ISSN 2186-5647

7-66

AES 暗号回路におけるトロイ設計の影響評価

日大生産工（院）○荻田 英実　日大生産工　細川 利典
九大　吉村 正義

1. はじめに

近年，大規模集積回路(Large Scale Integrated circuits:LSI)の製造は，海外工場に委託されることが多い[1]．生産コストが安い国に外注することにより，製造コストの削減が図られる．一方で，外注先において，攻撃者による LSI の設計および製造工程への攻撃が容易になっている[1].

LSI の設計，製造過程における攻撃の一例として，トロイ回路の混入がある[1]．トロイ回路とは，攻撃者が設定した特定のトリガー条件により起動し，攻撃動作を行う付加回路である．トロイ回路が動作することにより，LSI が組込まれた製品やシステムは，正常な機能の無効化，機密情報の漏洩や改ざん，回路破壊といった脅威にさらされる[1,2]．トロイ回路の挿入により，もととなる回路の物理的な外観は損なわれない．また，入出力動作が回路の設計仕様を満たすように設計される．さらに，トロイ回路として動作するのはごく限られたトリガー条件下のみである．このため，一般的な LSI の機能検証やテストでは，トロイ回路の発見が困難である[1,2,3]．現在では製品に混入したトロイ回路を検出するための様々な研究が行われている[2].

本論文では，トロイ回路を挿入する対象回路として，128 ビットの暗号鍵に対応した AES(Advanced Encryption Standard)暗号回路[3,4]を用いる．そして，トロイ回路の攻撃機能として，AES 暗号回路の暗号鍵を逆算する情報を取得する機能を実装し，その影響を評価する．

2. トロイ回路

2-1.トロイ回路

トロイ回路とは，LSI の設計もしくは製造段階において攻撃者により挿入される，特定条件下でもとの回路に対して攻撃動作を行う回路のことである．トロイ回路は一般的に，起動条件の判定を行うトリガー部と攻撃動作を行うペイロード部から成る[5,6].

トリガー部とは，もとの回路の状態や信号線の値が，攻撃者の設定したトロイ回路の起動条件を満たすかを判定する回路である．起動条件の一例としては，回路内のカウンタが特定の値になる，対象信号線に特定の制御値が入力されるなどがある[6,7].

ペイロード部とは，もとの回路に対して，機密情報の漏洩や回路機能の無効化，あるいは回路破壊などの攻撃動作を実行する回路である．ペイロード部はトリガー部で設定された起動条件を満たす場合のみ，動作する[6,7].

2-2.トロイ回路の検出手法

トロイ回路検出の従来法の 1 つとして，電力解析を用いた手法がある．Agrawal らは，回路を動作させた際に発生する消費電力に着目し，回路内にトロイ回路が挿入されたか否かを判定するための指標とした[1]．トロイ回路が起動条件を満たすと，通常の回路とは異なる消費電力影響が現れる．しかしながら，回路の消費電力影響は，もとの回路面積に対してトロイ回路が占める割合が小さいほど少なくなる．また，消費電力を指標とする場合，回路動作時の微小なノイズや，回路自身の製造ばらつきも問題となる．Agrawal らは，これらの問題を主成分分析(Principal Component Analysis:PCA)を用いることで改善した．Agrawal らは PCA を用いて複数の回路の消費電力パターンを分析し，トロイ回路判別に用いる指標の設定を行った．これにより，製造ばらつきを含む回路においても，トロイ回路検出が可能になった.

本論文では，消費電力影響，並びに面積影響をなるべく抑え、従来手法でも検出が困難なトロイ回路の実装を目指す.

3. AES 暗号回路

3-1.AES 暗号

AES 暗号とは，2001 年にアメリカ国立標準技術研究所(National Institute of Standards and Technology:NIST)で規格化された共通鍵暗号方式の暗号である[3,4].

AES は平文サイズが 128 ビットで固定される．鍵のサイズは 128，192，256 ビットの 3 種類が用意されているが，本論文では鍵長が 128 ビットの AES を扱う.

AES において，鍵と平文を EXOR 演算やシフト演算によって暗号化する処理をラウンド処理と呼ぶ．図 1 に AES の暗号化構造を示す．128 ビット鍵対応の AES のラウンド処理回数は 10 回である．AES の暗号化構造は図 1 のように，暗号化処理を行う暗号化部と，各ラウンドで使用する鍵を生成する鍵拡張部からなる.

