2008－SLDM－137（9）
2008/11/17

周辺回路を含むAES-LSIへのスキャンベース攻撃

奈良 竜太† 戸川 望† 柳澤 政生† 大附 辰夫†

† 早稲田大学大学院基幹理工学研究科
〒169–8555 東京都新宿区大久保 3–4–1
E-mail: †nara@togawa.cs.waseda.ac.jp

あらまし 暗号LSIに対するサイドチャネル攻撃の危険性が指摘されているなか，スキャンチェインを利用して秘密鍵を解読するスキャンベース攻撃が注目されている．スキャンチェインは必須のLSIテスト技術である一方，LSI内部のレジスタを直接観測できるため，暗号回路の秘密鍵解読に利用されている．従来のスキャンベース攻撃は暗号回路だけのレジスタだけで構成されたスキャンチェインにのみ有効であり，周辺回路のレジスタを考慮していない欠点があった．そこで本稿では暗号回路以外のレジスタがスキャンチェインに含まれていても秘密鍵を解読する手法を提案する．特定のレジスタに着目し，その値の変化を見ることで秘密鍵を解析する．他のレジスタに影響を受けないため，スキャンチェインの構成に依存しない．そのため，周辺回路を含んだ，より現実に近い暗号LSIに対しスキャンベース攻撃することができる．

キーワード AES, 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

Scan-based Attack for an AES-LSI included with other IPs

Ryuta NARA†, Nozomu TOGAWA†, Masao YANAGISAWA†, and Tatsuo OHTSUKI†

† Grad. of Fundamental Science and Engineering, Waseda University
3–4–1 Okubo, Shinjuku, Tokyo 169–8555, Japan
Tel: +81–3–3209–3211(5775), Fax: +81–3–3208–7439
E-mail: †nara@togawa.cs.waseda.ac.jp

Abstract The threat of side-channel attacks against the cryptography LSI is indicated. Especially, scan-based attacks, which use the scan chain, are watched. Scan chains are one of the most important testing techniques, but it is possible to use for attacks against the cryptography LSI. Conventional scan-based attacks only consider the scan chain made by registers of cryptography circuits. However, cryptography LSI usually has many IPs such as memories, micro-processors and other circuits. Because of the real scan chain consists of many kinds of registers, it is obscure whether conventional scan-based attacks can attack or cannot. In this paper, scan-based attack which enables to crack the secret key in the AES-LSI with other IPs is proposed. By focusing the bit pattern of the specific register and monitoring its change, and our method eliminates the influence of other circuit registers. Therefore, our scan-based attacks don't depend on the architecture of the scan chain, and it can crack real cryptography LSIs included with other IPs.

Key words Advanced Encryption Standard, AES, Side-channel attacks, Scan chain, Scan-based attack

1. まえがき

近年，暗号LSIに対するサイドチャネル攻撃の危険性が指摘されている．なかでも，テスト用のスキャンチェインを利用し暗号LSIの秘密鍵を解読するスキャンベース攻撃が注目されている．スキャンチェインとはLSI中のレジスタを直列に接続し，LSIの外部からレジスタを直接制御・観測できるようにする手法である．LSIをテストする際にレジスタを自由に制御できるので，テスト効率を大幅に高めることができる．一方，LSI中のレジスタデータを容易に取得できる性質を利用し，暗号回路の秘密鍵解読に応用したのがスキャンベース攻撃である．

スキャンベース攻撃のポイントは，いかにしてスキャンデータから秘密鍵を求めるかにある．スキャンテストには，テスト効率化のために回路中のあらゆるレジスタを接続する必要があ

るので，配線が複雑になるという問題がある．そのため一般にレイアウト時に距離が近いレジスタ同士を接続し，配線長が短くなるようにスキャンチェインが最適化される．レジスタの接続順番は配線の都合で決まるため，データ通りの順番に接続されることはなく，スキャンデータだけではスキャンチェインとレジスタとの対応関係は分からない．つまり，レジスタがでたらめに接続されたスキャンデータから秘密鍵を解読する必要がある．

スキャンベース攻撃の研究は，2004 年に Yang 氏らが DES に対し，スキャンチェインを利用して秘密鍵を解読できることを初めて示した [1]．DES-LSI 中のレジスタにより構成されたスキャンチェインから暗号処理中の中間値を取得し，そのレジスタ構成を解析することで秘密鍵の解読に成功している．2005 年には AES に対するスキャンベース攻撃 [2] が提案されており，DES や AES のような広く使用されている暗号に対し，スキャンベース攻撃による秘密鍵解読が有効であることを示している．

