ISSN 2186-5647

2-27

レジスタ転送レベルにおけるアンチ SAT に基づく論理暗号化法

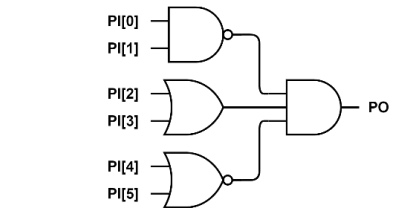
日大生産工(院)　○辻川 敦也　　日大生産工　細川　利典
京産大　　吉村　正義

1. はじめに

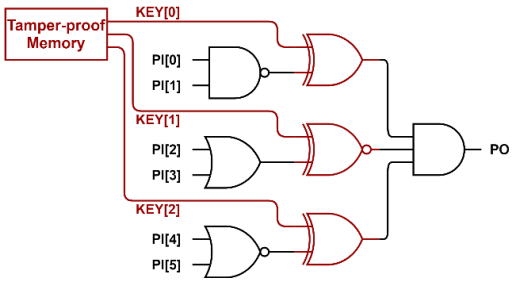
近年，超大規模集積回路(Very Large Scale Integrated circuits：VLSI)の大規模化に伴い，VLSI設計会社1社のみでVLSIの設計・製造を行うことが困難となっている．それゆえ，IPベンダよりIPコア(Intellectual Property Core)と呼ばれるVLSIを構成する部分的な回路情報を購入し，必要な部分のみ自社で設計する手法が多く用いられている．また，設計会社のファブレス化が進み，設計したVLSIの製造をファウンドリに外注するようになっている．しかしながら，信頼性の低いファウンドリによるトロイ回路の挿入，海賊版の闇市場での販売などの著作権侵害が問題となっている．その著作権侵害を防ぐ論理暗号化手法がIP設計技術として提案された [1]．論理暗号化を施した回路は鍵を入力する必要があり，正しい鍵は設計者が保持し，ファウンドリには知らされていない．回路の機能は，鍵が正しい場合にのみ正しいものとなる．ファウンドリが論理暗号化された回路を製造して設計会社に返却した後，鍵の情報を含む耐タンパメモリ(Tamper-proof Memory)を鍵入力に接続することで，正しい鍵が回路に適用される．このプロセスを活性化と呼ぶ．

従来の論理暗号化手法として，キーゲートであるEXOR/EXNORゲートをアルゴリズムに基づいて挿入する手法がある[1][2][3]．しかしながら，従来の論理暗号化手法はSAT攻撃によって容易に正しい鍵が特定されることが報告されている[4]．SAT攻撃に耐性のある論理暗号化手法としてTTLock[5]と呼ばれる手法が提案されている．TTLockはMLC(Modified Logic Circuit：修正済み論理回路)と比較器から構成されている．MLCはある1つの入力値の時のみ出力値を反転させた論理回路である．MLCの生成では，論理回路から鍵入力数と等しい極大ファンアウトフリーコーンを提案し，そのコーンの真理値表を求めて修正する一連の操作が必要である．しかしながら，この一連の操作が実際には困難である．

それゆえ，本論文では，レジスタ転送レベル(Register Transfer Level：RTL)においてTTLockを容易に実装し，SAT攻撃に耐性のある，面積オーバヘッドをできる限り抑制した論理暗号化手法を提案する．



(a)オリジナルのネットリスト



(b)論理暗号化が施されたネットリスト

図1．従来の論理暗号化手法

第2章では論理暗号化とSAT攻撃について説明し，第3章ではSAT攻撃への対策手法について説明する．そして第4章で提案手法を説明し，第5章で実験結果，第6章でまとめを述べる．

2. 論理暗号化とSAT攻撃

2.1 論理暗号化

論理暗号化とは元の論理回路に鍵入力と呼ばれる外部入力を追加し，鍵入力に正しい値が設定された場合にのみ正しい動作を行うように回路を変更する手法である．論理暗号化が施された回路には新たに鍵入力が追加され，正しい鍵が挿入された時に正しい動作するように設計される．従来の論理暗号化手法はそれぞれのアルゴリズムに基づきキーゲートと呼ばれるEXOR/EXNORゲートを元のネットリストに追加する[1][2][3]．図1(a)にオリジナルのネットリスト，図1(b)に論理暗号化を施したネットリストを示す．鍵入力には正しい鍵の情報を持った読み書き不可能な耐タンパメモリが接続される．EXORゲートの正しい鍵を’0’，EXNORゲートの鍵入力を’1’にすることで暗号化された

