Vol.2011-SLDM-152 No.11
2011/10/25

スキャンチェイン構造に依存しない DESに対するスキャンベース攻撃手法

小寺博和[†1] 柳澤政生[†2] 戸川望[†1]

近年，暗号モジュールや暗号 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．その中でもスキャンチェインから取得できるスキャンデータによっ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．スキャンベース攻撃では，スキャンチェインから容易にスキャンデータを取得できる性質を利用して秘密鍵を解読する．本稿では，スキャンチェインのレジスタ接続順や暗号 LSI の動作するタイミングの特定が不要となる DES に対するスキャンベース攻撃手法を提案する．提案手法では，暗号 LSI に複数の平文を入力したときの暗号化処理中のスキャンデータの特定の 1 ビットに着目し，対応するレジスタの変化を観察することで秘密鍵を解読する．暗号 LSI 以外のレジスタがスキャンチェインに含まれた場合でも秘密の鍵解読が可能となるため，より現実的な条件でスキャンベース攻撃による秘密鍵解読が可能となる．

Scan-based Attack against DES Cryptosystems Independent of Scan-structure

Hirokazu Kodera,[†1] Masao Yangisawa[†2] and Nozomu Togawa[†1]

Side-channel attacks against crypto modules and LSIs become a practical threat these days. Especially, a scan-based attack which retrieves secret keys from scan data is considered to be one of the strongest side-channel attacks. In this paper, a scan-based attack method against DES cryptosystems is proposed. In our method, several plain texts are inputted into a DES module. After that, an attacker retrieves a secret key by observing a specific bit line of these scanned data. Because the values of a specific bit line dependent on the secret key, an attacker can analyze secret key using these values. Even when an attacker does not know scan chain structure implemented on a DES module and even when scan chain includes registers other than DES crypto modules, our proposed method can successfully retrieve its secret key. Several Experimental evaluations are demonstrated to confirm the effectiveness of our proposed method.

1. はじめに

IC カードを利用したサービスを安全に運用するために，データを暗号化するための暗号モジュールが搭載されている．暗号モジュールや暗号 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が，その中でもテスト用スキャンチェインから取得したスキャンデータ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．スキャンデータとは，スキャンチェインから取得できる内部レジスタが保持しているのデータである．スキャンチェインとは LSI のフリップフロップ（FF）をスキャンフリップフロップ（SFF）に置き換えて，それらを直列に接続することで外部からレジスタを制御・観測することを可能にする手法である．スキャンチェインを実装することにより，テスト効率が大幅に向上する一方で，スキャンチェインから容易にスキャンデータを取得できる性質がある．この性質を利用したサイドチャネル攻撃がスキャンベース攻撃である．

スキャンベース攻撃に関する研究は，2004 年に Yang らによる共通鍵暗号 DES に対する秘密鍵解読手法が報告されている [4]．文献 [4] の手法は，スキャンチェインを利用して特定のタイミングで外部からレジスタにデータを挿入することや，スキャンデータを取得することで秘密鍵解読を実行している．また，2006 年に共通鍵暗号 AES に対するスキャンベース攻撃手法の研究も Yang らは報告している [5].

一般に，スキャンチェインは物理的に近いレジスタ同士を接続するように設計されるため，設計者はスキャンチェインにおけるレジスタの接続順を知っているが，攻撃者はレジスタの接続順が分からない．したがって，攻撃者がスキャンチェインから取得するスキャンデータは論理的な意味を持たないデータとなる．文献 [4,5] では，秘密鍵解読を実行する前に，複数の平文を DES 暗号 LSI，AES 暗号 LSI に入力することでレジスタ位置とスキャンデータのビットの対応関係を特定する操作を行っている．攻撃対象のスキャンチェインが暗号 LSI の特定のレジスタのみで構成されていると仮定した場合は有効な操作であり，スキャンデータとレジスタの対応関係を特定することができる．しかし，一般に暗号 LSI のレジスタのみでスキャンチェインが構成されることはなく，スキャンチェインは周辺回路のレジスタも含むため，この操作ではスキャンチェインの構成を特定することができない．また，暗号 LSI が動作するタイミングは不明なため，攻撃者が特定のタイミングでレジスタデータを挿入することや取得することは難しい．したがって，文献 [4,5] の手法を適用すること

