Vol.2015-SLDM-170 No.47
Vol.2015-EMB-36 No.47
2015/3/7

鍵長に依存しないLED暗号に対するスキャンベース攻撃

藤代 美佳[1]　柳澤 政生[2]　戸川 望[1]

概要：スマートカード等において利用される軽量ブロック暗号に LED 暗号があり，LED 暗号へのスキャンベース攻撃が報告されている．しかし，この手法では LED 暗号の鍵長を 64 ビットとしており他の鍵長を考慮していない．他の鍵長の場合，秘密鍵を解読できない．本稿では，LED 暗号の鍵長が 64 ビットより大きい場合のスキャンベース攻撃手法を提案する．計算機実験では，暗号回路のみをスキャンチェインに含む場合，提案手法を用いて平均 145 個の平文で 128 ビットの秘密鍵を復元可能と確認した．

キーワード：LED 暗号，サイドチャネル攻撃，スキャンチェイン，スキャンベース攻撃

Improved scan-based side-channel attack on the LED block cipher

Fujishiro Mika[1]　Yanagisawa Masao[2]　Togawa Nozomu[1]

Abstract: LED (Light Encryption Device) block cipher, one of lightweight block ciphers, is very compact in hardware. Although the conventional scan-based side-channel attack method on the LED can retrieve a 64-bit secret key, it would not retrieve a 128-bit secret key. In this paper, an improved scan-based attack method on the LED block cipher is proposed. Experimental results show that our proposed method successfully retrieves its 128-bit secret key using 145 plaintexts on average if the scan chain is only connected to the LED block cipher.

Keywords: LED, Light Encryption Device, side-channel attacks, scan chain, scan-based attack

1. はじめに

LSI の大規模化や微細化，高性能化に伴い，LSI の動作のテストや検証のコストを削減するテスト容易化設計 (DFT: Design for Test) が重要になっている．テスト容易化設計の 1 つにスキャンチェインを用いるスキャンパステストがある．スキャンチェインは LSI 内部のレジスタを直列に接続し，外部から直接，制御・観測可能にしたものである．スキャンチェインを利用することで LSI のテスト効率を高めることができる．スキャンパステストでは，スキャン・インで LSI 内部のレジスタにテストパターンを設定し，LSI を動作させた後，スキャン・アウトでレジスタ値を取得し，期待値と比較する．スキャン・アウトで取得したレジスタ値をスキャンデータという．

一方で，スマートカード等に組み込まれた暗号回路のハードウェアの特性を利用するサイドチャネル攻撃が広く研究されている．暗号回路の物理的な特性はサイドチャネル情報と呼ばれ，消費電力，タイミング情報，故障情報等がある．サイドチャネル攻撃では，これらを外部から観察し解析することで，暗号回路が保護する秘密情報を解読する．サイドチャネル攻撃の中にテスト用のスキャンチェインを利用するスキャンベース攻撃がある．スキャンベース攻撃は，暗号 LSI から暗号処理中のレジスタ値をスキャンデータとして取得し，解析することで秘密情報を取得する．

情報を保護するために暗号は必要不可欠であり，保護する情報の種類や実装するデバイスに合った暗号方式や暗号アルゴリズムが提案され，各分野で実際に採用されている．低消費電力が望まれ，リソースに高い制約があるセンサ等の装置においては，軽量ブロック暗号が望まれる．軽量暗

[1] 早稲田大学大学院基幹理工学研究科情報理工学専攻
Dept. of Computer Science and Engineering, Waseda University.
[2] 早稲田大学大学院基幹理工学研究科電子光システム学専攻
Dept. of Electronic and Photonic Systems, Waseda University.

Vol.2015-SLDM-170 No.47
Vol.2015-EMB-36 No.47
2015/3/7

号は低コスト，低消費電力であるため，スマートカードやRFID タグ等にも用いられる．最軽量ブロック暗号の 1 つに LED 暗号 [1] がある．小面積でハードウェアに実装可能なことが特徴である．LED 暗号は AES に似た構造であり，分割・転置等を実行するラウンド処理と鍵との加算を繰り返す．

LED 暗号へのスキャンベース攻撃手法として [2,3] がある．しかし，この手法は秘密鍵長を 64 ビットとしており，他の鍵長の場合，秘密鍵を正しく解読できるとは限らない．本稿では，鍵長に依存しない LED 暗号へのスキャンベース攻撃手法を提案する．計算機実験では，暗号回路のみをスキャンチェインに含む場合，提案手法を用いて平均 145 個の平文で 128 ビットの秘密鍵を復元可能と確認した．

