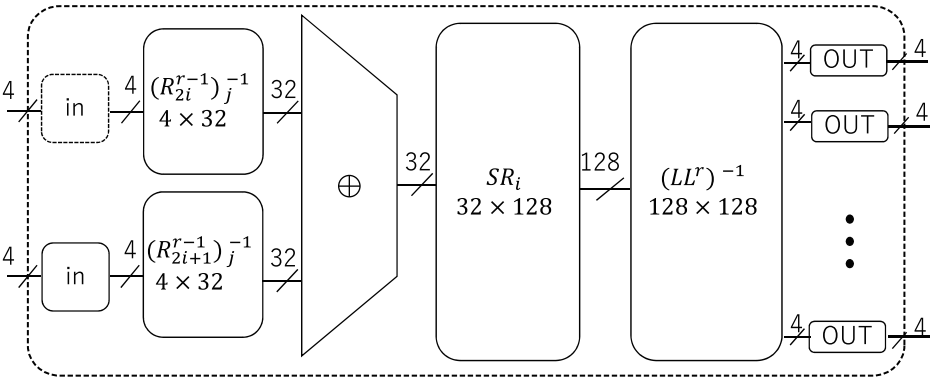

図 4 $\mathrm{TSR}_{i,j}^r$ の構成 $(2 \le r \le 10)$
Fig. 4 The Structure of $\mathrm{TSR}_{i,j}^r$ $(2 \le r \le 10)$

ラウンドで MixColumns MC が再構成される．すなわち，ラウンド r における式 (2) を，以下のような同値の式に変換する $(1 \le r \le 9, 0 \le i \le 3)$.

$$\begin{pmatrix} x'_{4i} \\ x'_{4i+1} \\ x'_{4i+2} \\ x'_{4i+3} \end{pmatrix} = \left(MC_{\sigma_i^r(0)+1,\sigma_i^r(1)+1} \begin{pmatrix} x_{4i+\sigma_i^r(0)} \\ x_{4i+\sigma_i^r(1)} \end{pmatrix} \oplus \alpha_i^r \right) \oplus \left(MC_{\sigma_i^r(2)+1,\sigma_i^r(3)+1} \begin{pmatrix} x_{4i+\sigma_i^r(2)} \\ x_{4i+\sigma_i^r(3)} \end{pmatrix} \oplus \alpha_i^r \right) \tag{5}$$

ここで，σ_i^r は，集合 $\{0,1,2,3\}$ における置換である．また，$MC_{a,b}$ $(a,b \in \{1,2,3,4\})$ は，MC における a 列めと b 列めからなる，4 行 2 列の行列である．たとえば，

$$MC_{2,4} := \begin{pmatrix} 03 & 01 \\ 02 & 01 \\ 01 & 03 \\ 01 & 02 \end{pmatrix}$$

となる．また，$\alpha_i^r \in_R GF(2)^{32}$ で，一様ランダムであるとする．

そして，式 (5) の右辺の第 1 項，第 2 項にそれぞれ基づいて，nTMC を構成すればよい．その際，S-box で加算されるラウンド鍵は，$x_{4i+\sigma_i^r(k)}$ $(0 \le k \le 3)$ に対応するようにする．

4.3 提案手法の構成

これまでの議論に基づき，我々が提案する手法の構成を**図 5** に示す．Luo らの方式と，我々の方式の違いは，ラウンド 1 からラウンド 10 までの，ステージ 3 とステージ 4 の構成である．それぞれを，以下に述べていく．

4.3.1 ステージ 3 の構成

ステージ 3 は，直感的に言えば，4.2 節で述べた，置換 σ_i^r を実現するためのものである．このステージは，2 種類のルックアップテーブル LST, TST からなる．それらの間の入出力の関係を**図 6** に記す．

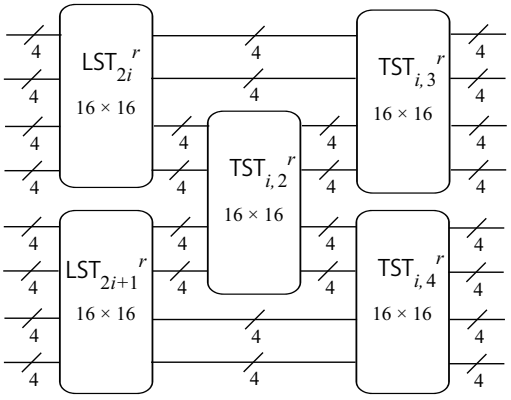

図 6 ステージ 3 の構成 $(0 \le i \le 7)$
Fig. 6 The Structure of Stage 3 $(0 \le i \le 7)$