†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

は，現実的には難しいと考えられる．

文献 [4,5] だけでなく，共通鍵暗号 AES や公開鍵暗号 RSA，楕円曲線暗号に対するスキャンベース攻撃手法に関する研究が奈良らにより報告されている [1–3]．奈良らの手法ではスキャンシグネチャと呼ばれるデータを利用することで，スキャンデータとレジスタの対応関係を特定する必要がなく，周辺回路のレジスタを含むスキャンチェインであっても攻撃手法を適用することができる．また，奈良らの手法は暗号 LSI が動作するタイミングも特定する必要がない．

共通鍵暗号方式 DES は 1976 年に米国商務省標準技術局（NIST）により連邦情報処理標準（FIPS）として承認された暗号システムである [6]．DES は認証レスポンスが高速であるため，現在でもクレジットカードや交通系電子マネー用 IC カードの通信を行う際の暗号化アルゴリズムとして広く使用されている．しかしながら，スキャンチェインのレジスタ接続順と動作タイミングに関する問題を解決した DES に対するスキャンベース攻撃手法は報告されていない．

本稿では，スキャンチェインのレジスタ接続順と暗号 LSI の動作タイミングに依存しない DES に対するスキャンベース攻撃手法を提案する．提案手法は，複数の平文を暗号 LSI に入力したときの暗号化処理中のスキャンデータの特定の 1 ビットに着目し，特定のレジスタの変化を観察することで秘密鍵を解読する．文献 [4] の手法で必要となるスキャンデータとレジスタの対応関係を特定する操作を行わずに秘密鍵の解読ができ，暗号化処理中のデータを格納するレジスタの一部を含むスキャンチェインが実装されてさえいれば，秘密鍵解読が可能となることを示す．また，暗号 LSI の動作するタイミングに非依存で，スキャンチェインに暗号 LSI 以外の周辺回路のレジスタが接続されていても秘密鍵解読が可能である．

2. 共通鍵暗号 DES [6]

共通鍵暗号 DES（Data Encryption Standard）は，64 ビットの秘密鍵を使用して 64 ビットの平文を暗号化する方式である．ただし，秘密鍵は 7 ビットおきにパリティビットが 1 ビットずつ挿入されているため，実際の暗号化に用いるビット数は 56 ビットである．DES の基本構造はラウンドと呼ばれる暗号化処理を繰り返し実行し，Feistal 構造と呼ばれる．

DES の暗号化処理を図 1 に示す．秘密鍵から鍵スケジュールによって生成されたラウンド鍵を使用して 16 回のラウンドを実行することで暗号化を行う．各ラウンドで実行される F 関数は，拡大転置 E，ラウンド鍵との XOR，S-Box 置換，転置 P に分けられる．i ラウンド目で実行される F 関数の動作を以下に示す．

拡大転置 E

32 ビットの入力を転置表に基づいて 48 ビットに並び替える変換を行う．($a = Expand(R_{i-1})$)

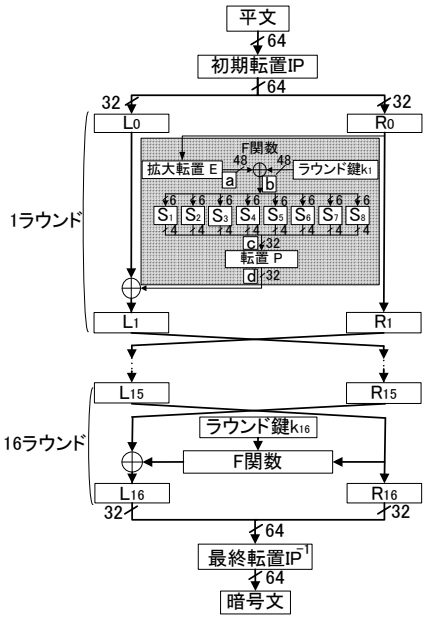

図 1 DES の暗号化処理.
Fig. 1 DES Encryption Algorithm.

