

目 次

第 1 章 序論	1
1.1 緒言	1
1.2 暗号回路に対する物理解析	4
1.3 本研究の目的と論文構成	5
参考文献	8
第 2 章 軽量暗号	13
2.1 緒言	13
2.2 暗号技術の概要	13
2.2.1 暗号技術の分類	13
共通鍵暗号	13
公開鍵暗号	15
2.2.2 暗号実装	15
ハードウェア実装	16
ソフトウェア実装	16
2.3 AES 暗号	16
2.4 軽量暗号の概要	18
2.4.1 各種軽量暗号アルゴリズム	19
TWINE	19
SIMECK	19
Midori	19
PRINCE	20
Minalpher	20

TWINE-OTR	21
2.5 結言	21
参考文献	21
第3章 サイドチャネル解析	25
3.1 緒言	25
3.2 暗号回路に対する物理解析	25
3.2.1 侵襲解析	25
3.2.2 半侵襲解析	26
3.2.3 非侵襲解析	26
3.3 電力・電磁波解析	27
3.3.1 CPA	27
3.3.2 FB-CPA	29
3.3.3 BS-CPA	30
3.4 フォールト解析	30
3.4.1 解析手法に関する研究	31
3.4.2 フォールト混入手法に関する研究	31
3.5 結言	32
参考文献	32
第4章 TWINE に対する電力解析手法	35
4.1 緒言	35
4.2 提案する電力解析手法	36
4.2.1 ベースとなる時間領域での電力解析手法	36
4.2.2 周波数領域での電力解析手法	37
4.2.3 複数ラウンドに対する解析手法	40
4.3 評価実験	43
4.3.1 実験環境	43
4.3.2 実験結果	44
4.3.3 解析に優位な周波数帯域の検証	46
4.3.4 周波数領域での解析効率の検証	47

4.3.5 部分鍵の組み込み処理の解析効率への影響	48
4.4 結言	50
参考文献	50
第5章 SIMECK に対する電磁波解析手法	52
5.1 緒言	52
5.2 提案する電磁波解析手法	52
5.2.1 ベース電磁波解析手法	53
5.2.2 複数ラウンドを利用した解析	55
5.2.3 既知情報の組み込み処理	57
5.3 評価実験	60
5.3.1 実験環境	60
5.3.2 実験結果	62
5.3.3 複数ラウンドを利用した解析の有効性評価	64
5.3.4 既知情報の組み込み処理の有効性評価	66
5.4 結言	67
参考文献	67
第6章 Midori に対する電力解析手法	68
6.1 緒言	68
6.2 提案する電力解析手法	68
6.2.1 階層的な電力解析手法	69
6.2.2 1段階目の電力解析手法	69
6.2.3 2段階目の電力解析手法	70
6.2.4 解析の効率化手法	73
6.3 評価実験	76
6.3.1 実験環境	76
6.3.2 実験結果	78
6.3.3 相関係数の比較	79
6.3.4 処理時間・計算量の比較	81
6.4 結言	81

参考文献	82
第 7 章 Minalpher に対する電力解析手法	83
7.1 緒言	83
7.2 提案する電力解析手法	83
7.2.1 2 段階での電力解析手法	84
7.2.2 1 段階目の電力解析手法	84
7.2.3 2 段階目の電力解析手法	86
7.2.4 2 段階での電力解析の周波数領域での適用方法	88
7.3 評価実験	90
7.3.1 実験環境	90
7.3.2 実験結果	90
7.3.3 周波数領域での電力解析の有効性評価	94
7.3.4 相関係数の比較	94
7.3.5 使用する周波数帯域が解析に与える影響についての評価	96
7.4 結言	98
参考文献	98
第 8 章 TWINE-OTR に対する電力解析手法	100
8.1 緒言	100
8.2 提案する電力解析手法	100
8.2.1 2 段階電力解析の概要	101
8.2.2 1 段階目の電力解析手法	101
8.2.3 2 段階目の電力解析手法	103
8.2.4 複数の暗号文生成部分を利用した解析	105
8.3 評価実験	110
8.3.1 実験環境	110
8.3.2 実験結果	112
8.3.3 複数の暗号文生成部分を利用した解析の有効性評価	112
8.3.4 相関係数の比較	113
8.4 結言	113

