7-51

AES 設計におけるトロイ生成の一考察

日大生産工（学部）○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，ハードウェアの集積回路(Large Scale Integration circuits:LSI)製造は，海外工場に委託されることが多い[1]. LSI 製造を生産コストが安い国に外注することにより，製品コストの削減が進んでいる．一方で，第三者による LSI サプライチェーンへの攻撃が容易になっている[1].

LSI サプライチェーンに対する攻撃の一例としては，製品運送時のすり替えやトロイ回路の混入がある[1]．トロイ回路とは，攻撃者が設定したトリガー条件により起動し，回路の無効化や機密情報を外部に漏洩させるものである[1][2].

ハードウェアに仕掛けられるトロイ回路は，基となる回路の物理的外観を損なわず，入出力動作が基の回路と同じになるように設計される．また，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になる．このため，一般的な LSI の機能テストでは，トロイ回路の発見がほぼ不可能である[1][2][3]．現在では製品に混入したトロイ回路を発見するための様々な研究が行われている[2].

本稿では，トロイ回路を製品や設計段階の LSI から除去するための研究の前段階として，実際にどのようなトロイ回路を LSI へ挿入することができるかを検証し，報告する．対象回路としては，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，今後システムに採用される機会が増加することが予想される AES(Advanced Encryption Standard)暗号の暗号化回路[4][5][6][7]を用いる.

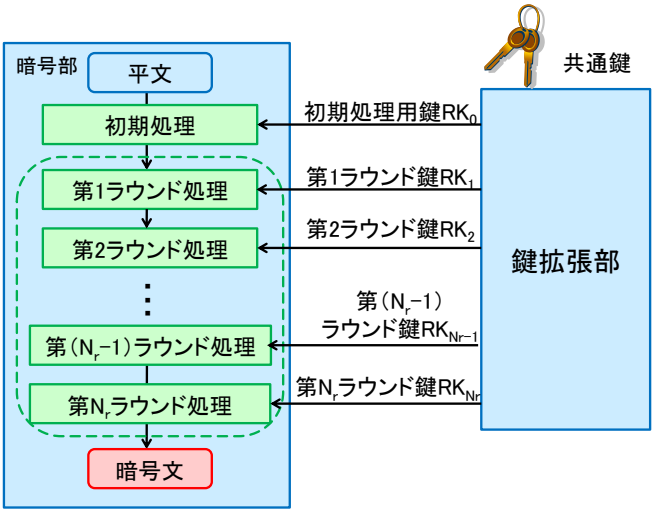

図 1. AES の暗号化構造

2. AES 概要

AES 暗号とは，米国で新たに規格化された共通鍵暗号方式である[4][5].

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，本稿では鍵長 128 ビット対応の AES を扱う.

AES では，鍵と入力文を 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と呼ぶ. 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.

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.

初期処理を除いたラウンド処理の繰り返し回数は，入力する鍵のビット数に応じて表 1 ように変化する.

ラウンド回数 N_r は式 1 により求められる.

$$N_r = N_k + 6 \cdots (1)$$

Investigation of Hardware Trojan Fabrication on the AES Design

Amy OGITA, Toshinori HOSOKAWA, and Masayoshi YOSHIMURA

表 1. AES の処理回数

	鍵長(N_k) 単位：ワード	ラウンド数(N_r) 単位：回数
AES-128	4	10
AES-192	6	12
AES-256	8	14

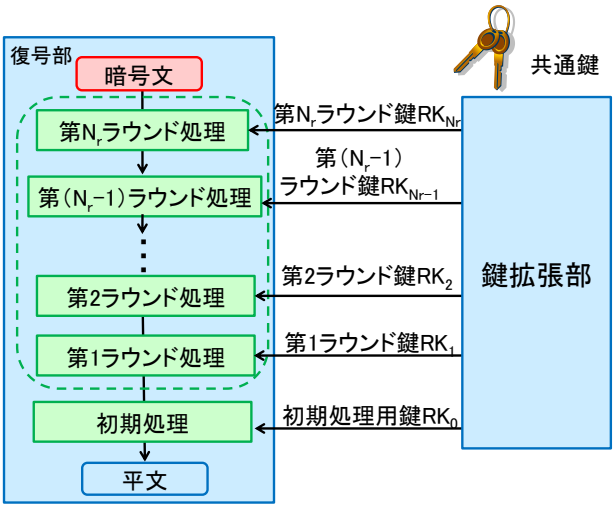

図 2. AES の復号化構造

N_k とは，鍵長をワード単位で表現した値である．鍵長が増すほどラウンド回数および鍵の拡張回数が増加するため，より強固な暗号化が可能である．

3. ハードウェアトロイ

世界的な市場経済が製品の低コスト化を求めた結果，多くの LSI 製造元は，人件費などの生産コストが安い国において製品の委託生産を行った．これにより LSI 製造元は，ハードウェアの回路部品や製品のコスト削減に成功した．しかしながら，これは同時に，LSI の製造プロセスにおける攻撃が，従来に比べて実行しやすくなるという弊害を招いた．