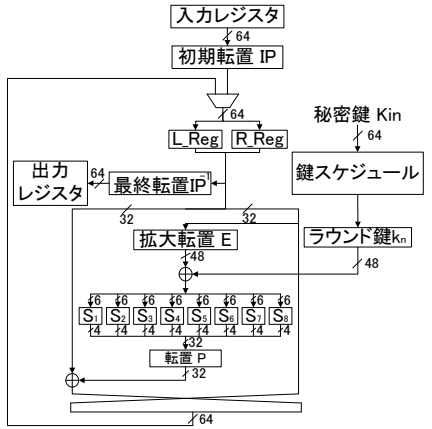

図 2 DES 暗号回路のハードウェアアーキテクチャ.
Fig. 2 DES Hardware Architecture.

ラウンド鍵との XOR

拡大転置 E の出力とラウンド鍵 k_n の排他的論理和を行う．($b = a \oplus k_i$)

S-Box 置換

8 通りの置換表 S1-S8 に基づいて，6 ビットから 4 ビットの非線形変換が行われる．それぞれの S-Box の出力は，S-Box の入力 6 ビットの先頭と最後尾を置換表の行，その他の 4 ビットを置換表の列に対応する値にする．($c = S\_box(b)$)

転置 P

32 ビットの入力を転置表に基づいて並び替える変換を行う．($d = permutation(c)$)

本稿では，ループ構造で実装された DES 暗号回路 [7] を攻撃対象とする．DES 暗号回路のハードウェアアーキテクチャを図 2 に示す．1 サイクル目で平文を入力レジスタに格納し，2 サイクル目で初期転置 IP の結果を L_Reg，R_Reg に格納し，3〜18 サイクル目で各ラウンド後の途中結果を L_Reg，R_Reg に格納し，19 サイクル目で最終転置 IP^{-1} を実行した結果を，暗号文を出力レジスタに格納する．

3. Yang らの手法: S-Box の性質を利用した鍵解読手法 [4]

スキャンベース攻撃とは，LSI に実装されたテスト用スキャンチェインから取得できるスキャンデータを中間値として秘密鍵を解読するサイドチャネル攻撃手法の一種である．スキャンベース攻撃は Yang らや奈良らにより報告されているが，DES に対するスキャンベース攻撃手法は，Yang らの手法 [4] だけであり，他の手法は提案されていない．本章では従来手法として Yang らの DES に対するスキャンベース攻撃の手法 [4] を説明する．

Yang らの手法では，秘密鍵の解読を 2 段階に分けて実行する．攻撃者はまずスキャンデータとスキャンチェイン内のレジスタの対応関係を特定する．次に，特定したスキャンデータとレジスタの対応関係をもとにラウンド鍵を解読し，秘密鍵を復元する．

3.1 スキャンデータとレジスタの対応関係の特定

秘密鍵の復元を実行する前にスキャンデータとレジスタの対応関係を特定する．暗号回路のスキャンチェインには 196 個のレジスタが含まれ，64 ビットの入力レジスタ，64 ビットの出力レジスタ，中間値を格納する各 32 ビットの L_Reg，R_Reg，4 ビットのコントロール用レジスタにそれぞれ割当てられている．

スキャンチェインを実装した LSI には，通常の動作を行うためのシステムモードと，スキャンデータを取得するためのテストモードがある．攻撃者は 2 つのモードを特定のタイミングで切り替えることで，スキャンデータがどのレジスタにそれぞれ対応するかを特定する．

まず攻撃者は 1 ビットだけ異なる平文 64 個を用意する．それらを暗号 LSI に入力して 1 サイクルと 2 サイクル後のスキャンデータを取得・比較することで，L_Reg，R_Reg，入力レジスタの 128 個のレジスタに対応するスキャンデータを特定することができる．

3.2 秘密鍵の復元方法

各ラウンドで使用される 48 ビットのラウンド鍵は 56 ビットの秘密鍵から鍵スケジュールを用いて生成される．また，鍵スケジュールはシフト演算と置換から構成されるため，複数のラウンド鍵を解読することで秘密鍵を復元できる．鍵スケジュールを分析すると，ラウンド鍵 (k_1, k_2, k_3) からパリティビットを除く DES の秘密鍵 56 ビットを復元できることが分かっている．Yang らの手法では，ラウンド鍵 k_1 をまず解読し，続いて k_2，k_3 と解読する．