参考文献	115
第 9 章 TWINE に対するフォールト解析手法	116
9.1 緒言	116
9.2 提案するフォールト解析手法	116
9.2.1 TWINE の信号伝搬時間の偏り	117
9.2.2 フォールトモデル	118
9.2.3 統計的な解析手法	119
9.3 評価実験	122
9.3.1 実験環境	122
9.3.2 実験結果	123
9.3.3 ハミング距離の比較	125
9.4 結言	126
参考文献	126
第 10 章 SIMECK に対するフォールト解析手法	128
10.1 緒言	128
10.2 提案するフォールト解析手法	128
10.2.1 フォールトモデル	128
10.2.2 解析手法の概要	130
10.2.3 1 段階目のフォールト解析手法	130
10.2.4 2 段階目のフォールト解析手法	134
10.3 評価実験	135
10.3.1 実験環境	135
10.3.2 実験結果	137
10.3.3 平均ハミング距離の比較	138
10.4 結言	140
参考文献	140
第 11 章 Midori に対するフォールト解析手法	141
11.1 緒言	141
11.2 提案するフォールト解析手法	141

11.2.1	解析手法の概要.....	141
11.2.2	フォールトモデル.....	142
11.2.3	1 段階目のフォールト解析手法.....	145
11.2.4	2 段階目のフォールト解析手法.....	148
11.4	評価実験.....	150
11.4.1	実験環境.....	150
11.4.2	実験結果.....	151
11.4.3	平均ハミング距離の比較.....	153
11.4.4	フォールト混入の有効性検証.....	154
11.5	結言.....	155
参考文献		156
第 12 章 PRINCE に対するフォールト解析手法.....		157
12.1	緒言	157
12.2	フォールトモデル.....	158
12.3	提案するフォールト解析手法.....	161
12.3.1	ベースとなる解析手法の概要.....	161
12.3.2	解析手法における逆算処理.....	163
12.3.3	解析手法における鍵の推定.....	165
12.3.4	階層的な解析モデル.....	166
12.3.5	枝刈リアルゴリズム.....	169
12.4	評価実験	170
12.4.1	実験結果.....	170
12.4.2	ハミング距離の比較結果.....	172
12.4.3	処理時間の比較.....	173
12.5	結言	174
参考文献		175
第 13 章 結論		177
謝辞		180
付録 軽量暗号アルゴリズム.....		182

A.1	TWINE.....	182
A.1.1	暗号処理部.....	183
A.1.2	鍵スケジュール部.....	185
A.1.3	実装方法.....	186
A.2	SIMECK	188
A.3	Midori	189
SubCell.....		190
ShuffleCell.....		190
MixColumn.....		191
KeyAdd.....		192
A.4	PRINCE	192
0R の定数加算処理.....		193
ラウンド関数 F (1R から 5R までの処理)		193
中間処理		196
逆ラウンド関数 F^{-1} (6R から 10R までの処理)		196
11R の定数加算処理.....		197
A.5	Minalpher.....	198
オフセット値生成部分.....		198
暗号文生成部分		199
S 関数		200
T 関数.....		201
M 関数.....		201
E 関数.....		202
A.6	TWINE-OTR	202
オフセット値生成部分.....		203
暗号文生成部分		203
	参考文献	204