しかしながら，Yang 氏らの手法には，解析できるスキャンチェインが限定されているという欠点がある．文献 [1] はスキャンチェイン中に DES 回路のレジスタのみが含まれているモデル上で鍵を解析している．スキャンチェイン中に DES 以外のレジスタが含まれていた場合，秘密鍵が解読できるかどうかは不明である．文献 [2] も AES のデータレジスタのみが含まれているモデルを想定しており，さらに，AES を制御するコントローラのレジスタすらスキャンチェインに含まれていない．そのため，周辺回路のレジスタがスキャンチェインに含まれていた場合，解読可能かどうかは同じく不明である．実際には，暗号回路単体で 1 つの LSI チップになることはほとんどなく，同一チップ上にメモリやプロセッサなど，暗号回路以外の様々な回路が載っているのが一般的である．さらに，各回路が同じスキャンチェインの I/O や制御信号を共有することで，面積や外部 I/O ポートを節約することは珍しくないため，現実の暗号 LSI を Yang 氏らの手法を使ってスキャンベース攻撃することは困難である．

そこで本稿では，AES に対する，スキャンチェインの構造に依存しないスキャンベース攻撃アルゴリズムを提案する．提案手法は，秘密鍵に依存したデータがスキャンデータ中のあるレジスタ値として存在するか否かで秘密鍵を解読する．Yang 氏らの手法と異なり，1 つのレジスタの値のみに着目するため，そのほかのレジスタに影響されることがない．つまり，暗号回路の実装方法に依存せず，周辺回路のレジスタがスキャンチェインに含まれている場合でも確実にスキャンベース攻撃を行うことができる．本手法を用いることにより，現実に近い暗号 LSI の秘密鍵でも解読することができる．

2. 従来研究

本節では，Yang 氏らの AES に対するスキャンベース攻撃手法 [2] について説明する．AES に対するサイドチャネル攻撃は，AES のラウンド鍵から秘密鍵を解読する手法が一般的である．ラウンド鍵 RKn は秘密鍵 k から単純な鍵拡張処理により計算

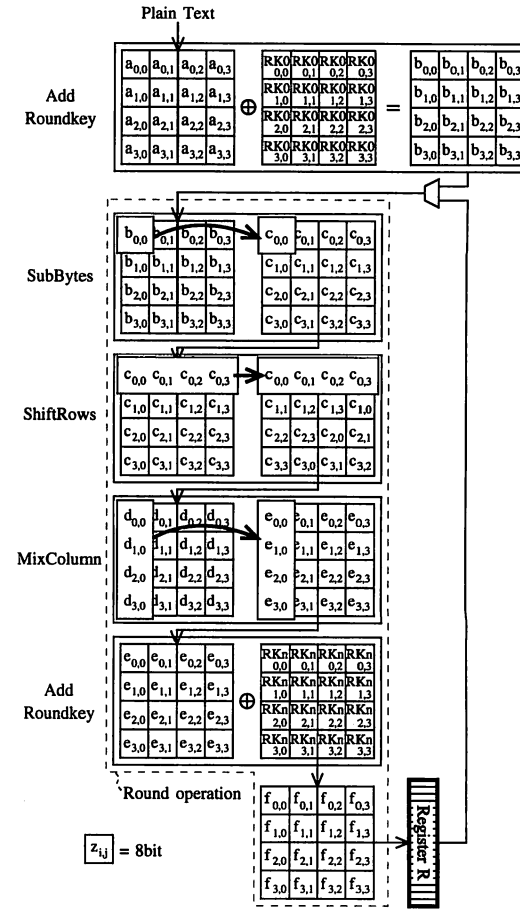

図 1　AES ラウンド処理のブロック図.