LST, TST の詳細については以下に述べる．

4.3.1.1 LST

LST_{2i}^r, LST_{2i+1}^r はいずれも，16 ビットの入力に対し，16 ビットの値を出力する (対応付ける) ルックアップテーブルである．したがってそのサイズは，$2^{16} \times 16$ ビットである．LST_{2i}^r, LST_{2i+1}^r の構成を，**図 7** に表す．

図 7 において，in/out は，4 ビットの入力 / 出力エンコーディングを表す．また，L_{2i}^r, L_{2i+1}^r は，[5] と同様，16 ビットの入力エンコーディングであり，ラウンド r のステージ 1 (TSR) における出力エンコーディング (128 ビット全体で，LL_r^{-1} となる) をキャンセルする．また，$ST_{i,0}^r$, $ST_{i,1}^r$ は，16 ビットの全単射であり，4.3.1.3 節で詳しく述べる．

4.3.1.2 TST

$\mathrm{TST}_{i,k}^r$ $(0 \le i \le 7, k = 2,3,4)$ は，16 ビットの入力に対し，16 ビットの値を出力するルックアップテーブルである．したがってそのサイズは，$2^{16} \times 16$ ビットである．$\mathrm{TST}_{i,k}^r$ $(k = 2,3,4)$ の構成を，**図 8** に表す．

$ST_{i,k}^r$ $(k = 2,3,4)$ は，16 ビットの全単射であり，4.3.1.3 節で詳しく述べる．

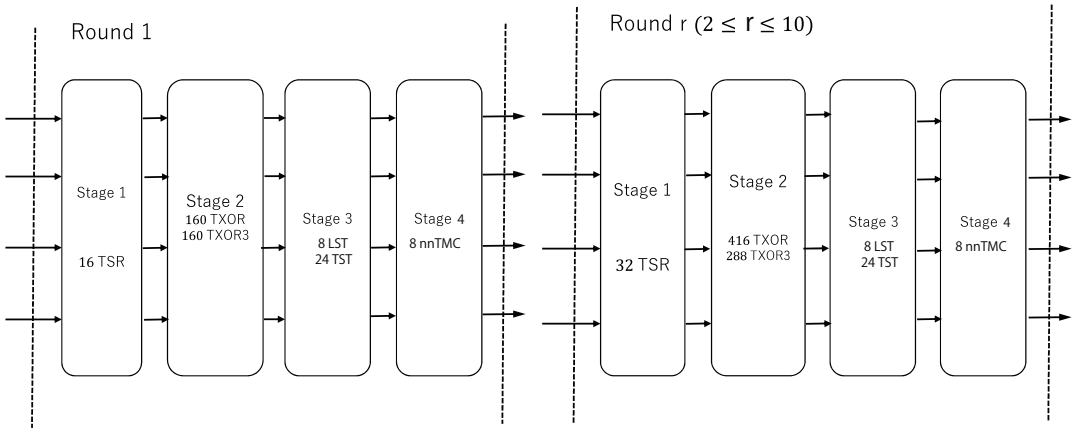

図 5 提案手法の構成
Fig. 5 The Proposed Structure

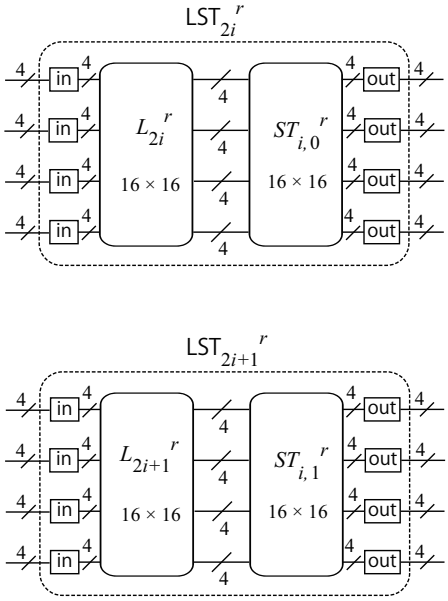

図 7 LST^r_{2i}, LST^r_{2i+1} の構成 $(0 \le i \le 7)$
Fig. 7 The Structure of LST^r_{2i} and LST^r_{2i+1} $(0 \le i \le 7)$