第 1 章

序論

1.1 緒言

現在，Internet of Things（IoT）によって，自動車や家電，ビル，工場などを含むあらゆるモノがネットワークに接続されるようになってきた。総務省の平成 30 年度版情報通信白書によると，2020 年には IoT 機器の数は約 400 億個に増加するといわれている [1]。例えば，コンシューマ分野ではスマートウォッチなどのウェアラブルデバイスからスマートテレビなど多岐に渡るデバイスが IoT 化している [2]。農業分野の IoT 化はスマート農業と呼ばれ，気温や日照量，湿度，降水量などをセンサでモニタリングすることで，農作物に与える水や肥料の量を調整し，生産性の向上を図っている [3]。自動車分野では‘コネクティッドカー’をキーワードに車車間や路車間での通信が行われている [4]。工場分野の IoT 化はスマートファクトリなどとも呼ばれ，米国では政府主導の Smart American Challenge [5]や General Electronic（GE）社のインダストリアル・インターネット [6]が，ドイツでは IoT 化によって工場の生産性を向上させるインダストリー4.0 [7]が推進されている。以上のように，IoT によって膨大なデータを収集・蓄積・解析することで，より良いサービスを生み出し，人々の生活の向上や，新しい価値・ビジネスモデルの創造が期待されている。

一方でこれら IoT 機器では，適切にセキュリティを確保することができない場合，様々なモノが繋がることによって，情報の漏えいや改ざんだけでなく，機器が乗っ取られることによる深刻な損害が生じる恐れがある [8]。そのため，セキュリティの確保は重要な課題である。セキュリティの確保に関して，やり取りされる情報の秘匿化や機器間の認証，データの改ざんを防止することが必要であり，暗号技術はこれらを実現するための最も重

要な技術である。

そして、この暗号技術には米国立標準技術研究所（National Institute of Standards and Technology: NIST）によって公募された標準暗号（Advanced Encryption Standard: AES）[9]が広く用いられている。しかしながら、IoT で用いられる組み込み機器では利用できる回路規模や消費電力、処理時間に制約のあるものも多く、AES 暗号を適用することが難しいことが指摘されている [10]。そのため、このような機器に対して利用可能な暗号技術として軽量暗号の研究が近年盛んに行われている。軽量暗号は 2007 年に発表され、ISO/IEC JTC 1/SC 27 で制定された国際標準規格 ISO/IEC29192-2 に採択されている PRESENT [11]や CLEFIA [12]をはじめとして、多くのアルゴリズムが提案されている [11]–[24]。また、セキュリティの確保にはデータの暗号化だけでなく、認証を行うことが非常に重要であり、暗号化と認証を同時に実現する軽量暗号（認証付暗号とも呼ばれる）のアルゴリズムの提案も行われている。

軽量暗号の国内外での研究開発動向を図 1.1 にまとめる。欧州では、2004 年から 2007 年の 4 年間にかけて、欧州の情報セキュリティの研究者の協力・交流を目的として、欧州委員会の第 6 次研究フレームワークプログラム（Sixth Framework Program: FP6）の Information Societies Technology (IST) プログラムの支援を受け、European Network of Excellence for Cryptology (ECRYPT) [25]が実施された。また、ECRYPT に続き、2008 年から 2013 年にかけては、欧州委員会の第 7 次研究フレームワークプログラム (FP7) の支援のもとで、ECRYPT-II [26]が実施された。ECRYPT, ECRYPT-II では、それぞれ目的に合わせた仮想ラボが構築され、情報セキュリティに関する研究が行われた。そして、これらの成果のうち、軽量暗号に関する研究レポートがいくつか報告されている [26]–[31]。また、欧州委員会の FP6 に支援されている Specific Targeted Research Project (STREP) の Ubiquitous Sensing and Security in the European Homeland (UbiSec&Sens) ではワイヤレスネットワークのセキュリティに関する研究が行われており、その成果として、軽量暗号 DESL [13]や PRESENT [11]が開発されている。

米国では、米国家安全保障局 (National Security Agency: NSA) が軽量暗号 SIMON/SPECK [18]を開発し、公表している。SIMON はハードウェア実装向けに、SPECK はソフトウェア実装向けに最適化された軽量暗号である。また、SIMON/SPECK はそれぞれ ISO/IEC29192-2 へ追加提案を行っていたが、2018 年にこの追加提案は棄却されている。