するため，一つでもラウンド鍵 RKn が分かれば秘密鍵 k を逆計算することができるからである．従って，スキャンベース攻撃の目的はラウンド鍵 RKn の内一つを解読することを目的としている．Yang 氏らは，特定の入力ペアから計算したラウンド値同士を XOR 加算してラウンド鍵 $RK1$ の影響をなくした上で，その値とラウンド鍵 $RK0$ との相関値を解析することで，スキャンデータからラウンド鍵を解読している．Yang 氏らが使用した AES ハードウェアと具体的なスキャンベース攻撃について示す．

2.1 AES ハードウェア

AES ハードウェアは文献 [3] のような標準的なアーキテクチャを使用する．図 1 に AES ラウンド処理のブロック図を示す．鍵長 128 ビット，データブロック 128 ビットの AES を処理する．AES で扱うデータ長は 128bit であるが，AES の処理単位は 1 バイトなので，1 バイトを 1 要素とした行列としてデータを表現する．行列の要素を $z_{i,j}$ のように表す．128bit の平文を a とする．次に平文 a をラウンド鍵 $RK0$ と XOR 加算し，その結果を b とする．この処理を前ラウンド処理とする．前ラウンド処理の結果を 1 回目のラウンド処理の入力とする．ラウンド処理は SubBytes → ShiftRows → MixColumn →ラウ

ンド鍵 RKn と XOR 加算という一連の処理から構成される．SubBytes は 1 バイト入力 1 バイト出力の関数，ShiftRows は行方向に左シフト，MixColumn は列方向に 4 バイト入力 4 バイト出力の関数である．MixColumn の出力とラウンド鍵 RKn を XOR 加算し，その出力をレジスタ R に保存する．このレジスタ出力を再びラウンド処理の入力とすることでラウンド処理を必要な回数分ループさせる．鍵長 128 ビット，データブロック 128 ビットの場合，ループ回数は 10 回である．**スキャンチェインはレジスタ R のみから構成されている．**また，スキャンチェインのレジスタ接続順はでたらめで，接続順の情報は攻撃者は知らないものとする．

2.2 ラウンド鍵 $RK0$ 解読

Yang 氏らの手法 [2] は前ラウンド処理のラウンド鍵 $RK0$ を解読している．図 1 のような標準的な AES アーキテクチャの場合，1 回目のループ時に前ラウンド処理とラウンド処理を 1 サイクルで処理するため，レジスタ R にはラウンド鍵 $RK0$ と $RK1$ を XOR 加算したデータが保存される．$RK0$ 解読を容易にするため，スキャンデータからラウンド出力 f から RK1 の影響を排除したデータを求める．このとき，同じ値を XOR 加算すると 0 になる性質を利用して $RK1$ を除いたデータを取得できる（式 1）．

$$\begin{aligned} f^1 \oplus f^2 &= (e^1 \oplus RK1) \oplus (e^2 \oplus RK1) \\ &= e^1 \oplus RK1 \oplus e^2 \oplus RK1 \\ &= e^1 \oplus e^2 \oplus RK1 \oplus RK1 \\ &= e^1 \oplus e^2 \end{aligned} \tag{1}$$

次に，$RK0$ を $f^1 \oplus f^2$ から計算する方法について考える．$RK0$ は 128 ビットなので，全数探査すると 2^{128} パターンを試す必要があり非常に効率が悪い．そのため，$RK0$ を要素ごとに解読する必要がある．$RK0_{0,0}$ に着目すると，$RK0_{0,0}$ は平文の要素 $a_{0,0}$ に XOR 加算されて $b_{0,0}$ になる．この $b_{0,0}$ が，ラウンド処理から得られる $(f_{0,0}, f_{1,0}, f_{2,0}, f_{3,0})$ の入力の 1 つになる．次に，$b^1_{0,0}$ と $b^2_{0,0}$ のみが異なり，そのほかの要素は同じ数値 b^1, b^2 を用意する．この数値から f^1 と f^2 を計算し，$f^1 \oplus f^2$ を求める．このとき，f^1 と f^2 は 0 列目のみが異なり，そのほかの要素は同じ値となるので，$f^1 \oplus f^2$ は 0 列目以外は 0 となる．0 列目について考えるために，MixColumn の結果同士を XOR 加算したとき $(e^1 \oplus e^2)$ の 0 列目の内容を式 2 に示す．

$$\begin{bmatrix} e^1_{0,0} \oplus e^2_{0,0} \\ e^1_{1,0} \oplus e^2_{1,0} \\ e^1_{2,0} \oplus e^2_{2,0} \\ e^1_{3,0} \oplus e^2_{3,0} \end{bmatrix} = \begin{bmatrix} 03 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} d^1_{0,0} \oplus d^2_{0,0} \\ d^1_{1,0} \oplus d^2_{1,0} \\ d^1_{2,0} \oplus d^2_{2,0} \\ d^1_{3,0} \oplus d^2_{3,0} \end{bmatrix}$$

