7-16

RSA暗号回路に対する安全なテスト容易化設計

日大生産工（院）　○早川 鉄平　　日大生産工　細川 利典
九大　吉村 正義

１．はじめに

近年パーソナルコンピュータや携帯電話をはじめとする個人向けの情報機器の普及によって，様々な通信を個人レベルでも利用できるようになった．そのため通信情報の暗号化など，プライバシー保護の面で暗号アルゴリズムが重要な役割を担っている．一般的に，暗号アルゴリズムは高速化や低消費電力化などの理由からソフトウェアでなくハードウェアで実装している[1]．ハードウェアで実装した暗号回路は回路の出力が入力と回路内部のフリップフロップ（Flip-Flop:FF）の値に依存する順序回路で生成される．

順序回路では，FFにより構成される各レジスタに値の設定を行うことや観測を行うことが困難である．それゆえ，高い故障検出率を達成するためのテスト生成が困難である．各FFに値を設定できるスキャンチェインを用い，すべてのFFを制御，観測可能とするフルスキャン設計[2]を適用すればテスト容易となるが，テスト用外部ピンが使用可能である場合，スキャンチェインを利用したサイドチャネル攻撃[3]を受ける危険性が生じる．暗号回路に対しては，正規の通信経路以外から取得できる情報を用いたサイドチャネル攻撃が考えられており，スキャンチェインを利用したサイドチャネル攻撃も問題の一つである．暗号回路では暗号化及び復号化のための秘密鍵，復号計算中の平文が回路の内のレジスタに格納されているため，その安全性が重要な問題となっている．

本稿では，RSA暗号回路[4]でのセキュリティを維持しつつ，フルスキャンテストと同等の高い故障検出率を得るために，パーシャルスキャン設計と制御ポイントを用いた安全なスキャン設計法でRSA暗号回路のテスト容易化を適用し，そのテスタビリティを評価する．パーシャルスキャン設計を適用することにより，一部レジスタをスキャンチェインに含まないことで秘密情報を持つレジスタを保護するテスト容易化設計を行う．さらに，可観測性の悪い演算器出力部にXOR木を挿入し，可制御性の低い演算器に制御ポイントを付加することによりセキュリティを維持しながらテスタビリティの向上を図るテスト容易化設計を提案する．

２．RSA暗号回路とその脆弱性

RSA暗号回路について説明する．公開鍵暗号であるRSA暗号では，暗号化前の文（平文），暗号化後の文（暗号文），暗号化に用いる鍵（公開鍵２つのセット），復号化に用いる鍵（秘密鍵と公開鍵のセット）を数値として扱い，暗号化および復号化では同じ演算処理を行う．暗号化において暗号文yを秘密鍵d，公開鍵mを用いて平文xに復号するとする．そのとき平文xと暗号文yには$x=y^d \bmod m$という関係がある．

べき乗計算を行うとき，yをd回かけると，べき乗計算でのかけ合わせは2^d-1回行われるので，dの値が大きくなると計算に膨大な時間が掛かる．それゆえdを2進数表記し，1となったビットのみでべき乗を繰り返し，y^dを計算する．またy^dの計算が完了してから剰余（mod）を行うのではなく，べき乗途中でも行うようにし，ビット数の増加を防ぐ．例．$V^{137} \bmod m$を計算する場合について考える．137は2進数表記で10001001となるので，ビット7，3，0が1となる．そのため，Vを2^7乗，2^3乗，2^0乗した値それぞれにmod m（2乗剰余算）を実行し，その結果すべてを乗算しさらにmod m（乗剰余算）を実行することで計算することが可能となる．この手法はBinaryMethod[5]といい，本稿のRSA暗号回路にも用いられている．

暗号回路に対する攻撃は，攻撃者の知識によりさまざまなレベルが考えられる．本稿では，一般的に得られる情報

A secure design for testability of RSA encryption circuits

Teppei HAYAKAWA, Toshinori HOSOKAWA, and Masayoshi YOSHIMURA

のみを利用する攻撃者を想定し，攻撃者の知識を以下のように仮定する．

(1) 暗号回路として実装される暗号アルゴリズムを知っている．さらに，実装されるレジスタ転送レベル（Register transfer level:RTL）回路を知っているか，いくつか候補を想定できる．

