C-013

実装方式の違いによるフォールト攻撃に対する耐性評価
Evaluation of fault injection attack for several implementations

後藤 輝 †　　塚平 峻矢 †　　吉川 雅弥 †
Hikaru Goto　　Takaya Tsukadaira　　Masaya Yoshikawa

1. はじめに

現在，クレジットカードやキャッシュカードなどに含まれている秘密情報は，AES などの暗号アルゴリズムによって保護されている．しかし近年，暗号処理中に故意に故障を起こし，正しい暗号文と誤った暗号文の組を用いて不正に秘密情報を取得するフォールト攻撃（サイドチャネル攻撃の一種）が大きな脅威となってきた．

暗号回路をハードウェアに実装する場合，その実装方法によって回路の構成が異なる．これらの違いはサイドチャネル攻撃に対する耐性と関係があり，先行研究では ASIC を対象にした評価実験により，耐性が回路の構成に依存することが報告されている[1]．そこで本研究では，標準暗号 AES を対象に，様々な FPGA の実装方式について，フォールト攻撃に対する耐性を評価する．

2. フォールト攻撃（故障利用解析）

2.1 フォールト攻撃とは

フォールト攻撃とは，暗号処理中に故意に故障を起こす事で，暗号化に用いられた元鍵を導出する解析手法である．ここでの故障とは，暗号処理中の中間値を強制的に変化させることを示す．故障を起こした事によって変化した中間値はその後の処理に影響を及ぼし，最後に出力される暗号文の値を正規の値から変化させる．この故障によって変化した暗号文と正規の暗号文を基に，元鍵の解析を行う．

2.2 故障発生方法

故障を発生させる方法は複数存在し，レーザ照射を用いる方法，電源電圧を降下させる方法などが挙げられる．しかし，前者は高価な装置を用いることからコスト面で有効ではなく，後者は回路を破壊する危険性があるため，有効とは言えない．そこで本研究では，グリッチを用いた故障発生方法を適用する．この方法は，LSI の動作クロックに通常よりも短い異常パルス（グリッチ）を用いる事によって，故障を発生させる手法であり，回路を破壊することがなく，低コストで行える．

2.3 グリッチによる故障発生原理

本研究で用いたグリッチによる故障発生原理を説明する．まず，異常パルスを特定のタイミングで混入させる．すると，暗号デバイス内の信号状態がグリッチの立ち上がりによって変化する際，短い周期のため，クロックのエッジの遷移前にデータ信号が安定していなければならない最少時間（セットアップタイム）を確保できず，セットアップタイム違反が発生し，演算誤りを引き起こす．以上がグリッチによる故障発生のメカニズムである．図 1 に，グリッチによるセットアップタイム違反を示す．

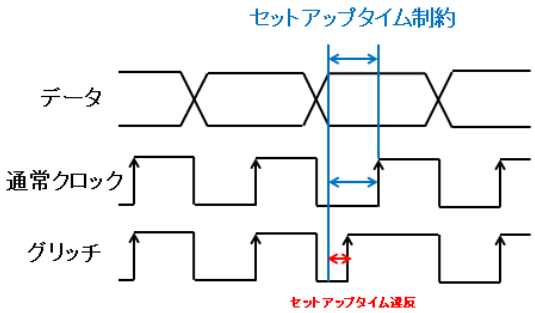

図 1　グリッチによるセットアップタイム違反

3. 実装方式・実装方法

3.1 実装方式

代表的な AES の実装方式として合成体方式と，テーブル方式が挙げられる．この 2 つの実装方式の違いは SubBytes 処理が行われる S-Box 回路の構成にある．合成体方式は，逆元演算回路に，アフィン変換回路を直列に繋ぐことで，S-Box 回路を実現する．テーブル方式は，逆元演算処理とアフィン変換処理の結果を，真理値表で実現することで S-Box 回路を実現する．

3.2 実装方法

本実験では，Verilog で記述した合成体方式の AES と，テーブル方式の AES をサイドチャネル攻撃用標準評価ボード SASEBO-G Ⅱ へ FPGA 実装する．合成体方式は構造上，ロジックセルのみで構成される．一方，テーブル方式は，ロジックセルのみの構成に加え，FPGA の内部リソースである BlockRAM を使用して構成することも可能である．BlockRAM を使用する場合は，実装ツールによる ROM 推論と，ROM モジュールを手動で生成し，BlockRAM として実装の 2 つの方法が存在する．尚，FPGA 実装した場合，回路規模は ROM 推論を使用した場合が最も小さく，ROM 推論を無効にした場合が最も大きくなる．

4. 評価実験

本研究では，ロジックセルのみで構成した合成体方式に加え，ロジックセルのみで構成したテーブル方式，ROM 推論を適用したテーブル方式，手動で ROM モジュールを生成し，BlockRAM として実装したテーブル方式の 4 つの実装方式に対して，2 章で述べたグリッチによるフォールト攻撃に対する耐性評価を行った．尚，今回実装した 4 つの AES はすべて暗号化処理しか行っておらず，復号化処理は行っていない．

† 名城大学理工学研究科情報工学専攻
Department of Information Engineering, Meijo University

4.1 実験環境

実験環境を表 1 に示す．SASEBO-Checker[2]とは PC 上で動作する SASEBO ボードと通信するソフトウェアで，乱数で平文（16 バイト）を生成した後 SASEBO ボードに伝送し，SASEBO ボードで処理を終え出力された暗号文（16 バイト）を受け取る．また，trace を変更することで試行回数を変えることができ，trace を 100 と設定した場合，上記の動作を 100 回繰り返して行うことができる．今回使用する SASEBO-Checker は，グリッチによるフォールト攻撃実験用に改良を加え，使用した平文，出力された暗号文，故障が発生した場合の暗号文，試行回数の内の故障が発生した回数をテキストファイルで出力できるようにしている