NIST も 2013 年から軽量暗号の研究開発を進めており、2015 年と 2016 年には、産学官

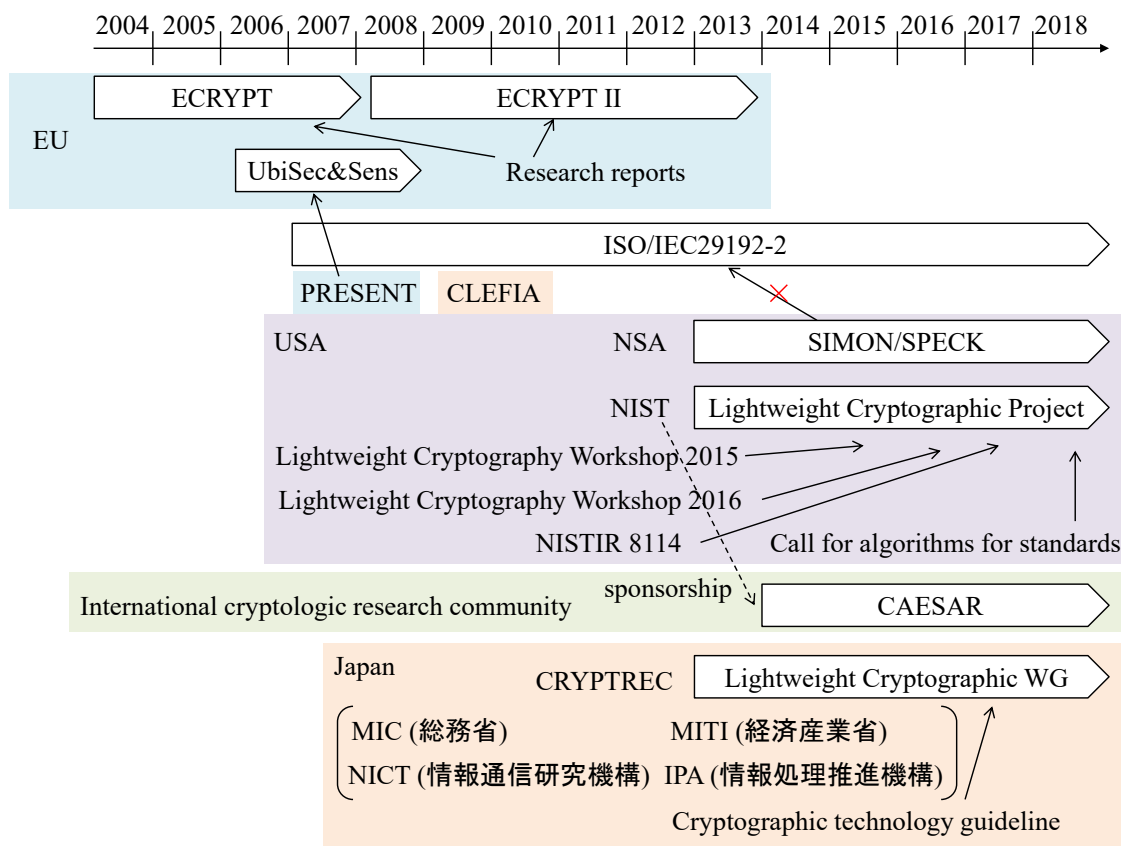


図 1.1 国内外の軽量暗号の研究開発の動向

Fig. 1.1 Trends of researches for lightweight ciphers in domestic and international

の専門家を集めたワークショップ（Lightweight Cryptography Workshop 2015 [33]，Lightweight Cryptography Workshop 2016 [34]）を開催している。また，2017 年 3 月には軽量暗号の標準化に向けた計画案 NISTIR 8114 が報告されている [35]。そして現在，2019 年 2 月を締切として，標準化のための軽量暗号アルゴリズムの公募を開始している [36]。