3.3 ラウンド鍵の解読

平文とスキャンデータから決定できる F 関数の途中のデータをもとにラウンド鍵 k_1 を解読する．ラウンド鍵 k_1 は図 1 の F 関数の a と b を用いて $k_1 = a \oplus b$ と表せるため，1 ラウンドの a と b の値から求めることができる．a の値は平文に初期転置 IP と拡大転置 E を行うことで知ることができる．b の値は c の値に S-Box の逆変換を行うことで知ることができる．c の値は 1 ラウンド後のスキャンデータである L_1 と，平文から分かる L_0 から知ることができる．しかし，S-Box の置換は出力に対して入力の候補が 4 つ存在するため，c の値から b の値を一意に決定できない．

S-Box 置換を利用してラウンド鍵を解読するために，ラウンド鍵を 6 ビットずつに分割して解読する．6 ビットに分割することで，1 つの S-Box 置換だけに着目することができる．

3 つの平文を暗号 LSI に入力することで，c の値から b の値を一意に決定できない問題を解決する．まず 1 つ目の平文で c の値から b の値の候補を 4 つに絞る．2 つ目の平文で b の値の候補を 2 つに絞り，3 つ目の平文で b の値を決定できる．Yang らはこのように 3 つの特定の平文を入力することにより b を決定し，$k_1 = a \oplus b$ を実行することでラウンド鍵 k_1 を解読する．

ラウンド鍵 k_2，k_3 を解読する場合は，3 つの特定のデータを 1 ラウンド後，2 ラウンド後のレジスタに挿入することで，k_1 と同様にして解読できる．

4. スキャンシグネチャを用いた DES に対するスキャンベース攻撃手法

第 3 章で説明した Yang らの手法では，以下に示す仮定を置いているため，現実的な条件では秘密鍵解読ができないと考えられる．

(1)　スキャンチェインが暗号モジュールのレジスタだけで構成される必要がある．
(2)　特定のタイミングでスキャンチェインにアクセスする必要がある．

特定のレジスタ以外がスキャンチェインに含まれる場合，その他のレジスタの動作を止めることが難しいため，実際には Yang らの手法でスキャンデータとスキャンチェインのレジスタ接続順を特定することが難しい．また，暗号モジュールの動作タイミングは外部から分からないため，特定のタイミングでスキャンチェインにアクセスできない．

これらの問題を解決するアイデアを示す．DES の秘密鍵はラウンド鍵 k_1，k_2，k_3 から秘密鍵を復元することが可能であるため [4]，ここではラウンド鍵を解読する方法を示す．まず，複数の平文（$PT_0, PT_1, \ldots, PT_{n-1}$）を暗号 LSI に入力し，図 3 上のように暗号化処理中のスキャンデータ（$sd_0, sd_1, \ldots, sd_{n-1}$）を行列のように並べてみよう．このときのそれぞれの列データは，複数の平文を暗号化したときのある特定のレジスタの変化をそれぞれ表している．ラウンド鍵を変えて同じ複数の平文を入力すれば，暗号 LSI のあるレジスタに対応する列データは，ラウンド鍵に応じて変化する．即ち，平文数を十分大きくすれば，暗号 LSI のレジスタに対応する列データは，暗号 LSI のラウンド鍵に依存する固有のデータとなるため，ラウンド鍵を解読するための手掛かりとなる．この列データをもとにラウンド鍵を解読できると期待する．

攻撃者はまずラウンド鍵を予想する．予想したラウンド鍵を用いて，暗号シミュレータ

に暗号 LSI に入力したものと同じ複数の平文を入力し，暗号化処理中のデータを取得する．このデータを中間値と呼ぶ．次に，取得した中間値をスキャンデータと同じように行列のように並べる．予想したラウンド鍵に固有の中間値の列データがスキャンデータの列データに含まれるかを比較する．ラウンド鍵解読に用いる中間値の適当な列データをスキャンシグネチャと呼ぶ．スキャンシグネチャがスキャンデータの列データに存在すれば，予想したラウンド鍵は正しいとし，スキャンシグネチャがスキャンデータの列データに存在しなければ，予想したラウンド鍵は誤りとし，ラウンド鍵を予想し直して，比較を続ける．