(2) テスト容易化設計としてスキャン設計が採用されている場合，テスト用外部ピンを認識できる．

(3) ゲートレベル回路の設計情報を知らない．

(4) スキャンチェインでのFFの接続順序を知らない．

上記の仮定の知識のみを攻撃者が持ち，テスト容易化設計の詳細は知らなくても，一般的によく用いられているいくつかの暗号回路の脆弱性が知られている．共通鍵暗号であるDES (Data Encryption Standard)，AES (Advanced Encryption Standard)を安全性を考慮せずフルスキャン設計を用いて実装したとき，暗号回路に対するスキャンチェインを利用したサイドチャネル攻撃が報告されている[6][7]．これらの攻撃法では，攻撃者は複数の入力パターンを用いてスキャンチェイン上でのFFの接続順序を特定し，秘密鍵を得ることに成功している．また本稿で扱うRSA暗号回路の復号アルゴリズムでは，2乗剰余算と乗剰余算が繰り返し実行される．乗剰余算を実行するか否かは秘密鍵のビットパターンに依存するため，乗剰余算の結果に関連した値を格納するレジスタを特定できれば秘密鍵を求めることが可能である[8]．RSA暗号回路の各レジスタは2乗剰余算と乗剰余算の計算周期に応じて特有の値の遷移をもつため，フルスキャン設計の場合，スキャンレジスタの値を定期的に観測することでスキャンチェイン上のFFとレジスタの対応を解析することも可能である．以上の考察により，RSA復号回路はスキャンチェインを利用したサイドチャネル攻撃に対し，脆弱であるといえる[8]．

3．テスト容易化設計

3-1. パーシャルスキャン設計

故障検出率の向上と安全性を両立させるために，秘密情報を持つレジスタ（危険レジスタ）をスキャンチェインに含めず，残りの秘密情報を持たないレジスタをスキャンチェインに含めるパーシャルスキャン設計を考える．

このパーシャルスキャン設計ならばオリジナル回路と同等のセキュリティを維持しつつ，オリジナル回路以上に故障検出率を向上することができる．

3-2. 排他的論理和木

パーシャルスキャン設計を適用した際，危険レジスタはスキャンチェインに含まれない．そのため，乗剰余算部のレジスタ周辺は可観測性が低くなっており，そのため観測困難故障が乗剰余算部内の演算器周辺に集中している．そこで排他的論理和木（XOR木）[9]挿入を利用した観測法を提案する．XOR木は各ビットをそれぞれXOR演算し，その結果が1ビットとなるまでXOR演算を行う回路である．

最終的に求められた1ビットはスキャン化するFF（スキャンFF）によって観測する．

XOR木の入力の奇数箇所に故障が伝搬した場合，最終的に正常値と反転した値を出力するため故障の影響の観測が容易である．また，出力される情報は1ビットで秘密情報の解析が非常に困難であるため，安全性も確保できていると考える．

図1にXOR木の例を示す．なお，例では8ビットのXOR木を使用しているが，実装したものは32ビットのXOR木である．

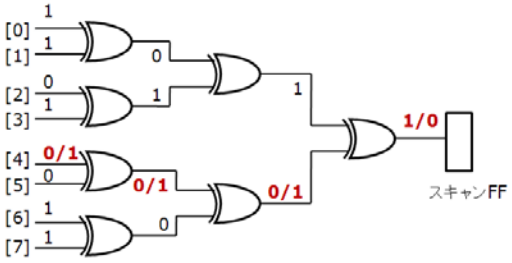

図１．XOR木実装例

3-3. 制御ポイント

パーシャルスキャン設計を適用したRSA暗号回路に残ったフィードバックループ中の演算器は，入力への値が設定しづらく，可制御性が低くなってしまっている．

そこで可制御性の低い演算器の入力の前段に，制御信号が0の場合は定数0または1を，1の場合は本来の値を与える働きをするような制御ポイント[10]を挿入する．これにより減算器であればA-0=Aを，加算器であればA+0=Aおよび1+0＝1を実現し，故障検出率の向上を図る．

図2は0制御ポイントを，図3は1制御ポイントを示す．

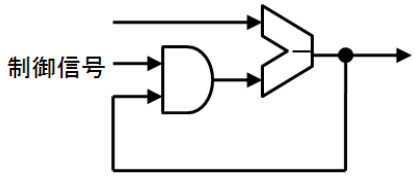

図２．0制御ポイント

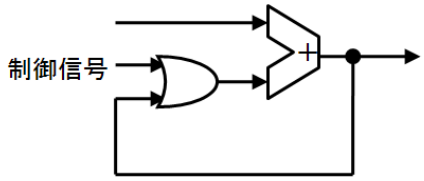

図３．1制御ポイント

４．実験結果

32ビットRSA暗号回路を用い，オリジナル回路，フルスキャン設計を適用した場合，パーシャルスキャン設計を適用した場合，パーシャルスキャン設計にXOR木挿入を適用した場合，また，比較対象として故障伝搬は確実ではないが，XORより面積の小さいNAND木を適用した場合，パーシャルスキャン設計にXOR木と制御ポイント挿入を適用した場合，パーシャルスキャン設計にNAND木と制御ポイント挿入を適用した場合において面積と故障検出率を比較した．