$d^1_{0,0} \neq d^2_{0,0}, d^1_{1,0} = d^2_{1,0}, d^1_{2,0} = d^2_{2,0}, d^1_{3,0} = d^2_{3,0}$なので

$$= \begin{bmatrix} 03 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} d^1_{0,0} \oplus d^2_{0,0} \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{2}$$

$f^1 \oplus f^2$ の 0 列目は $b_{0,0}$ と $RK0_{0,0}$ だけで決まることが分かる．

表 1　$f^1 \oplus f^2$ のハミング重みと $(b^1_{0,0}, b^2_{0,0})$ の関係

ハミング重み	$(b^1_{0,0}, b^2_{0,0})$
9	226, 227
12	242, 243
23	122, 123
24	130, 131

$RK1$ と同時に $RK0_{0,0}$ 以外の要素もすべて排除することができる．従って，$f^1 \oplus f^2$ の 0 列目から $b^1_{0,0}, b^2_{0,0}$ を求めることができれば，式 3 より $RK0_{0,0}$ を求めることができる．

$$\begin{aligned} a^1_{0,0} \oplus b^1_{0,0} &= a^1_{0,0} \oplus (a^1_{0,0} \oplus RK0_{0,0}) \\ &= (a^1_{0,0} \oplus a^1_{0,0}) \oplus RK0_{0,0} \\ &= RK0_{0,0} \end{aligned} \tag{3}$$

同様の手法を $RK_{0,0}$ 以外の要素について繰り返すことで，$RK0$ 全体を解析することができる．

2.3 攻撃 1:スキャンチェイン構造の解析

128bit のラウンド出力 f を保持するレジスタ R から，同じ列を保持する 32bit 分のレジスタを特定する．f の 0 列目は $(a_{0,0}, a_{1,1}, a_{2,2}, a_{3,3})$ を入力として計算する．$(a_{0,0}, a_{1,1}, a_{2,2}, a_{3,3})$ だけを変えたデータを複数入力し，スキャンデータを比較すれば，f の 0 列目を保持するレジスタ値だけが変化する．すなわち，$(a_{0,0}, a_{1,1}, a_{2,2}, a_{3,3})$ だけを変えた平文 a を何度か入力してスキャンデータの変化を比較すれば，$(f_{0,0}, f_{1,0}, f_{2,0}, f_{3,0})$ が保持されているレジスタが分かる．32 ビット特定するのに必要なパターン数は $(a_{0,0}, a_{1,1}, a_{2,2}, a_{3,3})$ に乱数を入力した場合，最悪時で 15 パターン，平均で 6 パターンである．このとき，レジスタの特定は 1 ビットごとの対応まで詳細に分析していないことに注意する．$(f_{0,0}, f_{1,0}, f_{2,0}, f_{3,0})$ 全体として保持されている場所を特定できればよい．

2.4 攻撃 2:RK0 解析

$(b^1_{0,0}, b^2_{0,0}) = (2m, 2m+1)$ or $(2m+1, 2m), 0 \leq m \leq 127$ のとき，$f^1 \oplus f^2$ のハミング重みは 7 から 25 の間になるが AES シミュレーションにより判明している．さらに，表 1 に示すように $f^1 \oplus f^2$ のハミング重みが 9,12,23,24 の場合，$(b^1_{0,0}, b^2_{0,0})$ は一つに定まることが分かってる．$b^1_{0,0}, b^2_{0,0}$ は LSB のみ異なるペアであるが，その性質は RK0 を XOR 加算しても変わらない．つまり，$(a^1_{0,0}, a^2_{0,0}) = (2t, 2t+1), 0 \leq t \leq 127$ とすれば，$(a^1_{0,0} \oplus RK0_{0,0}, a^2_{0,0} \oplus RK0_{0,0}) = (2m, 2m+1)$ or $(2m+1, 2m)$ が成り立つ．この性質を利用して，$RK0_{0,0}$ を解析する．

（1） 通常モードの AES-LSI に $a^1_{0,0} \in 2t, (0 \leq t \leq 127)$ を入力して，1 サイクル動かし，スキャンデータ sd_1 を取得

（2） 通常モードの AES-LSI に $a^2_{0,0} \in 2t+1, (0 \leq t \leq 127)$ を入力して，1 サイクル動かし，スキャンデータ sd_2 を取得