図 3 にスキャンシグネチャを 1 ビット列として，ラウンド鍵が K_5 である場合を示す．攻撃者は複数の平文を入力し，スキャンデータを取得する．次にラウンド鍵を予想し，中間値をシミュレータから取得し，中間値のスキャンシグネチャとスキャンデータを比較していく．予想した K_0〜K_4 で取得した中間値のスキャンシグネチャはスキャンデータの列データに存在しないため，K_0〜K_4 は誤りである．K_5 で取得した中間値のスキャンシグネチャはスキャンデータの列データに存在するため，予想した K_5 が正しいラウンド鍵であるとする．

スキャンシグネチャを利用することで，中間値を保存する暗号 LSI のレジスタさえ含まれていれば，スキャンデータとスキャンチェイン内のレジスタの対応関係を特定することなくラウンド鍵解読ができる．また，特定のタイミングのスキャンデータを取得する必要はなく，中間値を保存する暗号 LSI のレジスタさえ含まれていれば，スキャンベース攻撃をすることができる．

中間値を各ラウンド後に暗号化処理の途中データが保存される L_Reg，R_Reg とすると，そのサイズは 64 ビットである．しかし，ラウンド鍵を解読するためには以下のような問題が生じる．

問題 1: ラウンド鍵のパターン数

ラウンド鍵は 48 ビットであるので，ラウンド鍵は全部で 2^{48} 通り存在し，ラウンド鍵の全てのパターンを試すことは不可能である．

問題 2: スキャンシグネチャとスキャンデータの比較方法

中間値が 64 ビットであるため，スキャンシグネチャの候補は 64 列存在し，スキャンデータは非常に大きなビット列となることもあり得るため，全ての列データの探索をすることは困難である．

本章では以上の問題を解決し，スキャンシグネチャを用いて秘密鍵解読を実行する提案手法を説明する．

4.1 問題 1 と問題 2 の解決方法

問題 1 を次のように解決する．図 1 の F 関数の 8 個の S-Box に着目するとこれらは互いに独立なので，ラウンド鍵を S-Box の入力サイズである 6 ビットずつの部分鍵 $(k_1^1, k_1^2, \ldots, k_1^8)$

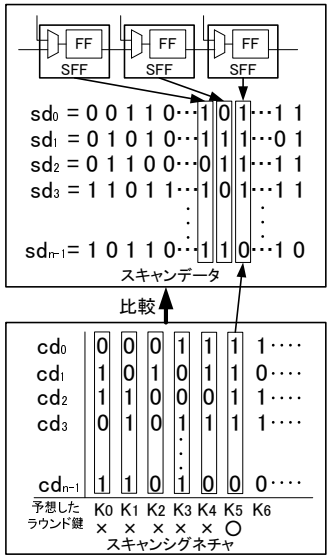

図 3 スキャンシグネチャを用いたスキャンベース攻撃のモデル．
Fig. 3 Scan-based attack using scan signature

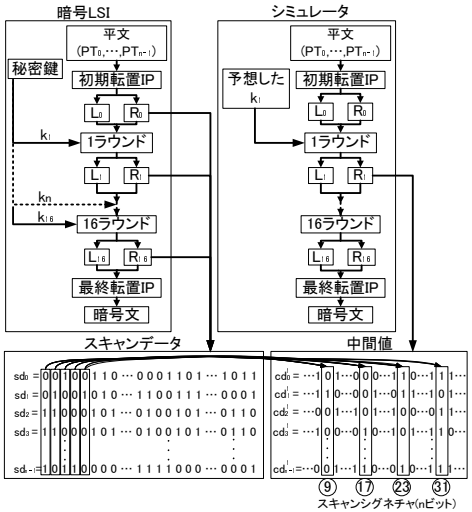

図 4 ラウンド鍵解読アルゴリズムのモデル (k_1^1 の場合)．
Fig. 4 Round key analysis algorithm.