RSA復号回路はオープンソースIPコアとして公開されているRSA回路 [11] を用いた．

RSA復号回路はデータパス部とコントローラ部からなり，データパス部は2乗剰余算部（modsqr）と乗剰余算部（modmult）から構成されている．

RSA復号回路のデータパス部のブロック図を図4に示し，RSA復号回路のコントローラのブロック図を図5に示す．図4のRSA復号回路のデータパス部では，射線の付いたレジスタ，すなわち乗剰余算部の結果に依存するレジスタ(cypher，tempin，modmult内のprodreg，mcreg)が危険レジスタとなり，図5のRSA復号回路のコントローラでは斜線のついたレジスタ，すなわち秘密鍵に依存するレジスタ（count）が危険レジスタとなる．

パーシャルスキャン設計に関しては，こちらの危険レジスタをスキャンチェインに含めず，それ以外のレジスタをスキャンレジスタとした．スキャンチェインに含まれたレジスタは，2乗剰余算部(modsqr)内のmpreg，mcreg，modreg1，modreg2，prodregと，乗剰余算部（modmult）内のmpreg，modreg1，modreg2とデータパス内のroot，modreg, sqrin, コントローラ内のmultgo, doneであった．

評価実験は論理合成ツールDesign Compiler（Synopsyp社）を用いて面積を，自動テスト生成ソフトTetraMAX（Synopsys社）を用いてスキャン化率，故障検出率を求めた．それぞれの場合についてそれらの評価項目を比較した．

表１に面積を示す．“Originl”はオリジナル回路の場合

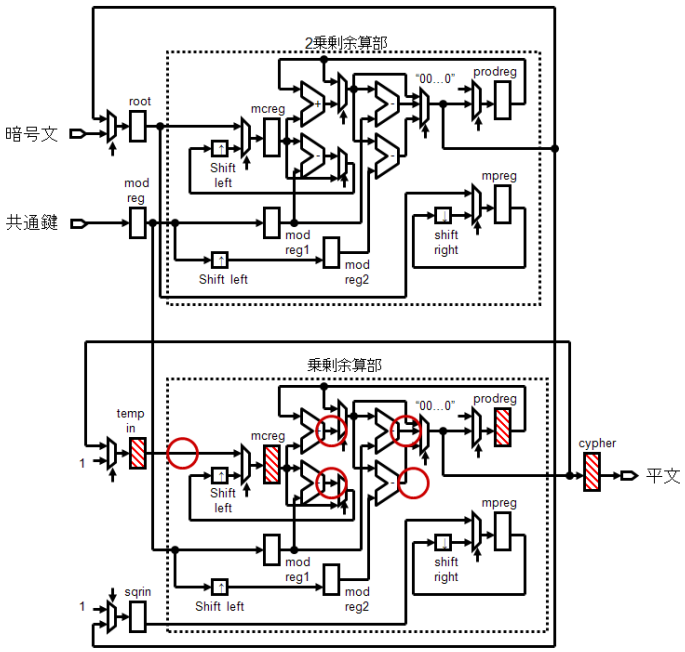

図４．RSA暗号回路データパス部ブロック図

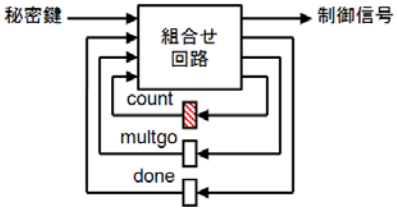

図５．RSA暗号回路コントローラ部ブロック図

の評価を，“FS”はフルスキャン設計を適用した場合の評価を，“PS”はパーシャルスキャン設計を適用した場合の評価を，“XOR”はパーシャルスキャン設計にXOR木挿入を適用した場合の評価を，“NAND”はパーシャルスキャン設計にNAND木挿入を適用した場合の評価を，“XOR+CP”はパーシャルスキャン設計にXOR木と制御ポイント挿入を適用した場合の評価を，“NAND+CP”はパーシャルスキャン設計にNAND木と制御ポイント挿入を適用した場合の評価を表す．

面積はNOTゲートを2として計算した値である．表１より，“PS”では，“FS”と比べ，面積オーバーヘッドが抑えられていることがわかる．これは，スキャンFFが回避した危険レジスタ分だけ面積オーバーヘッドが少ないことに起因する．“XOR”及び“NAND”ではXOR木またはNAND木と観測用FFを付加しているため，面積が若干大きくなっている．