2. LED 暗号 [1]

本章では LED 暗号の概要とアルゴリズムを示す．LED (Light Encryption Device) 暗号は 2011 年に Guo らが提案した 64 ビットブロック暗号である．ブロック暗号の中でも最軽量でハードウェアへの実装面では小面積で済む点が利点である．秘密鍵長は 64 ビットから 128 ビットである．AES に似た構造であるが AES より小面積であり，AES-256 等の暗号への攻撃手法として効果的な関連鍵攻撃 [4,5] に対し，耐性がある．

2.1 LED 暗号化処理

LED 暗号の演算処理単位は 4 ビットであるため，64 ビットのデータを 4 ビットで 1 要素とした 4×4 行列でデータを表現する．64 ビットの平文ブロックを $m_0 \parallel m_1 \parallel \ldots \parallel m_{15}$ とする時，以下のように表せる．

$$\begin{bmatrix} m_0 & m_1 & m_2 & m_3 \\ m_4 & m_5 & m_6 & m_7 \\ m_8 & m_9 & m_{10} & m_{11} \\ m_{12} & m_{13} & m_{14} & m_{15} \end{bmatrix}$$

LED 暗号ではラウンド処理を繰り返し実行する．4 ラウンドで 1 ステップとし，ステップ毎に 64 ビットの副鍵 SK^i と暗号処理データを排他的論理和する Add RoundKey を実行する．SK^0 は平文と排他的論理和され，SK^1 は 1 ステップ終了後のデータと排他的論理和され，SK^2 は 2 ステップ終了後のデータと排他的論理和される．暗号化処理するステップ数は秘密鍵長によって決定される．秘密鍵長が 64 ビットの時には 8 ステップ，128 ビットの時には 12 ステップ暗号化処理を実行する．図 1 に概略を示す．図 1 において P は平文，C は暗号文を示す．

副鍵 SK^i は 64 ビットであり i 番目の副鍵 SK^i は以下のように表せる．

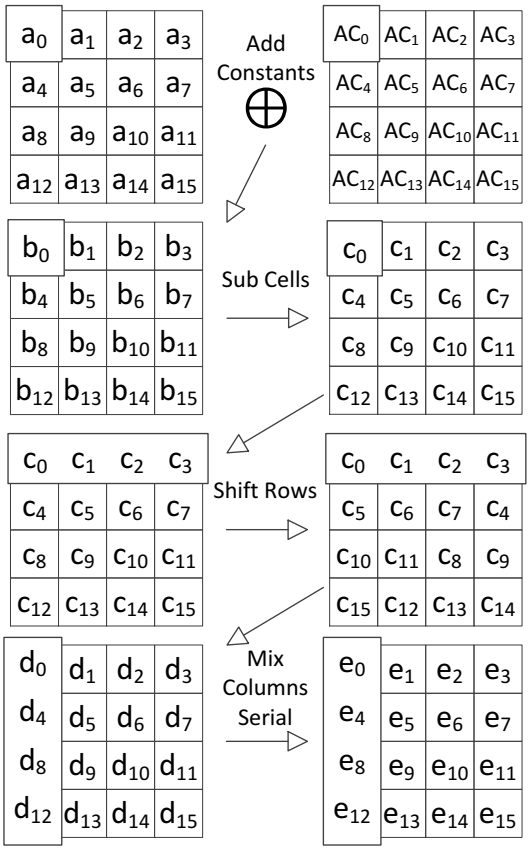

図 2　LED 暗号のラウンド処理.

$$\begin{bmatrix} sk_0^i & sk_1^i & sk_2^i & sk_3^i \\ sk_4^i & sk_5^i & sk_6^i & sk_7^i \\ sk_8^i & sk_9^i & sk_{10}^i & sk_{11}^i \\ sk_{12}^i & sk_{13}^i & sk_{14}^i & sk_{15}^i \end{bmatrix}$$