に分割して解読できることに気付く．6 ビットの部分鍵が取り得るパターンは 2^6 通りであるため，分割して解読することで $2^6 \times 8 = 512$ 通りにラウンド鍵のパターンを削減することができる．

問題 2 を次のように解決する．6 ビットに分割した部分鍵を解読するため，スキャンシグネチャは解読する部分鍵の S-Box の出力に対応する中間値の 4 つのビット列とすることができる．例えば，k_1^1 の場合は S1 の出力 4 ビットに対応する中間値の 4 つのビット列をスキャンシグネチャとする．中間値のどのビットが S-Box の出力と対応しているかは，転置 P の置換表を用いることで決定できる．S1 の出力に対応する中間値のビットは 9，17，23，31 番目のビットである．したがって，k_1^1 を解読するためのスキャンシグネチャは中間値における 9，17，23，31 番目の 4 つのビットとなり，4 つのスキャンシグネチャとスキャンデータを比較することで k_1^1 を解読できる．

4.2 ラウンド鍵解読アルゴリズム

以上の議論をもとに，ラウンド鍵解読アルゴリズムを提案する．まず，攻撃者は複数の平文（$PT_0, PT_1, \ldots, PT_{n-1}$）を暗号 LSI に入力し，暗号化処理中のスキャンデータ

（$sd_0, sd_1, \ldots, sd_{n-1}$）を取得する．$k_1$ を解読するために必要な中間値として，1 ラウンド終了後の L_Reg，R_Reg に対応するデータを考える．攻撃者が k_1 を予想し，予想した k_1 を用いて DES 暗号アルゴリズムを実装したシミュレータで 1 ラウンド終了後の L_Reg，R_Reg に対応するデータを中間値 $(cd_0^1, cd_1^1, \ldots, cd_{n-1}^1)$ として取得する．このとき 6 ビットの部分鍵の解読を行うため，k_1^1 だけを予想し，$k_1^2, \ldots, k_1^8$ は 6 ビットデータの $00_{(16)}$ とする．

図 4 のように，取得したスキャンデータの列データに中間値の 4 つのスキャンシグネチャが含まれているかを比較していく．k_1^1 を解読するときのスキャンシグネチャは，中間値の 9，17，23，31 番目のビット列である．もしスキャンシグネチャがスキャンデータの列データに 1 つでも含まれる場合，予想した k_1^1 は暗号 LSI の k_1^1 と一致することになる．もしスキャンシグネチャがスキャンデータの列データに含まれない場合，予想した k_1^1 は間違っているため，k_1^1 を予想し直してシミュレーションを繰り返し実行する．同様に $k_1^2, \ldots, k_1^8$ も解読することができ，k_1 を解読が完了する．

k_2，k_3 の場合も 2 ラウンド後，3 ラウンド後の中間値をそれぞれ使うことで解読可能となる．k_2 の場合は，決定した k_1 を利用することで 2 ラウンドの入力を平文から計算することができるので，k_1 と同様に 6 ビットずつの部分鍵に分割して解読できる．k_3 の場合も同様に，決定した k_1，k_2 をもとに 3 ラウンドの入力を平文から計算ことができるので，k_1，k_2 と同様に解読可能となる．

4.3 改善手法

本節では，ラウンド鍵解読アルゴリズムをより高速・正確に実行するために以下に示す 2 つの改善手法を提案する．1 つ目は，F 関数の 8 つの S-Box の入出力は互いに独立しているという点に着目し，8 つの S-Box に対してラウンド鍵の解析を 48 ビット同時に実行することである．ラウンド鍵を 6 ビットずつに分割した部分鍵として解読するため，中間値を取得する回数が増加する．この手法を適用することで中間値の取得回数が 8 回から 1 回になるため，実行時間を短縮できる．

2 つ目は，スキャンチェインに含まれるレジスタがある程度分かっていれば効果を発揮する手法である．提案手法では，入力レジスタ，L_Reg，R_Reg，出力レジスタをスキャンチェインに含むことを条件としている．DES のアルゴリズムでは（$n-1$）ラウンドの L_Reg と，n ラウンドの R_Reg は同じデータとなるため，スキャンデータ中に同じ列データが必ず 2 つ存在する．また，L_Reg と R_Reg のデータを最終転置 IP^{-1} した後のデータを含む出力レジスタもスキャンチェインに含まれるため，スキャンデータ中に同じ列データが必ず計 4 つ存在する．したがって，4 つのスキャンシグネチャが，スキャンデータ中にちょうど 4 列ずつ含まれるまで比較することで，誤った部分鍵である場合に成功探索としてしまう間違いを減らすことができ，より精度の高い部分鍵の解読ができる．これによって，より少ない平文数で秘密鍵解読を実行できる．