（3） $sd_1 \oplus sd_2$ を求め，特定した $(f_{0,0}, f_{1,0}, f_{2,0}, f_{3,0})$ のハミング重みが 9,12,23,24 なら $(b^1_{0,0}, b^2_{0,0})$ を決める．それ以外なら 1. に戻る．

（4） $b^1_{0,0}$ から $a^1_{0,0} \oplus b^1_{0,0} = RK0_{0,0}$ を計算し，$RK0_{0,0}$ を求める．

（5） 同様の手法で $RK0$ 全体を求める．

入力の組み合わせ数は $2^8 = 128$ 通りで，特定につながる数は 4 個なので，平均入力回数は 32(=128/4)，最悪入力回数は 124(128-4) である．32 ビットのスキャンチェイン構造解析に平均 6 個の平文が必要になる．スキャンサイズは 128 ビットなので 24 個の平文が必要である．$RK0$ を 1 バイト分求めるのに平均 32 個の平文が必要で，$RK0$ 全体を求めるのに 32*128/8=512 個の平文が必要である．従って $RK0$ を求めるのに平均 544 個の平文が必要になる．

3. 提案手法

Yang 氏らの手法 [2] は，スキャンチェインがラウンド処理の結果を保持するレジスタ R のみで構成されていることを前提としている．しかし，AES の制御回路や AES 以外の周辺回路のレジスタがスキャンチェインに含まれている場合，レジスタ値の変化を見てスキャンチェイン中の $(f_{0,0}, f_{1,0}, f_{2,0}, f_{3,0})$ を保持するレジスタを解析できるとは限らない．なぜなら，暗号処理中に同時に動作する回路が存在した場合，その回路も制御できるかは設計に依存するからである．そこで本稿では，スキャンチェインの構造にかかわらず $RK0$ を解析できるスキャンベース攻撃手法を提案する．提案手法は，入力に対する $f_1 \oplus f_2$ の変化と同等の変わり方をするレジスタ値をスキャンデータから発見することで $RK0$ を解読する．従来手法のように 32bit の $(f_{0,0}, f_{1,0}, f_{2,0}, f_{3,0})$ の数値を見るのではなく，1bit レジスタの変化を観測すれば良いので，スキャンデータ中にその値が存在するか否かで解析できる．従って，スキャンチェインのレジスタ位置を解析する必要がなくなるので，スキャンチェインの構造に依存しない解析が可能となる．

3.1 解析アルゴリズム

提案手法の $RK0$ 解析アルゴリズムについて説明する．提案手法も従来手法と同様に要素ごとに $RK0$ を解読する．まず，解析に必要なスキャンデータを取得する方法について説明する．8 ビットの入力 a_0 を n 個ランダムに選び，$a_0^i, i = 1, \cdots n$ とする．次に，a_0^i を $a_{0,0}$ の要素とし，その他の要素をすべて同じ値に固定した，n 個の平文 a^i を生成する．この平文 a^i を AES-LSI に入力し，1 サイクル後の，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^i を取得する．スキャンデータ sd^i から $RK1$ と $RK0_{0,0}$ 以外を取り除いたデータ $sd^1 \oplus sd^2, sd^1 \oplus sd^3, \cdots, sd^1 \oplus sd^n$ を計算する．このデータは n 個のスキャンデータから $n-1$ 個取得できる．

続いて，AES シミュレータから $RK0_{0,0}$ を解析するデータを取得する方法について説明する．AES-LSI に入力した n 個の平文 a^i を使って，MixColumn の出力 e^i を計算する．このとき $RK_{0,0}$ は 256 パターンあるので，$RK0_{0,0}$ 以外の要素を固定した $RK0$ が 256 パターン必要になる．AES-LSI から取得したスキャンデータに合わせるため，AES シミュレータから求めた MixColumn の出力から $e^1 \oplus e^2, e^1 \oplus e^3, \cdots, e^1 \oplus e^n$ を計算する．入力 a^i は a_0^i を $a_{0,0}$ の要素とし，その他の要素をすべて同じ値に固定しているので，$e^1 \oplus e^2, e^1 \oplus e^3, \cdots, e^1 \oplus e^n$ は 0 列目が異なり，それ以外の列の要素は 0 である．そして，