ここで l ビットの秘密鍵 K の各ビットを $k_0, k_1, \ldots, k_{l-1}$ とすると，

$$sk_j^i = k_{j+i\times 16 mod\ l}$$

である．秘密鍵が 64 ビットの時，副鍵 $SK^i(0 \le i \le 8)$ は全て秘密鍵と等しくなる．秘密鍵が 128 ビットの時，副鍵 $SK^i(0 \le i \le 12)$ は秘密鍵の前半，後半の値と交互に等しくなる．

2.2 ラウンド処理

本節では，LED 暗号処理において繰り返し実行されるラウンド処理を説明する．図 2 に LED 暗号のラウンド処理を示す．ラウンド処理では，Add Constants, Sub Cells, Shift Rows, Mix Columns Serial を実行する．

Add Constants ではラウンド定数を XOR 加算する．ラウンド定数 $AC_0, \ldots AC_{15}$ を以下に示す．

$$\begin{bmatrix} 0\oplus(ks_7 \parallel ks_6 \parallel ks_5 \parallel ks_4) & (rc_5 \parallel rc_4 \parallel rc_3) & 0 & 0 \\ 1\oplus(ks_7 \parallel ks_6 \parallel ks_5 \parallel ks_4) & (rc_2 \parallel rc_1 \parallel rc_0) & 0 & 0 \\ 2\oplus(ks_3 \parallel ks_2 \parallel ks_1 \parallel ks_0) & (rc_5 \parallel rc_4 \parallel rc_3) & 0 & 0 \\ 3\oplus(ks_3 \parallel ks_2 \parallel ks_1 \parallel ks_0) & (rc_2 \parallel rc_1 \parallel rc_0) & 0 & 0 \end{bmatrix}$$

Vol.2015-SLDM-170 No.47
Vol.2015-EMB-36 No.47
2015/3/7

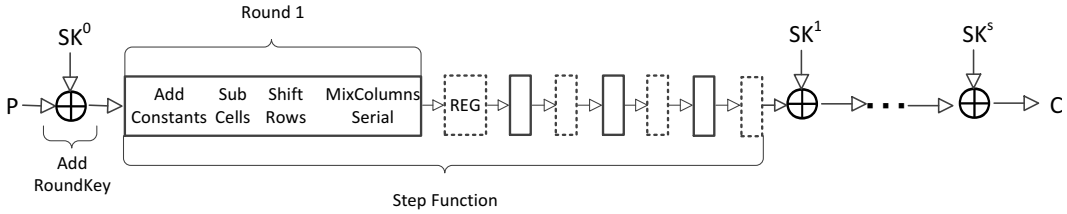

図 1 LED 暗号処理.

$(rc_5, rc_4, rc_3, rc_2, rc_1, rc_0)$ は最初に 0 に初期化される．ラウンド毎に左に 1 シフトし，rc_0 は $rc_5 \oplus rc_4 \oplus 1$ の値を設定することで更新する．$ks_7 ks_6 \ldots ks_0$ は秘密鍵長を 8 ビットで表した値である．

Sub Cells では Sbox で換字処理を行う．表 1 に Sbox の動作を 16 進数で示す．

Shift Rows では行列の $i(0 \leq i \leq 3)$ 行目を左に i シフトする．

Mix Columns Serial では行列 M と列ごとに乗算する．以下に行列 M を示す．

$$M = \begin{bmatrix} 4 & 1 & 2 & 2 \\ 8 & 6 & 5 & 6 \\ B & E & A & 9 \\ 2 & 2 & F & B \end{bmatrix}$$

ラウンド処理実行後のデータはレジスタに格納される．

3. LED 暗号へのスキャンベース攻撃手法

LED 暗号へのスキャンベース攻撃手法として [2,3] がある．しかし，この手法では鍵長が 64 ビットより大きい場合，解読できない．

本章では，鍵長に依存しない LED 暗号へのスキャンベース攻撃手法を提案する．攻撃対象の LSI のスキャンチェインは，Mix Columns Serial 後のデータを格納するレジスタを含むものとする．秘密鍵の値を保持するレジスタは含まないものとする．