特に，LSI の設計，製造，流通プロセスの一連の流れの中で，製品化前の回路に対して，攻撃者が設計者の意図しない機能を実行するように追加した回路部分を総称してハードウェアトロイと定義する[1][2]．

ハードウェアトロイの混入方法は様々である．具体的な混入手法としては，

- 海外工場における LSI 製造中に意図的に回路を改ざんする
- LSI の機能や仕様などの要求定義のみを依頼主が設定し，LSI 設計を海外企業に委託した際，海外企業側で改ざんを加える
- 海外工場で完成した回路を，販売元となる LSI 製造元に輸送する間に，攻撃者が製造したハードウェアトロイ入り回路と正規の回路をすり替える

などがある．

一度混入したハードウェアトロイは，製品としてユーザーに出回った際に，様々な弊害をもたらす．一例としては，ユーザーが気付かないうちに，製品の回路内で扱う秘密情報を外部の攻撃者に漏洩してしまうものがある．特に，重要情報を扱う暗号回路において，暗号化・復号化の仕組みや鍵情報が漏洩してしまうと，システムや製品の信頼性を損ねるという問題がある[3]．

4. AES におけるハードウェアトロイ実装

4-1. ハードウェアトロイの機能

AES の暗号化アルゴリズムのアーキテクチャを図 1 に，復号化アルゴリズムのアーキテクチャを図 2 に示す．AES の復号化処理は，暗号化処理を真逆にすることで実行される[5]．

AES は，暗号化と復号化に同じ鍵を使用する[5]．また，AES の暗号化・復号化手順は公開されており，回路の設計・製作は容易である．

つまり，鍵情報を AES 回路から取り出すことができれば，暗号文を復号化回路に入力し，平文を取り出すことが可能であると考えられる．よって，本稿で開発するハードウェアトロイの機能としては，第 N_r ラウンド処理，すなわち最終ラウンド処理で使用された鍵情報の取り出しを設定する．

4-2. ハードウェアトロイのトリガー条件

AES は，鍵の長さによって規定された回数のラウンド処理を実行する(表 1)．本稿では AES の処理回数をカウントする変数に注目

する．

本稿で扱う 128 ビット鍵対応の AES のラウンド処理の回数は表 1 より，10 回である．値'10'を 2 進数表現する場合，'1010'となり，4 ビット必要である．4 ビットのデータを確保した場合，2 進表現で'0000'～'1111'となり，10 進数としては 0~15 まで表現可能である．0 から 15 までの値のうち，暗号化処理として図 1 の初期処理およびラウンド処理を 0~10 の値に定義すると，残る 11~15 の 5 つの値は未定義となる．本稿ではこの未定義になるカウント数を利用してハードウェアトロイのトリガー条件を実装する．

トリガー条件としては，以下を設定する．

① カウント数が既定の処理回数に達し，暗号化処理が終了する
② 外部入力から特定の値が入力される

①の条件は，AES の暗号化処理終了を待つための条件である．①を満たした状態で②の条件が外部から与えられると，AES 回路内でトロイ機能を設定したカウント値が与えられる．これによりトロイ機能を起動させる．

4-3. ハードウェアトロイの動作

ハードウェアトロイ付き AES の実装例を図 3 に示す．本稿で実装する AES は，回路のコントロールを行う組合せ回路 C1 に平文，鍵，制御値を入力して動作させる．組合せ回路 C1 には，初めに平文，鍵を記憶装置(Random Access Memory:RAM)に入力する制御信号を与える．RAM に平文，鍵を入力したら，AES の暗号処理を実行する制御信号を AES 回路に与える．通常動作では，AES 回路の暗号化処理が終了すると，AES 回路の外部出力に暗号文が出力され，回路処理が終了する．

攻撃者がハードウェアトロイのトロイ機能を使用する場合は，AES 回路から暗号文が出力されている状態で，攻撃者が設定した特定の制御値を組合せ回路 C1 に与える．すると，4-2 章で示したトリガー条件①②を満たす状態になり，組合せ回路 C1 内部の処理カウンタ値が，通常の AES 回路処理では使用