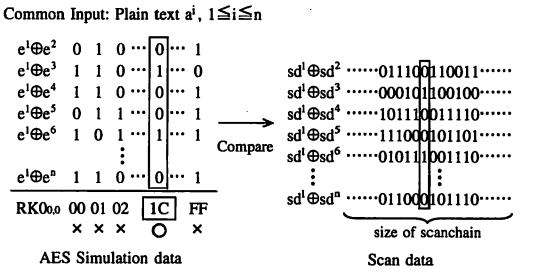

図 2 提案手法：$RK0_{0,0}$=1C 解読時の様子．

式 2 により 0 列目は平文 a^i と $RK0_{0,0}$ によって決まる．ここで，$e^1 \oplus e^2, e^1 \oplus e^3, \cdots, e^1 \oplus e^n$ の要素 (0,0) の LSB に着目する．入力 a^i はランダムに選択した n 個のデータなので，n 個の MixColumn の出力値は平文 a^i と $RK0_{0,0}$ によってのみ決まる．同様に，要素 (0,0) の LSB も平文 a^i と $RK0_{0,0}$ によってのみ決まる．このとき，平文 a^i は AES-LSI への入力と共通なので，つまり，$e^1 \oplus e^2, e^1 \oplus e^3, \cdots, e^1 \oplus e^n$ の要素 (0,0) の LSB は $RK0_{0,0}$ によってのみ決まることが分かる．データサイズは平文 a^i の数 $n$bit であるため，2^n 通りの組み合わせがあるので，n が十分大きいとき，$e^1 \oplus e^2, e^1 \oplus e^3, \cdots, e^1 \oplus e^n$ の要素 (0,0) の LSB と $RK0_{0,0}$ とは一意に決まる．

AES-LSI の $RK0_{0,0}$ は，AES シミュレータ上で $RK0_{0,0}$ ごとに求めた $e^1 \oplus e^2, e^1 \oplus e^3, \cdots, e^1 \oplus e^n$ の要素 (0,0) の LSB がスキャンデータから求めた $sd^1 \oplus sd^2, sd^1 \oplus sd^3, \cdots, sd^1 \oplus sd^n$ 中に存在するかどうかを調べることで解析する．存在すれば，対応する $RK0_{0,0}$ が解読できたことになり，存在しなければその $RK0_{0,0}$ は違うということになる．図 2 にその様子を示す．解析手順は以下のようになる．

（1） 通常モードの AES-LSI に $a_{0,0}$ が乱数で，それ以外が固定の n 個の平文 a^i を入力し，1 サイクル動かして，n 個のスキャンデータ sd^i を取得し，$sd^1 \oplus sd^2, sd^1 \oplus sd^3, \cdots, sd^1 \oplus sd^n$ を求める．

（2） 同じ平文 a^i と $0 \leqq RK0_{0,0} \leqq 255$ を AES シミュレータに入力し，MixColumn の出力 e^i を取得し，$e^1 \oplus e^2, e^1 \oplus e^3, \cdots, e^1 \oplus e^n$ を求める．

（3） $e^1 \oplus e^2, e^1 \oplus e^3, \cdots, e^1 \oplus e^n$ の要素 (0,0) の LSB が $sd^1 \oplus sd^2, sd^1 \oplus sd^3, \cdots, sd^1 \oplus sd^n$ 中に存在すれば $RK0_{0,0}$ の解読成功．存在しなければ (2) に戻る．

4. 評価・考察

提案手法の解読にかかるサイクル数を求める．AES-LSI で処理する平文数は $RK0$ の 1 要素あたり n 個必要である．要素は 16 個あるので，全体で $16n$ 個の平文が必要である．また同じ行の要素は MixColumn においてそれぞれ独立しているので，同じ行の要素 4 つのスキャンデータを同時に取得できる．この場合 $4n$ 個の平文で済む．ここで，現実的な n を求める．スキャンチェインのサイズを $size\_of\_scanchain$ ビットとすると，このとき $RK0_{0,0}$ が存在しないにも関わらず，同じ値が存在する確率は $size\_of\_scanchain/2^n$ となる．現実的なスキャ