3.1 前提条件

LED 暗号へのスキャンベース攻撃において，攻撃者が出来ることを以下に示す．

- 暗号化アルゴリズムを知っている
- 暗号回路で任意の平文を暗号化できる
- 任意のタイミングでスキャンチェインへアクセスできる

攻撃者は任意の平文を暗号 LSI に入力し，任意のタイミングでスキャンデータを取得できる．

攻撃者が分からないことを以下に示す．

- スキャンチェインのレジスタ接続順
- スキャンチェインに含まれるレジスタの数や種類

スキャンチェインは総配線長が短くなるように接続されるため，レジスタの接続順は攻撃者には分からない．そのため，暗号回路のレジスタの値がスキャンデータ上のどのビット位置にあるか攻撃者には不明である．また，スキャンチェインには通常，LSI チップ上の他の周辺回路のレジスタも含まれており，スキャンチェインで接続されているレジスタの種類や数は攻撃者には分からない．スキャンチェインに含まれるレジスタの接続順，種類が分からない場合でも秘密鍵を解読できる手法を提案する．

3.2 スキャンシグネチャを用いた解析

多数の平文を想定し，それぞれを攻撃対象 LSI に設定し，暗号化処理中にそれぞれ同じタイミングでスキャンデータを取得したとする．図 3 下のように用いた平文毎に取得したスキャンデータを縦に並べる．スキャンチェイン長を k ビット，入力した平文数を n 個とすると，横に k 個，縦に n 個のビットが並ぶことになる．これらを n ビットの列データが k 個横に並んでいるものとして見ると，それぞれの列データは攻撃対象 LSI 中のある 1 ビットレジスタの平文に対する値の変化を表していることに気付く．攻撃対象 LSI に入力する平文数 n が大きいとき，この n ビットの列データの値はその 1 ビットレジスタ固有の値になる．これをスキャンシグネチャという．

次に，攻撃対象 LSI に入力した平文 n 個について，それぞれ暗号化シミュレータで暗号化処理を行い，スキャンデータを取得したタイミングと同じ時点のレジスタ値を求める．但し，秘密鍵のとりうる全ての値について想定し，それぞれシミュレータでレジスタ値を計算するものとする．これらシミュレータで求めた値を平文毎に縦に並べる．そして，秘密鍵を K_1 と想定し，n 個の平文を暗号シミュレータで暗号化したときに得られたレジスタ r のスキャンシグネチャが LSI から取得したスキャンデータ中に存在するかを探索する (図 3)．秘密鍵を K_1 と想定したときのレジスタ r のスキャンシグネチャがスキャンデータに存在すれば，秘密鍵の値は K_1 と考えることができる．レジスタ r のスキャンシグネチャがスキャンデータに存在しなければ，予想した秘密鍵の値は誤っていることが分かる．

ところがここで大きな問題が起こる．秘密鍵長が 64 ビットの LED 暗号では副鍵 SK^0 を，秘密鍵長が 64 ビットより大きい場合には副鍵 SK^0，SK^1 を解読できれば直ちに秘密鍵 K を解読可能であるが，副鍵 SK^0，SK^1 はそれぞれ 64 ビットであるため，秘密鍵長を kl とするとその候補

表 1 Sbox.

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S [x]	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

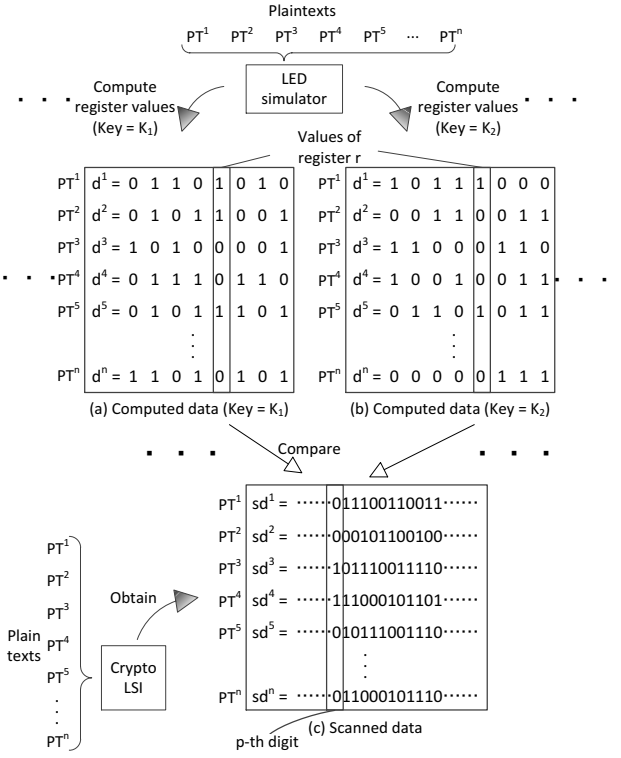

図 3 スキャンシグネチャと秘密鍵の予想.