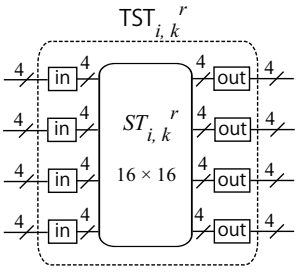

図 8 $\mathrm{TST}^r_{i,k}$ の構成 $(0 \le i \le 7,\ k = 2, 3, 4)$
Fig. 8 The Structure of $\mathrm{TST}^r_{i,k}$ $(0 \le i \le 7,\ k = 2, 3, 4)$

4.3.1.3 ST

$ST^r_{i,k}$ $(0 \le k \le 4)$ は，それぞれ 16 ビットの全単射であり，4.2 節で述べた，置換 σ^r_i を実現する．

以下では，[8] の記述と同様に，16 ビットの値について，LSB (least significant bit) を 0 ビット目とし，MSB (most significant bit) を 15 ビット目とする．さらに，16 ビットの値 x について，a ビット目から b ビット目までの $(a \le b$ とする)，連続する $b-a+1$ ビットを，$x[a:b]$ と書くことにする．

ここで，ステージ 3 において，L^r_{2i} の出力を x_{4i}, x_{4i+1}，また，L^r_{2i+1} の出力を x_{4i+2}, x_{4i+3}，とそれぞれ書くことができる $(0 \le i \le 7,\ x^\top_j \in GF(2)^8,\ 0 \le j \le 31)$．すなわち，$x_{4i}, x_{4i+1}$ が $ST^r_{i,0}$ への入力で，x_{4i+2}, x_{4i+3} が $ST^r_{i,1}$ への入力となる．

以上の準備のもとに，$ST^r_{i,k}$ $(0 \le k \le 4)$ の入出力の関係を考える．なお，LST, TST において，対応する出力エンコーディングと入力エンコーディングの効果はキャンセルされるので，$ST^r_{i,k}$ 間の入出力の関係のみに着目すればよい．

そこで，図 6，図 7，図 8 より，$ST^r_{i,k}$ $(0 \le k \le 4)$ は，式 (6) を満たす必要があることが分かる．

$$
\begin{aligned}
&(x_{4i+\sigma^r_i(0)}, x_{4i+\sigma^r_i(1)}, x_{4i+\sigma^r_i(2)}, x_{4i+\sigma^r_i(3)}) = \\
&\quad (ST^r_{i,3}(ST^r_{i,0}(x_{4i}, x_{4i+1})[0:7], \\
&\qquad ST^r_{i,2}(ST^r_{i,0}(x_{4i}, x_{4i+1})[8:15], \\
&\qquad\quad ST^r_{i,1}(x_{4i+2}, x_{4i+3})[0:7])[0:7]), \\
&\quad ST^r_{i,4}(ST^r_{i,2}(ST^r_{i,0}(x_{4i}, x_{4i+1})[8:15], \\
&\qquad ST^r_{i,1}(x_{4i+2}, x_{4i+3})[0:7])[8:15], \\
&\qquad\quad ST^r_{i,1}(x_{4i+2}, x_{4i+3})[8:15])) \qquad (6)
\end{aligned}
$$

4.3.2 ステージ 4 の構成

ステージ 4 は，我々が nnTMC と呼ぶ，16 ビット入力，32 ビット出力のルックアップテーブルから構成される．nnTMC の構成を，**図 9** に与える．なお，図を簡単にするため，図 9 では，σ^r_i を単に σ と書いている．