Evaluation of the Effect for Hardware Trojan Designs in AES Encryption Circuits

Amy OGITA, Toshinori HOSOKAWA, and Masayoshi YOSHIMURA

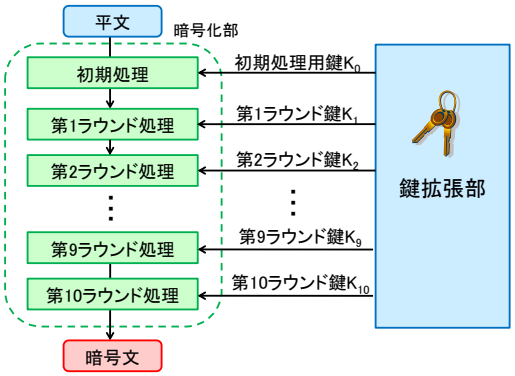

図 1. AES の復号化構造

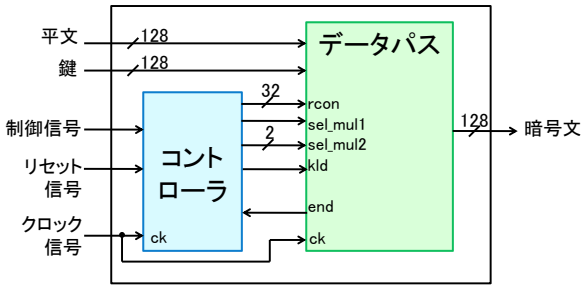

図 2. AES 暗号回路実装概要

AES は，ラウンド処理を繰り返すことにより，暗号文を生成する．ラウンド処理に使用する鍵は，鍵拡張部を用いてラウンドごとに変化させる．

3-2.AES 暗号回路

本論文において使用する AES 暗号回路の実装概要を図 2 に，コントローラおよびデータパスをそれぞれ図 3，4 に示す．本論文で使用する AES 暗号回路は，暗号化処理を実行する前に，リセット信号線からリセット信号線に'1'を入力し，データパス内の信号線を初期化する．リセット処理実行後，リセット信号を'0'に戻し，外部入力から制御信号線へ制御値'1'を入力すると，平文，鍵を初期入力として AES の暗号化処理を開始する．暗号化処理は図 4 に示す状態遷移図の状態 ST2 から状態 ST11 の間行われる．そして，最終状態 ST12 で暗号文が出力され，回路処理が終了する．

4. トロイ回路設計

4-1.トロイ回路混入の前提

LSI の製造は，1 つのメーカー内で設計から販売までの全工程を行う場合もあるが，設計，製造，販売，それぞれの業務に特化した企業が各工程を分担することもある．本論文では図 5 に示すように，A 社が仕様設計から論理設計までを，B 社がテスト容易化設計からレイアウト設計を，C 社が製造業務をそれぞれ担当するものとする．

通常，A 社は自社で顧客の要求仕様から論理設計までを行う．その後，論理設計を行った回路データを，B 社に渡す．B 社は，A 社から渡された回路データをもとに，テスト容易化設計を行ってから，レイアウト設計を行う．その後，B 社