Logic Locking Based on Anti-SAT at Register Transfer Level

Atsuya TSUJIKAWA, Toshinori HOSOKAWA and Masayoshi YOSHIMURA

```
Input  : 論理暗号化済みゲートレベルのネットリストC
Input  : オラクルV
Output : 正しい鍵K

1. x0←適当な入力値;
2. y0← V(x0);
3. F←C(x0, k, y0);
4. while True do
5.     k1←SAT_solver(F(k1));
6.     xi←SAT_solver(F(k2)∧C(xi, k1, y1)∧C(xi, k2, y2)∧(y1≠y2));
7.     if Don't find xi then
8.         return k1;
9.     else
10.        yi←V(xi);
11.        F←F∧C(xi, k, yi);
12.    end
13. end
```

図2. SAT攻撃アルゴリズム

回路の動作を元の回路と等価にすることが可能である．EXORゲートに'1', EXNORゲートに'0'を入力すると，出力はEXOR(EXNOR)ゲートのサイドインプットの値を反転させた値を出力する．

2.2 SAT攻撃

近年，論理暗号化の安全性はSAT攻撃によって脅かされている[4]．SAT攻撃は，SATソルバを使用して鍵探索空間を削除することで正しい鍵を求める攻撃アルゴリズムである．SAT攻撃の攻撃者は，論理暗号化された回路の正しい鍵を入手することが目的であり，ファウンドリであると仮定する．攻撃者は，論理暗号化された論理回路レベルのネットリスト，オラクルと呼ばれる活性化済みのVLSIを有する．

次にSAT攻撃のアルゴリズムについて説明する．図2はSAT攻撃のアルゴリズムである．SAT攻撃の入力は論理暗号化済みのネットリストCとオラクルVであり，出力は正しい鍵Kである．まず初めに識別入力x_0を適当に求め，Vを用いて識別入力x_0に対する正しい出力y_0を求める．次に論理暗号化済みのネットリストCをCNF式に変換した後，x_0，y_0に対応する変数に実際の値を代入し，鍵入力条件Fに追加する．次に$F(k_1)$を満たすk_1と$F(k_2) \wedge C(x_i, k_1, y_1) \wedge C(x_i, k_2, y_2) \wedge (y_1 \neq y_2)$を満たす$x_i$をSATソルバを用いて解く．2つ目のSAT問題を解いた際にUNSATとなり，$F(k_2) \wedge C(x_i, k_1, y_1) \wedge C(x_i, k_2, y_2) \wedge (y_1 \neq y_2)$を満たす$x_i$を求めることができなかった場合は，1回目のSAT問題で求まったk_1を正しい鍵Kとして出力する．2つ目のSAT問題を解いた際にSATとなり，$F(k_2) \wedge C(x_i, k_1, y_1) \wedge C(x_i, k_2, y_2) \wedge (y_1 \neq y_2)$を満たす$x_i$を求めることができた場合は，オラクル$V$を使用して，識別入力$x_i$に対する正しい出力$y_i$を求める．最後に$x_i$，$y_i$に対応する変数に実際の値を代入し，鍵入力条件$F$に追加する．求まった識別入力の入出力応答を鍵入力条件Fに追加していくことで，鍵探索空間の正しい鍵の候補を削除していき，最後UNSATになり求まった識別入力に対する出力をする鍵の候補k_1，k_2を求めることができない場合，それはk_1, k_2に同じ鍵が当てはまってしまうということである．そのためUNSATになる前に解いたSAT問題$F(k_1)$の結果k_1が正しい鍵となる．

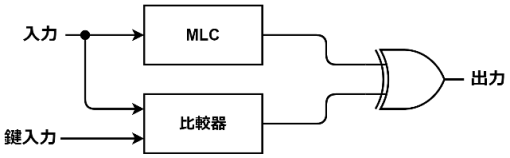


図3. TTLockアーキテクチャ

表1. MLCの生成

(a)元の真理値表

入力	出力
000	0
001	1
010	1
011	0
100	0
101	1
110	0
111	1

(b)MLCの真理値表

入力	出力
000	0
001	1
010	1
011	0
100	0
101	1
110	0
111	1

3. SAT攻撃への対策手法

SAT攻撃へ耐性がある論理暗号化手法としてTTLockと呼ばれる手法が提案されている[5]．TTLockは1つの識別入力により削除される鍵入力を削減することに基づいている．ある入力に対して，誤った出力となる誤った鍵入力を1つにしている．これによって，SAT攻撃の繰り返し回数を増加させ，SAT攻撃への耐性を高めている．図3にTTLockのアーキテクチャを示す．TTLockはMLCと比較器の2つから構成されている．

表1(a)にオリジナルの論理回路の真理値表，表1(b)にオリジナルの論理回路をもとに生成したMLCの真理値表を示す．MLCはオリジナルの論理回路をもとにある1つの入力値に対する出力値のみを反転させた論理回路である．表1(a)の入力101の時の出力は1である．表1(b)のMLCでは入力101の出力を反転させているため，入力101の出力は0となる．TTLockでの正しい鍵は出力を反転させたMLCの入力となっており，表1の場合正しい鍵は101となる．