また，2 つの改善手法は同時に実装することが可能である．

5. 実験・結果

本章では，提案手法を C で実装し，ランダムに生成した秘密鍵を 1000 個解読したときの実験結果を説明する．図 2 のハードウェアと同様の動作をする DES 暗号アルゴリズムを C で実装し，スキャンデータに対応するデータをシミュレータ実行中にファイルに出力する．取得したデータをもとに，提案手法のシミュレータを実行し，秘密鍵の解読を実行する．秘密鍵解読時間は提案手法のシミュレータを用いて 1 つの秘密鍵を解読する時間とする．本実験は，CPU が AMD Quad-Core Opteron 2360 SE 2.5GHz×2，メモリが 16GB の計算機を用い，コンパイラは gcc を用い，コンパイルオプションを-O3 とした．

次に示すの 2 つの場合で，提案手法で正しく 1000 個の秘密鍵を解読するために必要な平文数を評価した．

(1) Yang らの実験 [4] と同じ条件
(2) スキャンチェインの長さを変化させた場合

（1）では，Yang らの実験と比較するために，図 2 の入力レジスタ，L_Reg，R_Reg，出力レジスタの 192 個のレジスタでスキャンチェインが構成されると仮定する．

（2）では，スキャンチェインの長さを 192 ビット，512 ビット，1024 ビット，2048 ビット，4096 ビットに変化させて秘密鍵解読を行う．このようにすることで，擬似的にスキャンチェイン内に暗号 LSI 以外のレジスタを追加した状態を想定した実験を行うことが可能となる．

5.1 実験 1: Yang らの実験 [4] と同じ条件

本節では，Yang らの実験 [4] と同じ条件で実験を実行した結果を示す．提案手法と，改善手法を適用した提案手法の平文数と秘密鍵解読成功率の関係を示すグラフを図 5 に示す．提案手法では，55 個の平文を用いることで全ての秘密鍵の解読に成功した．また，改善手法を適用した提案手法では，27 個の平文を用いることで全ての秘密鍵の解読に成功した．

Yang らの手法と，提案手法と，改善手法を適用した提案手法で秘密鍵解読に必要な平均平文数のグラフを図 6 に示す．Yang らの実験では，スキャンデータとスキャンチェイン内のレジスタの対応関係を特定するために 64 個の異なる平文が必要となる．また，k_1，k_2，k_3 の解読のために，3 つの平文の入力と，6 つのデータを暗号化処理中のレジスタに挿入する必要がある．したがって，Yang らの手法では 67 個の平文と，6 つのデータをレジスタに挿入することで秘密鍵解読が可能となる．提案手法は平均 23.04 個の平文数で秘密鍵が解読でき，改善手法を適用した提案手法は平均 16.71 個の平文数で秘密鍵が解読できることが示さ

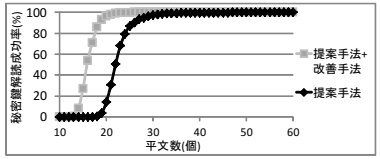

図 5 提案手法における秘密鍵解読成功率.
Fig. 5 Success Rate of Analysis.

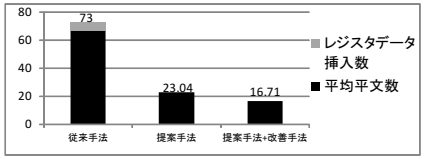

図 6 従来手法と提案手法の平均平文数の比較.
Fig. 6 Comparison of Conventional Method and Proposed Method.

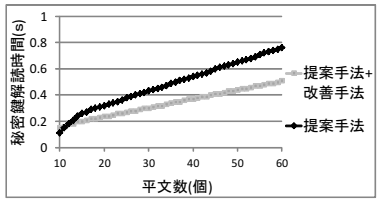

図 7 提案手法における秘密鍵解読時間.
Fig. 7 Analysis time.

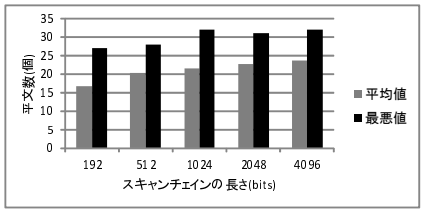

図 8 スキャンチェインの長さを変更した場合の平文数の比較.
Fig. 8 Number of plaintext V.S. Scan chain length.

れた.