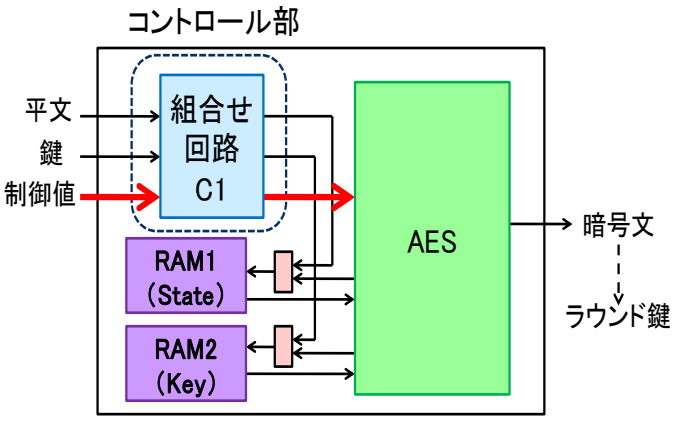

図 3. ハードウェアトロイ付き AES の実装例

しない未定義値に変化する．通常動作としては未定義とされているカウンタ値には，4-1 章で示した「最終ラウンド鍵を外部出力に出力する」機能をあらかじめ攻撃者が隠し機能として設定しておく．

これにより，4-2 章で示したトリガー条件①②が攻撃者の操作で満たされた場合，外部出力に出力するデータを暗号文から最終ラウンド鍵に切り替え，鍵情報を得ることが可能となる．

5. 実験結果

実験として，文献[4][5][6]を参考に設計した AES 回路に対し，4-3 で述べた処理を実行するハードウェアトロイ回路部を組み込み，ハードウェアトロイ回路の有無によって，回路面積，発生する故障の数とその検出率，テストパターン数とテスト生成時間の違いを比較した．

実験結果を表 2 に示す．表 2 において，「全故障数」は回路上にあるすべての故障数，「検出故障数」は全故障数のうち，生成したテストパターンを用いて検出可能な故障の数，「故障検出率」は全故障に対する故障の検出率，「故障検出効率」は検出可能な故障に対する故障の検出率，「テストパターン数」は生成したテストパターンの数，「テスト生成時間」はテストパターン生成に要した時間，「回路面積」は論理合成後の回路の面積である．

表 2. トロイ挿入の有無による AES 回路の比較

	トロイなし	トロイあり
全故障数	73082	74826
検出故障数	73062	74807
故障検出率(%)	99.97	99.97
故障検出効率(%)	100.00	100.00
テスト生成時間(s)	1.34	1.45
テストパターン数	335	334
回路面積	21874	22229

表 2 より，故障の数やテスト生成時間，回路面積は，ハードウェアトロイを含む回路のほうが，含まない回路よりも多くなることがわかる．特に，回路面積と故障の数は大幅に増加している．ハードウェアトロイの動作を実現する際, 図 3 より, 秘密情報の出力には，通常動作時に，暗号文の出力に使用する信号線と共通の信号線を使用する．このため，入出力信号線数については，ハードウェアトロイの有無に関わらず変化はない．よって，暗号文を出力する通常動作状態から，鍵情報を出力するトロイ動作状態へ切り替える処理を実現する追加回路部分について，本来の AES 回路では必要のない回路部品が余分に付加され，実験結果に表れたと考えられる．

一方で，テスト生成時間の増加に対し，テストパターン数は減少している．これは，トロイ回路の有無により，論理合成後の回路構造が変更されてしまい，結果としてハードウェアトロイが含まれない回路よりも，ハードウェアトロイを含む回路のほうがテスト容易な回路に変化したと考えられる．

6. おわりに

実験結果より，本稿で設計したハードウェアトロイ回路は，基の回路に対して回路面積や故障検出などのパラメータ値の差異が大きく，ハードウェアトロイとして一般的に定義される[1][2][3]回路とはほど遠いものになってしまった．

本稿で設計したハードウェアトロイ回路は，外部入出力数を増やさないことに重点を置いた設計であった．よって，今後の課題としては上記の条件に加え，回路のパラメータ値への影響が少ない回路設計，アルゴリズムの考案が必要である．

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 , 2007.

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

[3] Yier Jin, Yiorgos Makris, "Hardware Trojans in Wireless Cryptographic ICs ", IEEE Design & Test of Computers, Jan/Feb. 2010, pp.26-35.

[4] 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.

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

[6] 森岡澄夫, "数学いらずの AES 暗号 SubBytes 設計ガイド", Design Wave Magazine, CQ 出版, 2004, pp.152-157.

[7]下村高範, 阿部光輝, "暗号化アルゴリズム Rijndael のハードウェア実装と評価", システム LSI 設計技術研究会, 2003, pp. 13-17.