“XOR+CP”と“NAND+CP”の面積オーバーヘッドは，“PS”と比べ，更に制御ポイントを追加した大きさ分

だけ増加しているが，“FS”に比べて小さいことがわかる．

表１．面積オーバーヘッド

	面積	スキャンFF	スキャン化率
Original	10318	0	0.00%
FS	12626	577	100.00%
PS	11990	418	72.44%
NAND	12145	425	73.02%
XOR	12455	425	73.02%
NAND+CP	12247	425	73.02%
XOR+CP	12622	425	73.02%

表２にテスト生成結果を示す．故障モデルは単一縮退故障モデルである．「全故障」，「検出」，「テスト不可能」，「打ち切り」，「CPU」はそれぞれ，全故障数，検出故障数，テスト不可能故障数，テスト生成アルゴリズムでバックトラックの制限（1000回）により処理を打ち切られた故障数，テスト生成にかかったCPU時間を表す．

表２より，“PS”では，“FS”に比べ，検出故障数が少なくなっていることがわかる．これは回路中にフィードバックループが残り，テスト生成アルゴリズムで処理が打ち切られた故障数が増加したためである．

“NAND”および“XOR”の場合，故障検出率が向上している．これは観測ができないため打ち切られていた故障が観測可能となったことによる．

“XOR”の場合，“NAND”とは異なり，単一故障の場合確実に故障が検出可能なため，より良い結果となった．

“XOR+CP”，“NAND+CP”の場合，故障検出率がさらに向上している．これは制御不可能により打ち切られた故障が制御ポイントの付加により減少したことによる．

５．おわりに

本稿では，パーシャルスキャンと入力0制御機能，XOR木を利用した安全なスキャン設計法を提案し，評価した．スキャンチェインを利用したサイドチャネル攻撃に対し脆弱であると考えられる危険レジスタをスキャンチェインに含まないことで被テスト回路にテスト容易化設計を適用しない場合と同等の安全性を実現し，パーシャルスキャン設計にXOR木と制御ポイントを付加した場合，フルスキャン設計を適用した場合より少ない面積オーバーヘッドで，フルスキャン設計を適用した場合と同等の故障検出率を得ることができた．

「参考文献」

1）盛岡澄夫，HDLによる高性能ディジタル回路設計，CQ出版社，2002年

2）藤原秀雄，ディジタルシステムの設計とテスト，工学図書株式会社，2004年，pp.202-213

3）佐藤証，高橋芳夫，森岡澄夫，LSIを盗聴から守る―暗号回路のサイドチャネル攻撃とその対策―，デザインウェーブマガジン，2月号，2006年，pp.100-134

4）結城浩，暗号技術入門　秘密の国のアリス，ソフトバンクパブリッシング，2004年，pp.113-138

5）森岡澄夫，アルゴリズムのハードウェア化手法，デザインウェーブマガジン，1月号，2008年，pp.20-70

6）B.Yang, K.wu, and R.Karri , Secure scan: A design-for-test architecture for crypto chips, IEEE Transactions on Computer -Aided Design of Intergrated Circuit and Systems, 2006年10月， pp.2287-2293

7）B.Yang, K.wu, and R.Karri, Scan based side channel attack on dedicated hard ware implementations o data encryption standard, Proceedings of International Test Conference 2004 (ITC'04), 2004年，pp.339-344

8）長谷川宗士,井上美智子,藤原秀雄，平衡構造を利用した安全なスキャン設計，DC研究会，2008年2月，pp. 39-44

9) H.Fujiwara and A.Yamamoto,Parity-scan design to reduce the cost of test application,IEEE Transaction on Computer-Aided-Design, 1993,Vol.12,pp.1604-1611

10）C.Schotten and H.Meyr，Test Point insertion for an area efficient bist.Proc, Of the IEEE Int. Test Conf. 1995, 1995年，pp. 515-523

11）OPEMCORES, RSA processor, http://www.opencores.org/projects.cgi/web/rsa/overview

表２．テスト生成結果

	全故障	検出	テスト不可能	打ち切り	故障検出率	CPU
Original	30356	122	1498	27638	3.03%	1623.96
FS	32658	32637	17	4	99.98%	0.22
PS	32020	28120	313	3587	87.17%	203.91
NAND	32458	30820	106	1532	94.26%	92.74
XOR	32458	31878	53	527	97.76%	51.07
NAND+CP	33056	32303	112	641	97.32%	57.43
XOR+CP	33120	33004	51	65	99.26%	26.53