また，NIST は認証付暗号のコンペティション（Competition for Authenticated Encryption: Security, Applicability, and Robustness: CAESAR）[37]への出資も行っている。CAESAR は国際的な暗号研究者コミュニティによって，2014 年から 2018 年に実施されたもので，ISO/IEC 19772 でも標準化されている NIST 推奨の認証付暗号 AES-Galois/Counter Mode (AES-GCM) [38]をベンチマークとし，AES-GCM よりも優れた暗号方式の選定を行っている。コンペティションでは 57 個の方式が応募され，1 次選考，2 次選考で 15 個まで候補が絞られ，3 次選考を経て 7 個の方式が最終候補として選ばれており，現在最終ポートフォリオの選定が行われている。

国内では、CRYPTREC (Cryptography Research and Evaluation Committees) による軽量暗号ワーキンググループ (Working Group: WG) が 2013 年に発足している。CRYPTREC は、電子政府推奨暗号の安全性の評価や監視および、暗号技術の適切な実装方法や運用方法に関する調査や検討を行うプロジェクトである。CRYPTREC は主に、暗号技術検討会と暗号技術評価委員会、暗号技術活用委員会で構成しており、それぞれ暗号技術検討会は総務省と経済産業省が、暗号技術評価委員会と暗号技術活用委員会は国立研究開発法人情報通信研究機構 (National Institute of Information and Communications Technology: NICT) と独立行政法人情報処理推進機構 (Information-technology Promotion Agency: IPA) が運営している。CRYPTREC 軽量暗号 WG は、暗号技術活用 (軽量暗号) ガイドライン [39] を発行し、様々な軽量暗号の実装性能を比較し、どのアプリケーションに対してどの軽量暗号が適しているのかについて議論がなされている。

1.2 暗号回路に対する物理解析

軽量暗号を含む暗号アルゴリズムは、通常計算量的にその安全性が保障されている。例えば、軽量暗号では鍵長として 80bit または 128bit が広く用いられており、鍵候補を 1bit ずつ全て探索する総当たり攻撃によって秘密鍵を解析する場合、その計算量は 2^{80} または 2^{128} となり、この計算を行うのは困難である。しかし、計算量的に安全な暗号アルゴリズムであっても、ソフトウェアやハードウェアで実装されたもの (暗号回路) を対象に、秘密鍵を推定する物理解析の脅威が報告されている [40]–[47]。物理解析は主に侵襲解析、半侵襲解析、非侵襲解析に分類される。侵襲解析・半侵襲解析は回路のパッケージの開封を伴い、解析に高価な装置や高度な専門知識を必要とする [40][41]。一方で、非侵襲解析であるサイドチャネル解析は、暗号回路から漏洩する情報を利用して解析を行う [42]–[47]。サイドチャネル解析は、主に回路動作時の消費電力を利用する電力解析 [42][43]、漏洩電磁波を利用する電磁波解析 [44][45]、暗号回路を誤動作させることで得られる情報を利用するフォールト解析 [46][47] などがある。また、サイドチャネル解析は比較的安価な機器で行うことができ、解析の痕跡を残さないため、非常に強力な解析手法である。そのため、サイドチャネル解析に対する安全性を確保することは非常に重要であり、これまでに Data Encryption Standard (DES) 暗号 [48] や AES 暗号などの共通鍵暗号をはじめ、RSA 暗号 [49] や楕円曲線暗号 (Elliptic Curve Cryptography: ECC) [50] などの公開鍵暗号を対象としたサイドチャネル解析の研究が数多く行われている。

1.3 本研究の目的と論文構成

サイドチャネル解析は暗号回路にとって非常に脅威となる解析であり、CRYPTREC 軽量暗号 WG の軽量暗号活用ガイドライン [39]でも述べられている通り、今後 IoT によって様々な機器での利用が期待される軽量暗号において、その安全性について検討することは非常に重要である。

本研究の目的は、IoT 時代において今後利用が期待される軽量暗号の安全性（耐タンパ性）を検証するために必要なサイドチャネル解析手法を開発することにある。具体的に本論文では、小回路規模実装・低消費電力実装・低遅延実装・認証付暗号に加え、商用利用の5つを軽量暗号における重要な要素と位置付ける。そして表 1.1 に示すように、これら5大要素の代表的な軽量暗号 TWINE [16], SIMECK [19], Midori [20], PRINCE [17], Minalpher [23], TWINE-Offset Two-Round (TWINE-OTR) [24]に対するサイドチャネル解析手法を提案する。