5. 考察

この節では，提案手法に対して，セキュリティやテーブルサイズの増加について考察する．

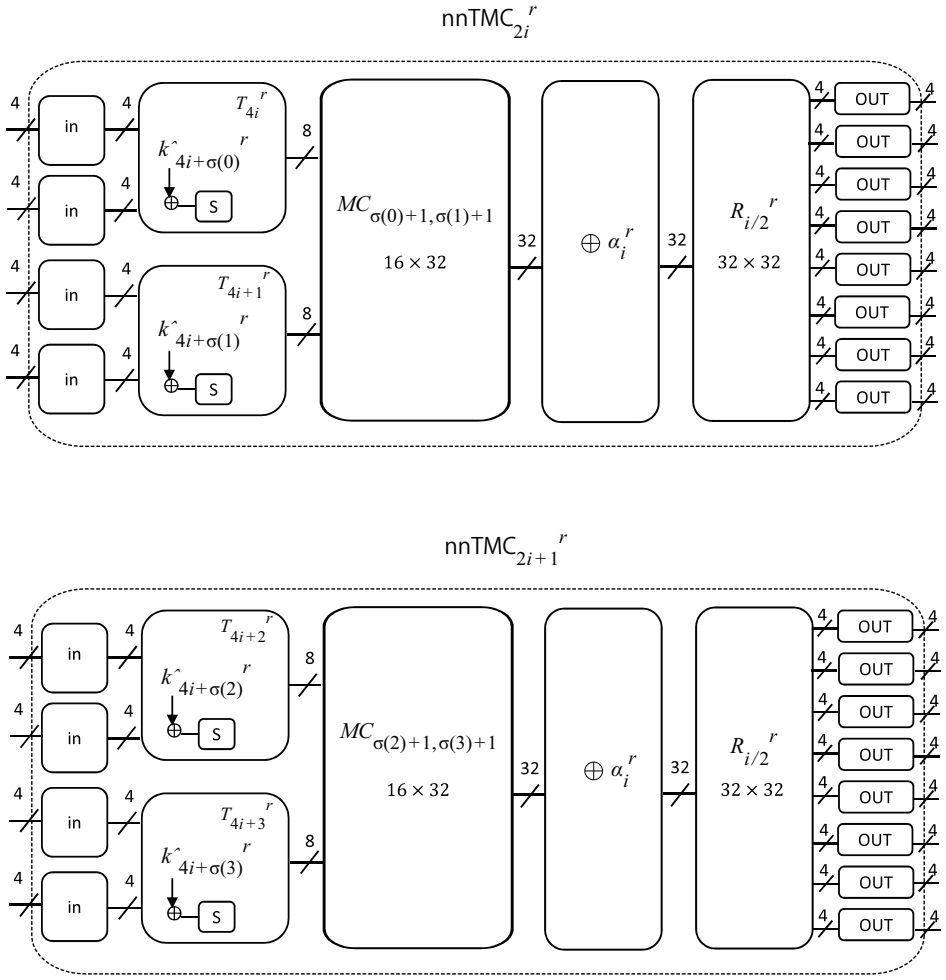

図 9 nnTMC_{2i}^{r}, $\mathrm{nnTMC}_{2i+1}^{r}$ の構成 $(0 \le i \le 7)$
Fig. 9 The Structure of nnTMC_{2i}^{r} and $\mathrm{nnTMC}_{2i+1}^{r}$ $(0 \le i \le 7)$

5.1 提案手法のセキュリティ

提案手法のセキュリティについて，以下に簡単に考察する．詳細な評価は今後の課題である．

まず，Billet らの攻撃 [2] は，[5] と同様の理由で，適用できないと考えられる．

次に，Luo らの方式がベースにした Xiao らのホワイトボックス方式 [9] は，3.2 節で述べたように，Mulder らによって，暗号解析された [7]．Mulder らの解析は，Linear Equivalence (LE) Algorithm とよばれるアルゴリズムを利用するものである．これは，ある関数 F について，$F' = B \circ F \circ A$ となるような線型変換 A, B を求めるものである．ここで，大まかに言うと，Xiao らの TMC は図 3 と同様の構造を持っており，S-box を 2 個並べたものを F と考えることで，$B \circ F \circ A$ という構成であることが分かる．この性質を利用して，Mulder らは Xiao らの方式を攻撃することに成功した．

このような Mulder らの暗号解析については，Luo らは，Luo らの方式における nTMC には適用できないと主張しており [5]，我々の提案方式についても，同様の理由で適用できないと考えられる．さらに我々の提案方式における nnTMC では，MixColumns の後に乱数 α_i^r が加算されることで，Mulder らの暗号解析に対する耐性は高まっていると考えられる．

また，Baek らは，ホワイトボックス AES 実装のより一般的なモデルを考え，それに対する攻撃方法および解決策を提案した [1]．しかしながら，[1] においては，Xiao らの方式 [9] については引用されているものの，Luo らの方式 [5] については考察されていない．したがって，Baek らの暗号解析が，Luo らの方式に対して如何に有効であるかは不明である．我々の提案方式においても同様であると考えられる．

最後に，提案方式の一般的なセキュリティについて簡単に検討する．Luo らの方式に対して，ある攻撃 A が成功したとする．

まず，提案方式の LST, TST については，その入出力エンコーディングのために，対応する ST を推測することは困難であると考えられる．しかしながら，たとえ ST が分からなくても，攻撃者は，$x_{4i+\sigma_i^r(0)}, x_{4i+\sigma_i^r(1)}$ がどれ