表 1　実験環境

FPGA 実装ツール	Xilinx ISE Design Suite 12.1
暗号回路用デバイス	Virtex-5 XC5VLX30
制御回路用デバイス	Spartan-3A XC3S400A
オシロスコープ	Agilen TechnologiesDSO1024A
プローブ	Agilent N2863A 10.1 PROBE
電源	PC から USB による供給
ソフトウェア	SASEBO-Checker(C#)

4.2 評価方法

4.1 で述べた実験環境で，実装方式が異なる 4 つの AES に対して，グリッチによるフォールト攻撃（試行回数 1000 回）を行う．その際，SASEBO-Checker を用いて 1000 回の内の故障が発生した回数を測定する．フォールト攻撃は，故障が混入した暗号文が必要となるため，故障が発生しなかった場合は元鍵の解析を行うことができず，フォールト攻撃に対する耐性が高いと言える．よって，今回の実験では，故障が発生した回数に着目する．また，グリッチ周期 ***Tg*** によってはセットアップタイム違反を起こすタイミングがずれて故障数が変化する，あるいは違反が起きない場合があるので，グリッチ周期を変更した場合の実験も行う．

4.3 実験結果

表 2 に実験結果を示す．「COMP」は合成体方式，「NONROM」はロジックセルのみで構成したテーブル方式，「TBL」は ROM 推論を適用したテーブル方式，「ALLROM」は手動で ROM モジュールを生成し，BlockRAM として実装したテーブル方式を示している．

表 2 を見ると，合成体方式はテーブル方式に比べ，故障発生数が多く，また，故障が発生するグリッチ周期の範囲が広いことが分かる．これは，合成体方式の AES の S-Box 処理で行われる逆元演算の計算量が多く，計算時間が長いことにより，セットアップタイム違反を引き起こせる範囲が広がったためであると考えられる．それに対し，テーブル方式は，故障発生数が少なく，故障を起こす事ができるグリッチ周期の範囲も狭い事が分かる．これは，テーブル方式の AES の S-Box 処理はテーブルを参照して値を置き換える処理のため，逆元演算を行う合成体方式の AES に比べ，計算量が少なく，計算時間が短いためであると考えられる．TBL，ALLROM に関しては，グリッチ周期が 5.0ns，5.2ns の場合で故障の発生を確認できた．この故障によって出力された暗号文を用いて，元鍵を導出する実験は本研究では行っていないが，解析できる可能性がある．しかし，故障を起こす事ができるグリッチ周期の範囲は合成体方式に比べて狭く，グリッチを発生させるタイミングが難しいことが分かった．NONROM に関してはいずれのグリッチ幅でも故障を確認することができなかったため，フォールト攻撃に対する耐性が高いということが明らかになった．

尚，グリッチ周期が 4.6ns 未満の場合と，6.0ns 超過の場合については，いずれも故障が発生しなかったので省いている．6.0ns 超過については，グリッチ周期が長く，セットアップタイムを確保できたため，故障が発生しなかったと考えられる．4.6ns 未満に関しては，本研究で採用したグリッチ生成方法によるもので，グリッチを発生させることができなかったためである．図 2 に，オシロスコープで測定した 5.4ns 時のクロック波形(左図)と，4.4ns 時のクロック波形(右図)を示す．①は立ち上がり閾値，②は立ち下がり閾値である．4.4ns 時はグリッチが立ち下がり閾値を超えておらず，故障誘発に失敗していることが分かる．

表 2　フォールト攻撃による故障発生数

Tg	COMP	NONROM	TBL	ALLROM
4.6ns	**0**	**0**	**0**	**0**
4.8ns	**9**	**0**	**0**	**0**
5.0ns	**202**	**0**	**6**	**5**
5.2ns	**565**	**0**	**1**	**1**
5.4ns	**80**	**0**	**0**	**0**
5.6ns	**14**	**0**	**0**	**0**
5.8ns	**2**	**0**	**0**	**0**

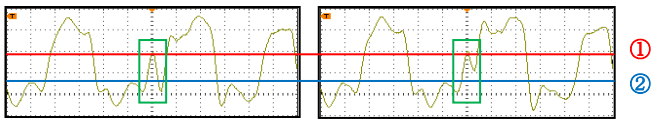

図 2　オシロスコープで測定したクロック波形
（四角で囲った部分がグリッチ波形）

5. まとめ

本研究では，実装方式・方法が異なる 4 つの AES に対して，グリッチによるフォールト攻撃解析を行い，故障発生数を測定する事で，フォールト攻撃に対する耐性を評価した．その結果，故障発生数は実装方式によって変化し，フォールト攻撃に対する耐性は，実装方式に依存することが明らかになった．

今後の課題は，取得した故障入りの暗号文を用いて，元鍵の解析を行い，さらに細かい耐性評価をしていく予定である．

謝辞

本研究は JST,CREST「ディペンダブル VLSI システム基盤技術」の研究の一環として行われた．

参考文献

[1] 森岡澄夫, 佐藤証, “共通鍵暗号 AES の低消費電力論理回路構成法,” 情報処理学会論文誌, vol.44, No.5, pp.1321-1328, 2003.
[2] SAKURA Hardware Security Project, http://www.morita-tech.co.jp/SAKURA/en/tools.html