本論文の全体の構成図を図 1.2 に示す。本論文は 13 章で構成する。

第 2 章では、基本的な暗号技術の分類や実装方法、本論文で対象とする各軽量暗号アルゴリズムについて概説する。

第 3 章では、暗号回路に対する物理解析について、サイドチャネル解析を中心に説明する。

第 4 章から第 9 章では、軽量暗号に対する電力・電磁波解析手法を提案する。

具体的に第 4 章では、TWINE を対象とした電力解析手法を提案し、Field Programmable Gate Array (FPGA) を用いた評価実験によってその有効性を示す。提案する電力解析手法では、解析効率を向上させるために、周波数領域で電力解析を適用する。更に、複数ラウンドに対する解析では別のラウンドで解析済の部分鍵を他のラウンドの解析に組込む処理を導入することで解析効率の改善を図る。

表 1.1 5 大要素の軽量暗号

Table 1.1 Lightweight ciphers of five elements

	Commercial	Small area	Low power	Low latency	Authenticated encryption
Lightweight ciphers	TWINE	SIMECK	Midori	PRINCE	Minalpher TWINE-OTR

第 5 章では、SIMECK を対象とした電磁波解析手法を提案し、FPGA を用いた評価実験によってその有効性を示す。提案する電磁波解析手法では、SIMECK の構造に着目した複数のラウンドを利用した解析や、既知情報の組み込み処理を導入することで、解析効率を向上させる。

第 6 章では、Midori を対象とした電力解析手法を提案し、FPGA を用いた評価実験によってその有効性を示す。提案する電力解析手法では、2 段階の階層的な電力解析手法を導入することで、全ての秘密鍵を推定する。また、Midori のアルゴリズムに着目した解析の効率化手法を導入することで、解析に必要な計算量を大幅に削減する。

第 7 章と第 8 章では、Minalpher と TWINE-OTR に対する電力解析手法を提案し、FPGA を用いた評価実験によってその有効性を示す。提案する電力解析手法では、認証付暗号のアルゴリズムに着目した、2 段階の解析手法を導入することで、秘密鍵を解析する。

第 9 章から第 12 章では、軽量暗号に対するフォールト解析手法を提案する。

具体的に第 9 章では、TWINE を対象としたフォールト解析手法を提案し、FPGA を用いた評価実験によってその有効性を示す。提案するフォールト解析手法では、クロックグリッチによるフォールトでも解析が可能な統計的な解析手法を導入する。

第 10 章では、SIMECK を対象としたフォールト解析手法を提案し、シミュレーション実験によってその有効性を示す。提案するフォールト解析手法では、クロックグリッチによるフォールトを想定したフォールトモデルを導入する。また、SIMECK の構造に起因したフォールト伝搬を利用した 2 段階の解析手法を導入することで、ラウンド鍵を解析する。

第 11 章では、Midori を対象としたフォールト解析手法を提案し、シミュレーション実験によってその有効性を示す。提案するフォールト解析手法では、クロックグリッチによるフォールトでも解析が可能な統計的な解析手法を導入する。また、Midori のアルゴリズムに合わせた階層的な解析方法を導入することで、全ての秘密鍵を推定する。

第 12 章では、PRINCE に対するフォールト解析手法を提案し、シミュレーション実験によってその有効性を示す。提案するフォールト解析手法では、フォールト混入位置を精密に制御できない場合でも解析が可能な統計的な解析モデルを導入する。また、全ての秘密鍵を推定するために、階層的な解析モデルを導入する。更に、枝刈りアルゴリズムを導入することで、解析に必要な処理時間・計算量を削減する。