比較器の入力はMLCへの入力と鍵入力の2つで比較器からの出力はMLCの出力とEXORゲートで接続される．比較器ではMLCへの入

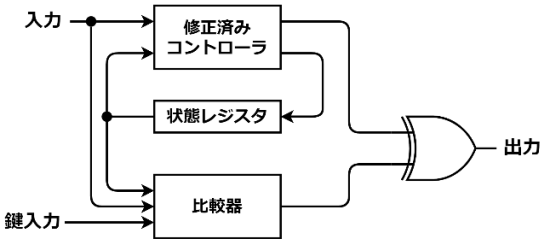
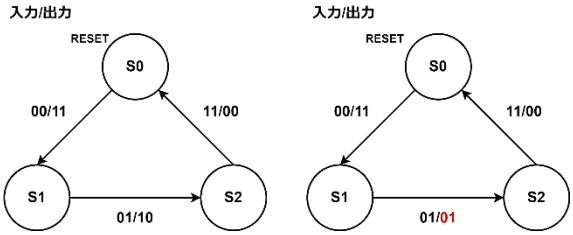


図4. 提案手法アーキテクチャ



(a) (b)

図5. 修正済みコントローラの設計

力と鍵入力が一致しているかを判定し, 一致している場合は'1', 一致していない場合は'0'を出力する. 比較器の出力はEXORゲートでMLCの出力と接続されているため, MLCへの入力と鍵入力が一致した場合は, MLCの出力を反転させて伝搬し, MLCへの入力と鍵入力が一致しない場合は, MLCの出力をそのまま伝搬する.

4. 提案手法

従来の論理暗号化手法[1, 2, 3]では, SAT攻撃によって容易に正しい鍵を解読されてしまう. SAT攻撃に耐性のある論理暗号化手法としてTTLockと呼ばれる手法が提案されたが, TTLockは論理回路レベルでのMLCの生成が必要である. MLCの生成では, 論理回路から鍵入力数と等しい極大ファンアウトフリーコーンを提案し, そのコーンの真理値表を求めて修正する一連の操作が必要である. しかしながら, この一連の操作が実際には困難である. よって本論文ではRTLでTTLockを施すことで容易にMLCの生成を行い, SAT攻撃に耐性のある論理暗号化手法の提案を行う.

本手法はRTLでの論理暗号化手法であり, RTLはコントローラとデータパスの2つから構成されているとする. コントローラからデータパスへは制御信号線, データパスからコントローラへは状態信号線で接続されている. コントローラは現状態と状態信号に基づいて, 次状態と制御信号を出力する.

TTLockはMLCと比較器から構成されており, 本手法ではMLCの設計をコントローラで行う. 図4は本手法のアーキテクチャである. 本手法は修正済みコントローラと比較器の2つから構成されている.

修正済みコントローラの組合せ回路は, ある1つ状態, 状態信号の時の出力を反転させたものである. 図5(a)はオリジナルのコントローラ, 図5(b)は修正済みのコントローラである. TTLockでは, ある1つの入力に対する出力を反転させたMLCを生成した. 修正済みコントローラはMLCをコントローラで実装したものである. 図5の修正済みコントローラの生成の例では状態S1において入力が01の時の出力を反転させている. オリジナルのコントローラでは状態S1における入力01の出力は10であるが, 修正済みコントローラでは出力を10から01に反転させている.

比較器の入力はTTLockではMLCへの入力と鍵入力である. 本手法での比較器への入力は修正済みコントローラへの入力, 修正済みコントローラの現在状態, 鍵入力である. 比較器では鍵入力が, 修正済みコントローラの現状態と入力と一致しているかを判定し, 一致している場合は'1', 一致していない場合は'0'を出力する. 比較器の出力はEXORゲートで修正済みコントローラへ接続される. そのため修正済みコントローラでの現状態, 入力と鍵入力が一致した場合は, 修正済みコントローラの出力を反転させて伝搬し, 修正済みコントローラでの現状態, 入力が鍵入力と一致しない場合は, 修正済みコントローラの出力をそのまま伝搬する.

SAT攻撃への耐性を高めるために, 鍵入力数は設計者が決定することができるべきである. しかしながら, コントローラの入力数, 状態レジスタ数は決まっており, 鍵入力数はコントローラの入力数, 状態レジスタ数に依存してしまう. それゆえ, 本手法では, コントローラ拡大を用いてコントローラの状態レジスタ数を増やすことで鍵入力数の増加を行う. コントローラ拡大とは, 有限状態機械であるコントローラに任意の状態及び状態遷移の追加を行う手法である[7]. 本手法では文献[6]のコントローラ拡大手法を用いている. 文献[6]の手法で状態レジスタ数を増加させることで鍵入力数を設計者が決定できるようにすることで, SAT攻撃への耐性をより高めることができる.