は $2^{kl}(64 \leq kl \leq 128)$ 個となり，これらを総当たりで試行し，スキャンデータと比較することは事実上不可能である．副鍵 SK^0，SK^1 のデータ全部でなく，その一部のみが影響を与えるように，データを加工する必要がある．

3.3 SK^0 の解読

本節では，副鍵 SK^0，SK^1 の 128 ビットのデータ全部でなく，その一部のみが影響を与えるように，データを加工することを考える．

ラウンド処理では，Add Constants, Sub Cells, Shift Rows, Mix Columns Serial を順に実行し，レジスタに値が格納される．図 2 において，a_0 の値は e_0, e_4, e_8, e_{12} の値に影響を及ぼし，他の e_i の値は a_0 の値とは独立である．同様に，$a_i(1 \leq i \leq 15)$ の値に Mix Columns Serial 後の 4 つの要素の値がそれぞれ依存しており，残りの 12 の要素は a_i の値とは独立になっている．

また，図 2 において，e_0 の値は a_0, a_5, a_{10}, a_{15} に依存している．同様に，$e_i(1 \leq i \leq 15)$ の値は Add Constants 実行前の 4 つの要素の値にそれぞれ依存している．

ここで，先頭の 4 ビット a_0 と a'_0 のみが異なり，他の要素は同じ 64 ビットの 2 つの数値 a, a' を用意する．これらに対し，それぞれ Add Constants, Sub Cells, Shift Rows, Mix Columns Serial を順に実行して e, e' を計算する．求めた e, e' の排他的論理和をとって $e \oplus e'$ を求める．e と e' は 0 列目のみが異なり，他の要素は同じ値になるため，$e \oplus e'$ は 0 列目以外は 0 になる．$e \oplus e'$ の 0 列目は以下のようになる．

$$\begin{aligned}\begin{bmatrix} e_0 \oplus e'_0 \\ e_4 \oplus e'_4 \\ e_8 \oplus e'_8 \\ e_{12} \oplus e'_{12} \end{bmatrix} &= \begin{bmatrix} 4 & 1 & 2 & 2 \\ 8 & 6 & 5 & 6 \\ B & E & A & 9 \\ 2 & 2 & F & B \end{bmatrix} \begin{bmatrix} d_0 \oplus d'_0 \\ d_4 \oplus d'_4 \\ d_8 \oplus d'_8 \\ d_{12} \oplus d'_{12} \end{bmatrix} \\ &= \begin{bmatrix} 4 & 1 & 2 & 2 \\ 8 & 6 & 5 & 6 \\ B & E & A & 9 \\ 2 & 2 & F & B \end{bmatrix} \begin{bmatrix} d_0 \oplus d'_0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \end{aligned} \quad (1)$$

式 1 より，$e \oplus e'$ の 0 列目は d_0, d'_0 を用いて表せることが分かる．d_0, d'_0 は a_0, a'_0 に依存する．a_0, a'_0 は 1 つ前のラウンド処理で実行された Mix Columns Serial の出力値の 0 番目の要素，または直前に実行された Add RoundKey の出力値の 0 番目の要素である．

さてここで，1 ラウンド目を考えよう．つまりこれら 64 ビットの数値 a, b, c, d, e と a', b', c', d', e' が 1 ラウンド目の値であるとする．64 ビットの 2 つの平文をそれぞれ m，m' とすると a_0, a'_0 は，

$$a_0 = m_0 \oplus sk_0^0$$
$$a'_0 = m'_0 \oplus sk_0^0$$

である．ここで sk_0^0 は 64 ビットの副鍵 SK^0 の上位 4 ビット，つまり第一要素のみを表している．よって $e \oplus e'$ の 0 列目は平文 m_0, m'_0 と副鍵 SK^0 の第一要素 sk_0^0 のみに依存する値になっていることが分かる．つまり，この値は副鍵 SK^0 全体 64 ビットに依存する値でなくその第一要素 sk_0^0 の 4 ビットにのみ依存する値であり，この値は sk_0^0 の全数探索によって，十分現実的にそのすべての値を試行することができる．