であるかは推測できる．このとき，式 (5) から明らかなように，$x_{4i+\sigma_i^r(0)}$, $x_{4i+\sigma_i^r(1)}$ の順番は本質的には問題ではない．$x_{4i+\sigma_i^r(2)}$, $x_{4i+\sigma_i^r(3)}$ の順番についても同様である．したがって，攻撃者は，$1/\binom{4}{2} = 1/6$ の確率で，$x_{4i+\sigma_i^r(0)}$, $x_{4i+\sigma_i^r(1)}$ のペア (順不同)，および $x_{4i+\sigma_i^r(2)}$, $x_{4i+\sigma_i^r(3)}$ のペア (順不同) を推測でき，それをもとに A を実施できる可能性がある．

さらに攻撃者は，32 ビットの乱数 α_i^r を推測しなければならない．その成功確率は，$1/2^{32}$ である．

以上まとめて, 提案方式では, A の成功確率を, $1/(6{\cdot}2^{32}) = 1/(3 \cdot 2^{33})$ 倍にでき，Luo らの方式のセキュリティを大きく高めることができると期待できる．

5.2 ルックアップテーブルのサイズ

図 5 から分かるように，Luo らの方式と比較すると，提案方式では，ステージ 3 のためにルックアップテーブルが新たに必要になる．すなわち，各ラウンドで，8 個の LST, 24 個の TST が新たに必要である．LST も TST も，それぞれ 16 ビット入力，16 ビット出力であるから，そのサイズは，$2^{16} \times 16$ ビットである．全体で 10 ラウンドあるので，まとめると，

$$2^{16} \times 16 \times (8+24) \times 10 = 5 \times 2^{26}(\text{ビット}) = 40(\text{M バイト})$$

のテーブルが新たに必要になる．

6. 結論

ホワイトボックス暗号方式は，攻撃者が暗号化デバイスを完全に制御できるような状況下でも，秘密鍵が漏洩しないことを目的としている．本研究では，現時点では明示的な攻撃が知られていない Luo らによるホワイトボックス AES 暗号方式を対象として，それらのセキュリティを向上させる技術を提案した．

謝辞

本研究は科学研究費補助金 (課題番号 15K00189) の助成，および国立研究開発法人科学技術振興機構（JST）の国際科学技術協力基盤整備事業の支援を受けたものである．

参考文献

[1] Baek, C. H., Cheon, J. H. and Hong, H.: White-box AES Implementation Revisited, *Journal of Communications and Networks*, Vol. 18, No. 3, pp. 273–287 (2016).
[2] Billet, O., Gilbert, H. and Ech-Chatbi, C.: Cryptanalysis of a White Box AES Implementation, *Selected Areas in Cryptography (SAC)*, Lecture Notes in Computer Science, Vol. 3357, Springer-Verlag, pp. 227–240 (2005).
[3] Chow, S., Eisen, P., Johnson, H. and Oorschot, P. C.: White-Box Cryptography and an AES Implementation, *Selected Areas in Cryptography (SAC)*, Lecture Notes in Computer Science, Vol. 2595, Springer-Verlag, pp. 250–270 (2002).
[4] Joye, M.: On White-Box Cryptography, *Proceedings of the First International Conference on Security of Information and Networks (SIN 2007)*, Trafford Publishing, pp. 7–12 (2008).
[5] Luo, R., Lai, X. and You, R.: A New Attempt of White-box AES Implementation, *2014 IEEE International Conference on Security, Pattern Analysis, and Cybernetics (SPAC)*, pp. 423–429 (2014).
[6] Muir, J. A.: A Tutorial on White-Box AES, *Mathematics in Industry*, Vol. 18, pp. 209–229 (2013).
[7] Mulder, Y. D., Roelse, P. and Preneel, B.: Cryptanalysis of the Xiao – Lai White-Box AES Implementation, *Selected Areas in Cryptography (SAC)*, Lecture Notes in Computer Science, Vol. 7707, Springer-Verlag, pp. 34–49 (2012).
[8] National Institute of Standards and Technology (NIST): Advanced Encryption Standard (AES), Federal Information Processing Standards Publication 197 (FIPS-197) (2001).
[9] Xiao, Y. and Lai, X.: A Secure Implementation of White-Box AES, *2nd International Conference on Computer Science and its Applications (CSA '09)*, pp. 1–6 (2009).