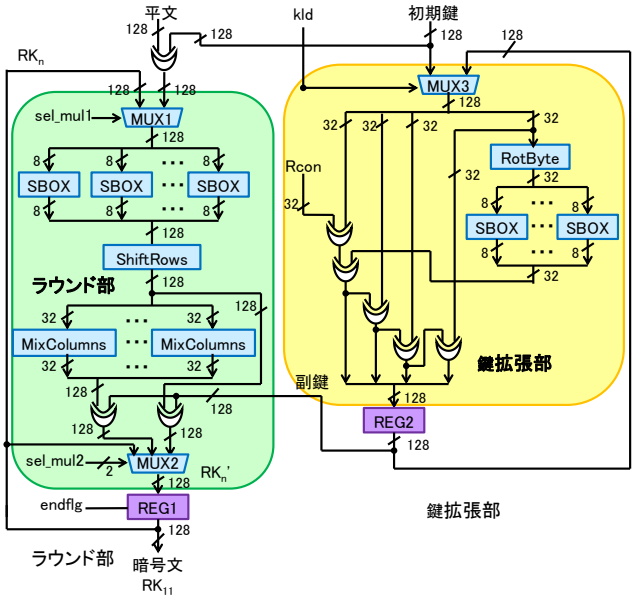

図 3. AES データパス図

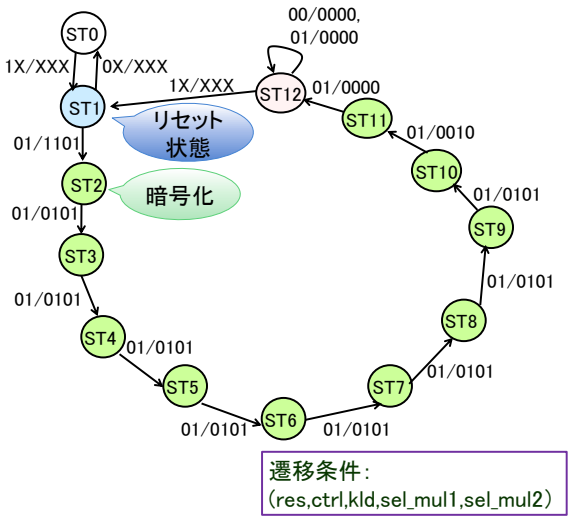

図 4. AES コントローラ図

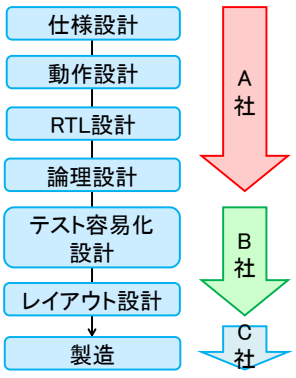

図 5. LSI の設計フロー

はレイアウト設計の完了した回路データを依頼主である A 社に納入する．A 社は，B 社から納品される回路データを検証し，その後，回路データを C 社に渡し，製造を依頼する．このような流れで，通常の LSI の設計，製造フローが進むもの

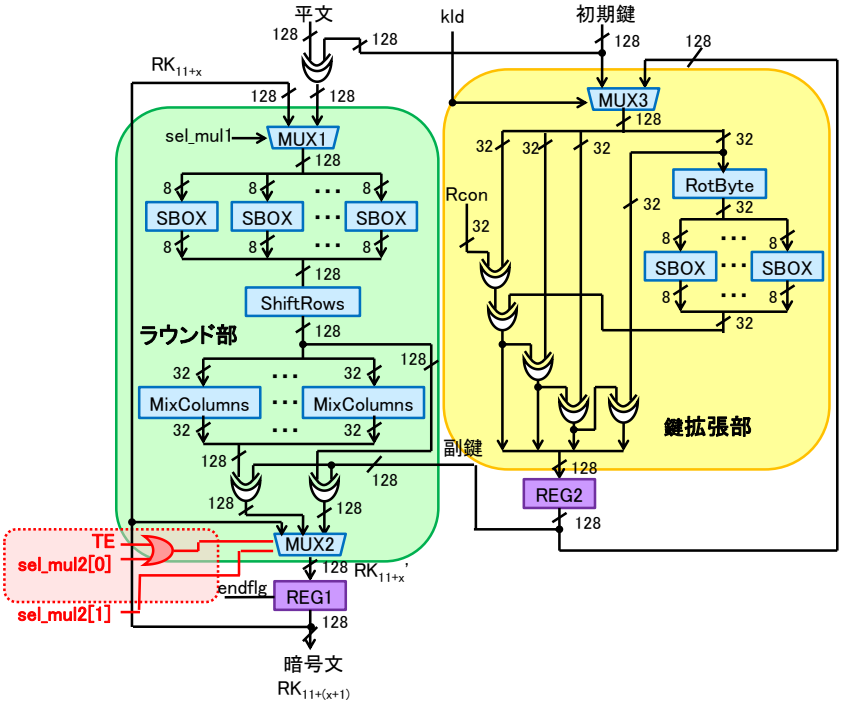

図６．トロイ回路実装

とする．上記の設計，製造フローに対し，攻撃者が攻撃を仕掛けることを考える．本論文では，テスト容易化設計およびレイアウト設計を担当する B 社において，攻撃者がテスト容易化設計に乗じてトロイ回路を混入させると仮定する．これは，テストモード時の動作と通常動作の等価性検証は困難であるため，テスト容易化設計後にトロイ回路が混入しても，その有無を検証するのは困難であると想定されるためである．