そして，0 番目の要素のみ値が異なる 2 つの平文 m, m' を攻撃対象の LSI に設定し，1 ラウンド目の Mix Columns Serial 後の値をスキャンデータとしてそれぞれ取得する．取得したスキャンデータを排他的論理和したデータ中で $e \oplus e'$ の 0 列目に当たる部分は，平文 m_0, m'_0 と副鍵 sk_0^0 のみに依存する．つまり，排他的論理和したデータ中で $e_0 \oplus e'_0$, $e_4 \oplus e'_4$, $e_8 \oplus e'_8$, $e_{12} \oplus e'_{12}$ の 16 個のビットは副鍵 sk_0^0 に依存しており，他の副鍵の要素 $sk_i^0(i \neq 0)$ とは独立である．

全て 0 の平文 $m^{(0)}$ と 0 番目の要素のみ値が異なる多数の平文 $m^{(1)}, \ldots, m^{(n)}$ をそれぞれ攻撃対象の LED 暗号 LSI

Vol.2015-SLDM-170 No.47
Vol.2015-EMB-36 No.47
2015/3/7

に入力し，1 ラウンド目の Mix Columns Serial 後の値をスキャンデータとしてそれぞれ取得し，$sd^{(0)}, \ldots, sd^{(n)}$ とする．次に

$$\begin{aligned} &sd^{(0)} \oplus sd^{(1)} \\ &sd^{(0)} \oplus sd^{(2)} \\ &\quad\vdots \\ &sd^{(0)} \oplus sd^{(n)}. \end{aligned}$$

を求める．これらのデータ中で

$$\begin{array}{cccc} e_0^{(0)} \oplus e_0^{(1)}, & e_4^{(0)} \oplus e_4^{(1)}, & e_8^{(0)} \oplus e_8^{(1)}, & e_{12}^{(0)} \oplus e_{12}^{(1)} \\ e_0^{(0)} \oplus e_0^{(2)}, & e_4^{(0)} \oplus e_4^{(2)}, & e_8^{(0)} \oplus e_8^{(2)}, & e_{12}^{(0)} \oplus e_{12}^{(2)} \\ \vdots & \vdots & \vdots & \vdots \\ e_0^{(0)} \oplus e_0^{(n)}, & e_4^{(0)} \oplus e_4^{(n)}, & e_8^{(0)} \oplus e_8^{(n)}, & e_{12}^{(0)} \oplus e_{12}^{(n)} \end{array}$$

は副鍵 sk_0^0 に依存している．これら 16 個の列データをスキャンシグネチャ SS_0 とする．

スキャンシグネチャ SS_0 は副鍵の要素の中で sk_0^0 のみに依存している．3.2 節より，副鍵 sk_0^0 の全パターンについてシミュレータでスキャンシグネチャ SS_0 を求め，スキャンデータを排他的論理和したデータ中にこのスキャンシグネチャ SS_0 があるかを探索する．スキャンシグネチャ SS_0 が存在すれば予想した sk_0^0 の値が正しいと分かる．同様に，$sk_1^0, \ldots sk_{15}^0$ を解読できれば，副鍵 SK^0 全体を解読できる．

sk_0^0 を解読する手法をまとめる．

(1) **攻撃に使用する平文を生成し，スキャンデータを取得**
全て 0 の平文 $m^{(0)}$ と，0 番目の要素をランダムに生成し，0 番目の要素 (m_0) 以外はすべて 0 である n 個の平文を生成する．この平文群を $m^{(0)}, \ldots, m^{(n)}$ とする．平文群を攻撃対象の LED 暗号 LSI に入力し，1 回目のラウンド処理を終えた時点のスキャンデータ $sd^{(0)}, \ldots, sd^{(n)}$ を取得する．

(2) **スキャンデータの XOR 演算**
$sd^{(0)} \oplus sd^{(1)}, sd^{(0)} \oplus sd^{(2)}, \ldots, sd^{(0)} \oplus sd^{(n)}$ を計算し，解析データとする．

(3) **副鍵を予想してシミュレーション**
副鍵 SK^0 の第一要素 sk_0^0 について $2^4 = 16$ 通りの値をそれぞれ想定し，平文群 $m^{(0)}, \ldots, m^{(n)}$ を入力としたときの $e^{(0)}, \ldots, e^{(n)}$ を暗号シミュレータから求める．

(4) **シミュレーション値の XOR 演算**
$e^{(0)} \oplus e^{(1)}, e^{(0)} \oplus e^{(2)}, \ldots, e^{(0)} \oplus e^{(n)}$ を計算し，比較データとする．