5. 実験結果

本論文では, 提案手法に対してSAT攻撃と面積オーバヘッドの評価を行った. 使用した回路はMCNC'91_RTLベンチマーク回路と自作の

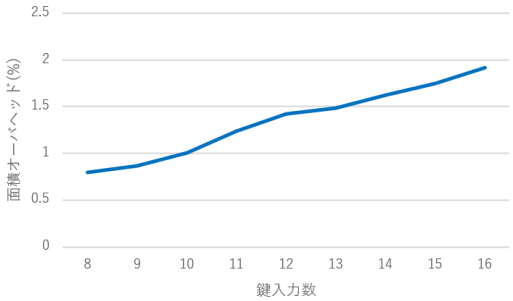


図6. 面積オーバヘッド

32ビットデータパスを組合せ, 論理回路レベルへ論理合成した回路である. 論理合成及び面積評価はsynopsys社のDesign Compilerを用いた. SAT攻撃中のSATソルバはLingeling[8]を用いた.

図6はMCNC'91ベンチマーク回路のdk15を提案手法で論理暗号化を施した面積オーバヘッドである. 鍵入力が8ビットの時の面積オーバヘッドが0.79%, 16ビットの時の面積オーバヘッドが1.91%と鍵入力数を増やしても面積オーバヘッドが非常に小さいことがわかる.

表2はSAT攻撃の実験結果である. 対象の回路において正しい鍵をランダムに生成し, 各5回SAT攻撃を行った. 本実験での鍵入力数は8ビットである. SAT攻撃への耐性の指標として, 攻撃回数が2^{n-1}以上(n: 鍵入力数)の場合SAT攻撃へ耐性があるといえる. 実験結果を見るとほとんどの回路においてSAT攻撃への耐性があるといえる. しかしながら, 一部の回路においては少ない攻撃回数で正しい鍵を解読できている. 少ない攻撃回数で正しい鍵を解読できている回路は小規模回路のためであり, 出力数が少なくTTLockの機能を完全に再現することができなかったためである.

6. まとめ

本論文では, RTLにおけるアンチSATに基づく論理暗号化法を提案した. TTLockではSAT攻撃に耐性はあるが, TTLockではMLCの生成が必要であり, MLCは論理回路から鍵入力数と等しい極大ファンアウトフリーコーンを提案し, そのコーンの真理値表を求めて修正する一連の操作が必要である. そのため, 論理回路レベルでTTLockを実装するのは困難である. そのため, 本手法ではRTLでコントローラを対象にTTLockを実装することで, SAT攻撃に耐性を示し, 面積オーバヘッドを最小限に抑制することができた. 今後の課題として, 鍵入力数の増加が挙げられる.

参考文献

[1] J. A. Roy, F. Koushanfar, and I. L. Markov, "EPIC: EndingPiracy of Integrated Circuits," in Design, Automation and Test in Europe, DATE 2008, Munich, Germany, March10-14, 2008.IEEE, 2008, pp. 1069–1074.

[2] J. Rajendran, H. Zhang, C. Zhang, G. S. Rose, Y. Pino,O. Sinanoglu, and R. Karri, "Fault Analysis-Based Logic Encryption,"IEEE Transactions on Computers, 64(2), Feb 2015.

[3] M. Yasin, J. J. V. Rajendran, O. Sinanoglu, and R. Karri, "On Improving the Security of Logic Locking," IEEE Trans. on CAD of Integrated Circuits and Systems, pp. 1411–1424, 2016.

[4] P. Subramanyan, S. Ray, and S. Malik, "Evaluating the security of logic encryption algorithms," 2015 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), pp.137-143, May 2015.

[5] M. Yasin, A. Sengupta, B. C. Schafer, Y. Makris, O. Sinanoglu, and J. V. Rajendran. "What to lock? : Functional and parametric locking, " In Proceedings of the on Great Lakes Symposium on VLSI 2017.

[6] 辻川 敦也, 細川 利典, 吉村 正義, "機能等価な有限状態機械生成に基づく面積削減指向コントローラ拡大法," 信学技報, DC2020-15, pp.93-98, Jul 2020.

[7] Yuta Ishiyama, Toshinori Hosokawa, Hiroshi Yamazaki, "A Design for Testability Method for k-Cycle Capture Test Generation," IEEE 25thInternational Symposium on On-Line Testing and Robust System Design(IOLTS'19), pp.40-43, Jul 2019.

[8] A. Biere, "Lingeling, plingeling and treengeling," In Proceedings of the SAT completition, 2013.

表2. SAT攻撃

回路	dk15	dk17	ex6	lion	sand	kirkman
平均攻撃回数	245.4	202	240	95.2	177.6	132.4
最大攻撃回数	256	256	256	193	256	256
最小攻撃回数	203	51	221	3	117	15