提案手法と改善手法を適用した提案手法の平文数と秘密鍵解読時間の関係を示すグラフを図 7 に示す. 改善手法を適用した提案手法は 8 つの部分鍵の解読が同時に実行できるため, 提案手法より 30%程度高速にできることが示された.

また, 本実験ではスキャンデータを取得する範囲を暗号化処理が行われる 17 サイクルとしているため, 特定のタイミングでスキャンチェインにアクセスする必要がないことが示された.

5.2 実験 2: スキャンチェインの長さを変化させた実験

実験 2 では改善手法を適用した提案手法で秘密鍵解読を実行する. 図 8 にスキャンチェインの長さを変化させた場合に, 秘密鍵解読に必要な平文数の平均と最悪の値を示す. 実験結果によると, スキャンチェインの長さが 2 倍になったとしても, 平文数を 1 つ増加させることで秘密鍵解読が可能であることが示された. また, 追加したデータはランダムな値となっているため, 暗号 LSI のレジスタ以外スキャンデータが含まれた場合でも, 秘密鍵解読が可能であることが示された.

5.3 Yang らの手法との比較

提案手法では, 実験 1 の結果で示されたように, 解読に必要な平文数を Yang らの手法の半分以下にすることができた. また, 提案手法ではスキャンチェインからレジスタにデータを挿入する必要がないため, 暗号 LSI の動作タイミングを知る必要がなく, 特定のタイミング以外のスキャンデータを含む場合であっても, 秘密鍵解読が実行できることを示した. 実験 2 の結果に示されるように, スキャンチェインが暗号 LSI の特定のレジスタを含む場合も秘密鍵解読を実行することができた. 以上より, 提案手法は Yang らの手法では必要な仮定を置かない場合でも秘密鍵解読を実行することができる.

6. おわりに

本稿では, スキャンチェイン構造に依存しない DES に対するスキャンベース攻撃手法を提案した. 従来手法はスキャンチェインが暗号 LSI のレジスタだけから構成されることや, 特定のタイミングのスキャンデータが必要となるため, 秘密鍵解読を実行するための条件が現実的なものではなかった. 提案手法は, スキャンチェインに暗号化処理中のデータを格納するレジスタが含まれてさえいれば良く, スキャンチェインの構造特定や動作タイミングを知る必要がないため, より現実的な条件で秘密鍵解読が実行できることを実験結果より示した. また, 必要な平文数も従来手法よりも少なくすることができた点においても, 従来手法よりも優れた手法であると言える.

今後の研究課題としては, 実際に FPGA 等のハードウェアに実装した DES 暗号回路に対して, 提案手法を適用できるかを実験することが考えられる.

謝辞 本研究の一部は, セコム科学技術振興財団の助成による.

参 考 文 献

1) R.Nara, N.Togawa, M.Yanagisawa and T.Ohtsuki, “A scan-based attack based on discriminators for AES cryptosystems,” *IEICE Trans. on Fundamentals of Electronics, Communications and Computer Sciences*, vol.E92-A, no.12, pp. 3229–3237, Dec.2009.
2) R.Nara, K.Satoh, M.Yanagisawa, T.Ohtsuki and N.Togawa, “Scan-based side-channel attack against RSA cryptosystems using scan signatures,” *IEICE Trans. on Fundamentals of Electronics, Communications and Computer Sciences*, vol.E93-A, no.12, pp.2481–2489, Dec.2010.
3) R.Nara, M.Yanagisawa, T.Ohtsuki and N.Togawa, “Scan vulnerability in elliptic curve cryptosystems,” *IPSJ Trans. System LSI Design Methodology*, vol.4, February issue, pp. 47–59, Feb.2011.
4) B.Yang, K.Wu and R.Karri, “Scan based side channel attack on dedicated hardware implementations of data encryption standard,” in *Proc. of the International Test Conference*, pp.339–344, 2004.
5) B.Yang, K.Wu and R.Karri, “Secure scan: a design-for-test architecture for crypto chips,” *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol.25, no.10, pp.2287–2293, 2006.
6) “FIPS 46-3, Data Encryption Standard (DES),” http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf.
7) “DES hardware macro specification,” http://www.aoki.ecei.tohoku.ac.jp/crypto/items/DESSpec2007Sep25.pdf.