A 社は AES 暗号回路のテスト容易化設計を依頼する．B 社は通常，テスト容易化用の機能としてスキャンベースの組込み自己テスト(Built In Self Test:BIST)機能[8]をもとの回路に追加する．以上より，本論文における「テストモード」機能とは，スキャンベース BIST を動作させるために使用されるテスト専用の機能である．

B 社の攻撃者は，スキャンベース BIST 実装時，スキャンベース BIST とは別にトロイ回路機能を A 社回路内に混入させる．そして，「テストモード」機能が動作すると同時に，トロイ回路が動作するような条件を設定する．

4-2.トロイ回路仕様

通常動作において，AES 暗号回路から出力される暗号文を RK_{11} とする．暗号文 RK_{11} が出力されてから x 時刻後に，AES 暗号回路から暗号文が出力できると仮定すると，その暗号文は時刻 x(x は任意の整数)を用いて，RK_{11+x} と表現することができる．RK_{11+x} は通常動作においては出力することができない情報である．

トロイ回路の機能として ，暗号化処理終了後に回路内で計算される暗号文 RK_{11+x} を取得する機能を実装する．RK_{11+x} の情報を用いて，AES 暗号回路内で扱う暗号鍵の導出する．トロイ回路の起動条件としては，暗号化処理終了後，テストモードが ON になる場合を起動条件として設定する．

AES 暗号回路のデータパス部に対するトロイ回路の実装方法を図 6 に示す．図 6 の破線部分で囲われた箇所がトロイ回路部である．図 3，4 より，本論文で用いる AES 暗号回路は，暗号化処理が完了する状態 ST12 において暗号文 RK_{11} が出力される．このとき，回路のテストモードを利用する際に用いられる制御信号"TE"に制御値'1'を入力すると，図 6 のトロイ回路部において，マルチプレクサ MUX2 の制御信号 sel_mul2 に"01"が入力される．これにより，マルチプレクサ MUX2 は RK_n'に MixColumns モジュールの演算結果を出力する．ホールド機能付きレジスタ REG1 を制御する信号線"endflg"は，暗号化処理終了後においてアクティブである．よってこれにより，MixColumns モジュールの演算が再開されると同時に，その結果が暗号文 RK_{11+x} として回路外部に出力できるようになる．

なお，MUX2 の制御値と出力値の関係は表 1 に示す通りである．また，図 3，6 におけるマルチプレクサ MUX2 の制御値と出力信号線の対応は，最右端から制御値"00"の出力，制御値"01"の出力，制御値"10"の出力である．

表 1．MUX2 の割当て表

sel_mul2	出力値
00	状態保持(RK_n)
01	MixColumnsモジュールの演算結果
10	ShiftRowsモジュールの演算結果
11	0

4-3.鍵抽出アルゴリズム

トロイ回路から得られる暗号文 RK_{11+x} および $RK_{11+(x+1)}$を用いて，暗号鍵を導出するアルゴリズムのフローチャートを図 7 に示す．

(step1)：AES 暗号回路を動作させ，出力値 RK_{11} を得る．

(step2)：トロイ回路を動作させ，暗号文 RK_{11+x} および

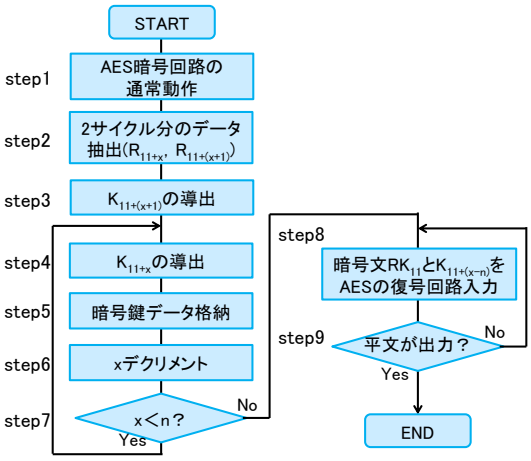

図 7．暗号鍵抽出アルゴリズム

$RK_{11+(x+1)}$(x は任意の値)を AES 暗号回路から出力する．

(step3)：式 1 を用いて 11+(x+1)ラウンド目の秘密鍵 K_{11+x} を導出する．

$$K_{11+x} = RK'_{11+x} \oplus RK_{11+(x+1)} \cdots (式\ 1)$$

式 1 中の RK'_{11+x} とは，トロイ回路が起動し，疑似暗号化処理状態になった際に，図 6 の RK_{11+x} をラウンド部モジュールに入力し，得られる RK'_{11+x} のことである．