(5) **スキャンシグネチャを利用した比較**
解析データの列データと比較データを比較し，比較データ中の 16 個の列データ，つまりスキャンシグネチャ SS_0 が解析データの列データ中に存在するとき，予想した sk_0^0 が正しい値であることが分かる (図 4)．

同様に $sk_1^0, \ldots, sk_{15}^0$ を求めれば，副鍵 SK^0 全体が解読できる．

秘密鍵長が 64 ビットの場合，上記の手法で副鍵 SK^0 を解読することで直ちに秘密鍵 K を解読可能である．秘密鍵長が 64 ビットより大きい場合には，上記の手法で副鍵 SK^0 を解読することで，秘密鍵の上位 64 ビットの値が判明する．秘密鍵の残りの部分は，SK^1 を求めることで判明する．

3.4 SK^1 の解読

式 1 より，$e \oplus e'$ の 0 列目は a_0, a_0' に依存することを示した．ここでこれらが 5 ラウンド目の値であるとすると，$e \oplus e'$ の 0 列目は 4 ラウンド目の出力値の 0 番目の要素，副鍵 SK^1 の第一要素 sk_0^1 のみに依存する値になっている．つまり，この値は副鍵 SK^1 全体 64 ビットに依存する値でなくその第一要素 sk_0^1 の 4 ビットにのみ依存する値であり，この値は sk_0^1 の全数探索によって，十分現実的にそのすべての値を試行することができる．

ここで，4 ラウンド目の出力値は 0 番目の要素のみ異なる値に設定する必要がある．3.3 節で求めた SK^0 を用いて，そのような値を与える平文を計算すればよい．

sk_0^1 を解読する手法をまとめる．

(1) **攻撃に使用する平文を生成し，スキャンデータを取得**
4 ラウンド目の出力値をそれぞれ，全て 0 にする平文，0 番目の要素のみ異なり，0 番目の要素以外全て 0 にする平文，n 個を 3.3 節で求めた SK^0 を用いて逆演算し求める．この平文群を $m^{(0)}, \ldots, m^{(n)}$ とする．平文群を攻撃対象の LED 暗号 LSI に入力し，5 回目のラウンド処理を終えた時点のスキャンデータ $sd^{(0)}, \ldots, sd^{(n)}$ を取得する．

(2) **スキャンデータの XOR 演算**
$sd^{(0)} \oplus sd^{(1)}, sd^{(0)} \oplus sd^{(2)}, \ldots, sd^{(0)} \oplus sd^{(n)}$ を計算し，解析データとする．

(3) **副鍵を予想してシミュレーション**
副鍵 SK^1 の第一要素 sk_0^1 について $2^4 = 16$ 通りの値をそれぞれ想定し，平文群 $m^{(0)}, \ldots, m^{(n)}$ を入力としたときの $e^{(0)}, \ldots, e^{(n)}$ を暗号シミュレータから求める．

(4) **シミュレーション値の XOR 演算**
$e^{(0)} \oplus e^{(1)}, e^{(0)} \oplus e^{(2)}, \ldots, e^{(0)} \oplus e^{(n)}$ を計算し，比較データとする．

(5) **スキャンシグネチャを利用した比較**
解析データの列データと比較データを比較し，比較データ中の 16 個の列データ，つまりスキャンシグネチャ SS_0 が解析データの列データ中に存在するとき，予想した sk_0^1 が正しい値であることが分かる．