ンチェインのサイズは大きくても 2000bit 程度である．また，スキャンチェイン中に AES 以外の回路のレジスタが含まれているような場合を考えると，これらのレジスタがランダムに変化する場合が最も誤認識されやすい状況である．入力数 n を 33 とし，比較データサイズを 32 としたとき，$RK0_{0,0}$ が存在しないにも関わらず，間違ってスキャンデータ中に検出される確率は $2000/2^{32} \cong 2^{-22}$ と非常に小さい．n は最大でも 33 あればほぼ確実に $RK0$ を解読できる．さらに，解読した $RK0$ が間違っているどうかは，解析した秘密鍵を使って AES 処理することで容易に確認できるため，$n = 16$ でも，誤解読の可能性は $2000/2^{16} \cong 2^{-6}$ 程度と小さいため，解析は十分可能である．従って，提案手法に必要な平文数は最悪でも $16 \times 32 = 512$，十分解析可能な平文数 $16 \times 16 = 256$ である．また，同じ行のスキャンデータをまとめて取得し，4 つの要素を同時に解析する場合は最悪でも $4 \times 32 = 128$，$4 \times 16 = 64$ 程度である．

提案手法が $RK0$ の解析に失敗する場合を考える．以下の 2 つの場合が考えられる．

- $RK0_{0,0}$ がスキャンデータ中に存在しないのに，同じ値がたまたま存在してしまう．
- 正しい $RK0_{0,0}$ にも関わらず，スキャンデータ中にその値が存在しない．

前者の失敗は，提案手法が $RK0$ の解析に失敗する可能性は，ある $RK0_{0,0}$ が存在しないにもかかわらず，対応する $e^1 \oplus e^2, e^1 \oplus e^3, \cdots, e^1 \oplus e^n$ の要素 (0,0) の LSB と同じ値が $sd^1 \oplus sd^2, sd^1 \oplus sd^3, \cdots, sd^1 \oplus sd^n$ 中に存在する場合である．平文 a^i の数 n がスキャンチェインのサイズに比べて小さい場合に起こる可能性がある．平文数 n を小さくすれば高速に処理でき，大きくすれば精度が上がるので，平文数 n は解析の速度と精度のトレードオフになるので，$n = 16$ 程度で十分解析可能な精度になる．

後者は，まずスキャンチェインに含まれていない AES レジスタが存在する場合が考えられる．しかし，提案手法は 1bit のレジスタで発見するのでラウンド鍵の結果を保持するレジスタ R がスキャンチェインに含まれていないときに解析不能となるため，スキャンベース攻撃の前提がなくなってしまう．あるいは，スキャンチェインにスキャンベース攻撃対策が組み込まれている場合が考えられる．この場合，その防御手法に対応した新たな解析手法を考える必要がある．ただし，レジスタを網羅し，なにも対策して通常のスキャンチェインの場合，正しい $RK0_{0,0}$ にも関わらず，発見できないことはない．

5. 結　　論

Yang 氏らによって提案されたスキャンベース攻撃は強力な攻撃手法である反面，スキャンチェイン構造が限定的で，想定していないレジスタが含まれた場合に対して攻撃できるかは不明である．一方，提案手法はスキャンチェインの構造に依存せずに攻撃することが可能になるため，複数の IP が搭載された SoC 上の AES 回路に対して秘密鍵を解析することができる．

暗号回路にスキャンチェインを実装する場合にはスキャンベース攻撃対策を講じる必要がある．最も確実なスキャンベース攻撃に対する防御手法は AES 回路にスキャンチェインを実装しないことである．暗号回路は回路規模が小さく，速度要求される用途でもないてめ，それほど厳しいテストは必要ないと考えられる．しかしながら，スキャンテストを使わないとテストに非常に時間がかかるため，スキャンチェインを使わなくてはならない状況が考えられる．暗号回路にスキャンチェインテストを実装する必要がある場合，スキャンチェインにスキャンベース攻撃の防御手法組み込む必要がある．以下のような防御手法が考えられる．

- ビット位置とレジスタとの対応が変化
- 各ビットがランダムに反転

そのほかにスキャンデータのアクセスを難しくするなどといった手法がある．

文　　献

[1] K. W. Bo Yang and R. Karri: "Scan based side channel attack on dedicated hardware implementations of data encryption standard", Test Conference, 2004. Proceedings. ITC 2004. International, pp. 339–344 (2004).

[2] K. W. Bo Yang and R. Karri: "Secure scan: a design-for-test architecture for crypto chips", Proceedings of the 42nd annual conference on Design automation, pp. 135–140 (2005).

[3] M. A. S. Mangard and S. Dominikus: "A highly regular and scalable aes hardware architecture", 1, pp. 483–491 (2004).