(step4)：RK_{11+x} を暗号化する際，用いられた秘密鍵 $K_{11+(x-1)}$ を，式 2 を用いて導出する．

$$K_{11+(x-1)} = KeyExp^{-1}(K_{11+x}, Rcon) \cdots (式\ 2)$$

関数 $KeyExp^{-1}$ は，AES 復号時に用いられる暗号鍵の逆算を行う関数である[3]. Rcon は変数である．式 2 では，Rcon=0 の定数となる．これは，Rcon のデフォルト値が 0 で設定されているためである[2].

(step5)：step3 で導いた秘密鍵 $K_{11+(x-1)}$ を格納する．

(step6)：x を 1 減らす．

(step7)：x が繰り返し回数 n 未満(n は 0 以上の任意の整数値)であれば，Step4 に戻り，秘密鍵を再度式 2 より導出，格納する．

(step8)：暗号文 RK_{11} と $K_{11+(x-n)}$ を AES の復号回路に入力し，復号化処理を行う．

(step9)：平文の出力が成功した場合は，step8 で用いた $K_{11+(x-n)}$ が，図 1 における AES の第 10 ラウンド処理で使用される第 10 ラウンド鍵 K_{10} と同一であると判定できるので，鍵抽出処理を終了する．平文の出力に失敗した場合は，n を 1 つ減らし，step8 の処理を繰り返す．

5. 実験結果

3 章で設計した AES 暗号回路に対し，4 章で設計したトロイ回路を組込み，その有無による比較結果を表 2 に示す．表 2 より，トロイ回路挿入により AES 暗号回路の面積は 0.01%増加した．また，実装したトロイ回路は暗号文 RK_{11+x} の取得に成功した．

表 2. トロイ回路挿入の有無による面積比較

	面積(ゲート数)
AES暗号回路（トロイなし）	22395
AES暗号回路（トロイあり）	22398

6. おわりに

本論文では，AES 暗号回路に対するトロイ回路設計を行った．提案したトロイ回路は，面積面でもとの AES 暗号回路と差異が少なく，面積影響からのトロイ検出が困難であると考えられる．

今後の課題としては，挿入したトロイ回路のトグル回数，消費電力，テスト容易性の比較，鍵抽出アルゴリズムの検証が挙げられる．

7. 参考文献

[1] Dakshi Agrawal, Selcuk Baktır, Deniz Karakoyunlu, Pankaj Rohatgi and Berk Sunar, "Trojan Detection using IC Fingerprinting", 2007 IEEE Symposium on Security and Privacy, pp.296-310, 2007.

[2] Mohammad Tehranipoor, Farinaz Kou-shanfar,"A Survey of Hardware Trojan Taxonomy and Detection", IEEE Design & Test of Computers, pp.10-25, Jan/Feb. 2010,

[3] 神永正博，山田聖，渡邊高志，"Java で作って学ぶ暗号技術-RSA,AES,SHA の基礎から SSL まで"，森北出版，2008, pp. 115-142.

[4] National Institute of Standards and Technology, "Advanced Encryption Standard (AES)", FIPS PUB 197, Nov. 2001.

[5] H. Salmani, M. Tehranipoo and J. Plusquellic, "New Design Strategy for Improving Hardware Trojan Detection and Reducing Trojan Activation Time", 2009 IEEE International Workshop on Hardware-Oriented Security and Trust (HOST'09), pp.66-73, July 2009.

[6] Yier Jin, Nathan Kupp and Yiorgos Ma-kris, "Experiences in Hard-ware Trojan Design and Implementation", 2009 IEEE International Workshop on Hard-ware-Oriented Security and Trust, 2009.

[7] X. Wang, M. Tehranipoor, and J. Plusquellic, "Detecting Malicious Inclu-sions in Secure Hardware: Challenges and Solutions", Proc. IEEE Int'l Work-shop Hardware Oriented Security and Trust (HOST 08), IEEE CS Press, pp. 15-19, 2008.

[8] M. Arai, S. Fukumoto, K. Iwasaki, T. Hiraide, T. Aikyo, "Test Data Compression Using TPG Reconstruction for BIST-Aided Test", proc IEEE 6th Workshop on RTL and High Level Testing, 211-8588, 2005.