同様に $sk_1^1, \ldots, sk_{15}^1$ を求めれば，副鍵 SK^1 全体が解読できる．

Vol.2015-SLDM-170 No.47
Vol.2015-EMB-36 No.47
2015/3/7

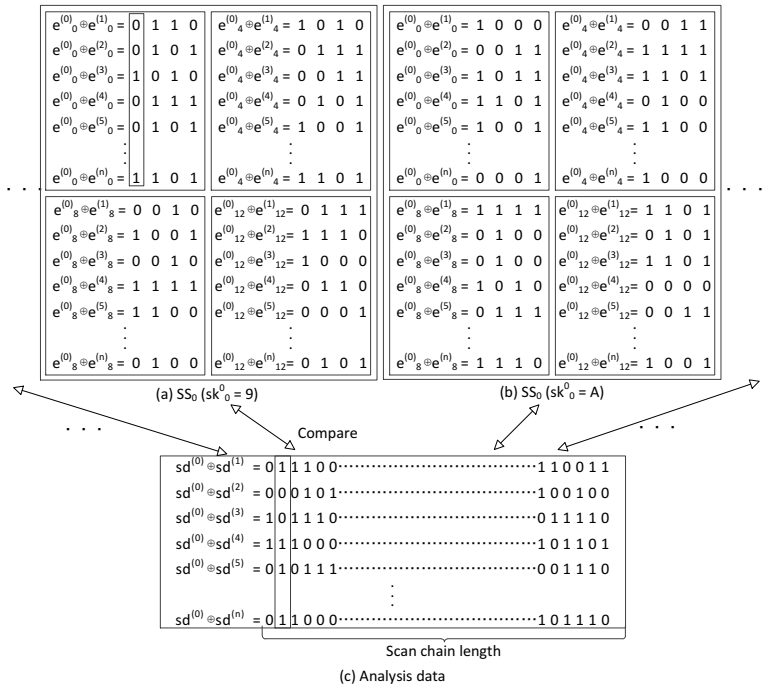

図 4　sk_0^0 の解読.

表 2　秘密鍵解読結果.

	平均平文数	最悪平文数	平均解析時間 [s]	最悪解析時間 [s]
SK^0 の解読	71.98	79	0.234	0.268
SK^1 の解読	72.15	78	0.234	0.268

4. 評価実験

本節では，提案手法のソフトウェアシミュレーション結果を説明する．実験では，提案手法を C 言語で実装し，暗号回路のシミュレータは [1] のコードを使用した．暗号回路のレジスタ 64 個のみがスキャンチェインに接続されていることとし，1 回目と 5 回目のラウンド処理後のレジスタ値をシミュレータにより取得し，スキャンデータと想定する．スキャンデータと想定したデータに対し，提案手法のシミュレータを実行して秘密鍵を解読する．本実験は，CPU が Intel(R) Core(TM) i7-2620M 2.70GHz × 4，メモリが 8GB の計算機を用い，コンパイラは gcc を使用した．

ランダムに生成した 128 ビットの秘密鍵 100 個の復元に必要な平均平文数，最悪平文数，平均解析時間，最悪解析時間を計測した．表 2 に解読結果を示す．提案手法で副鍵 SK^0，SK^1 を順に解読できることを確認した．

5. おわりに

本稿では，鍵長に依存しない LED 暗号に対するスキャンベース攻撃手法を提案した．提案手法は，特定の平文を入力した LSI から取得したスキャンデータの排他的論理和をとり，特定のビット列に着目することで秘密鍵を部分ごとに解読する手法である．提案手法は，暗号回路のみをスキャンチェインに含む場合，平均 145 個の平文で 128 ビットの秘密鍵を 0.468 秒で解読可能と確認した．

今後の研究課題として以下が挙げられる．現在，暗号化処理のタイミングが分かっていることを前提としているが，暗号化処理タイミングが不明の時でも提案手法は有効であると考えている．ラウンド 1 処理後，ラウンド 5 処理後のタイミングが分からない時に提案手法で秘密鍵の解析ができるか今後調査予定である．

謝辞

本研究は総務省・戦略的情報通信研究開発推進事業 (SCOPE) による．

参考文献

[1] J. Guo, T. Peyrin, A. Poschmann, and M. Robshaw, “The LED block cipher,” *Lecture Notes in Computer Science*, vol. 6917, pp. 326–341, 2011.

[2] M. Fujishiro, M. Yanagisawa, and N. Togawa, “Scan-based side-channel attack on the LED block cipher using scan signatures,” *IEICE Trans. on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E97-A, no. 12, pp. 2434–2442, 2014.

[3] M. Fujishiro, M. Yanagisawa, and N. Togawa, “Scan-based attack on the LED block cipher using scan signatures,” in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS 2014)*, pp. 1460–1463, 2014.

[4] A. Biryukov and D. Khovratovich, “Related-key cryptanalysis of the full AES-192 and AES-256,” *Lecture Notes in Computer Science*, vol. 5912, pp. 1–18, 2009.

[5] M. Ågren, “Some instant- and practical-time related-key attacks on KTANTAN32/48/64,” in *Proc. of the 18th international conference on Selected Areas in Cryptography*, pp. 213–229, 2011.