そして最後に、第 13 章では本論文で提案した軽量暗号に対するサイドチャネル解析手法のまとめと今後の展望について述べる。

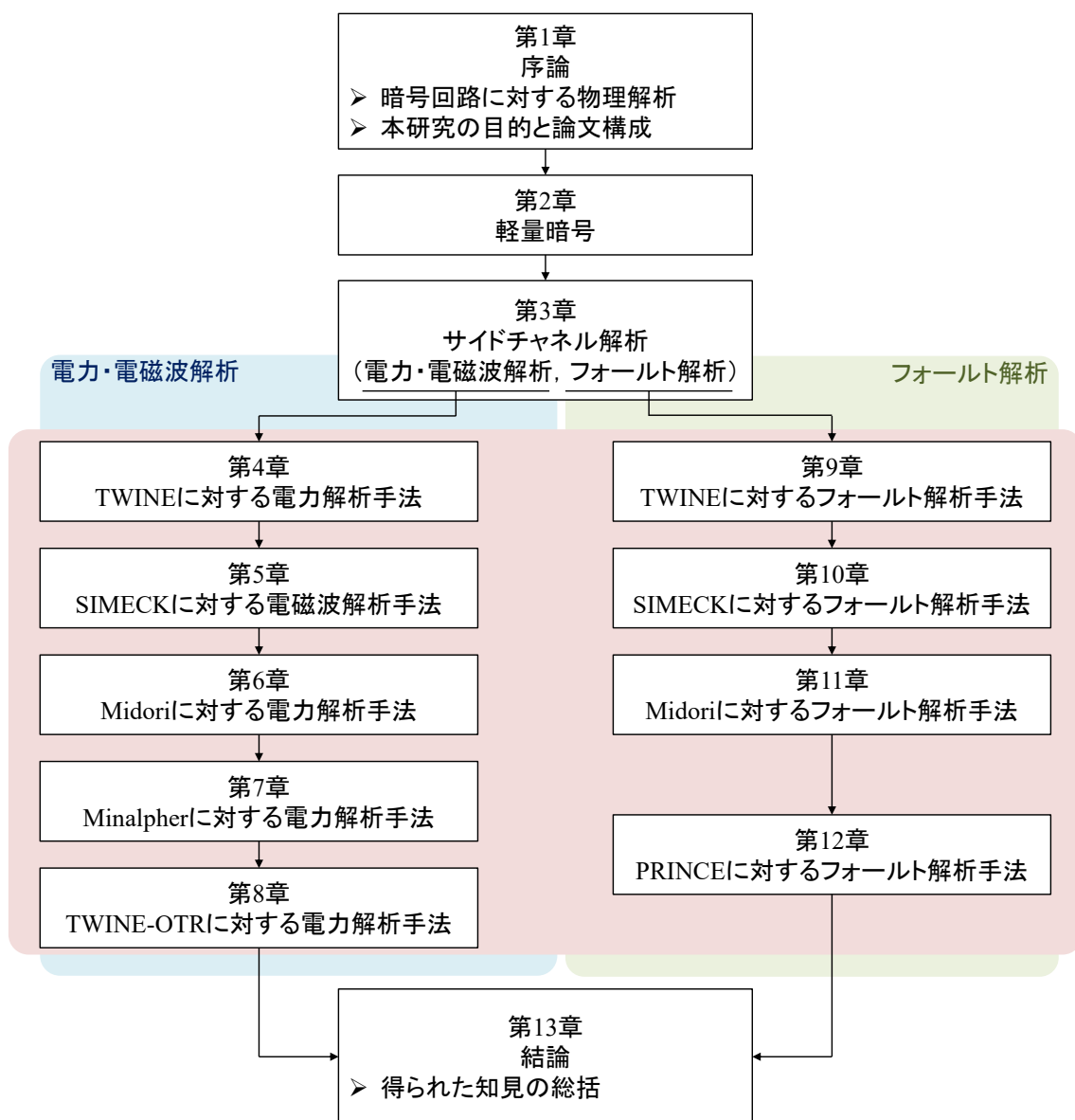


図 1.2 本論文の構成

Fig. 1.2 Structure of this paper

参考文献

- [1] “総務省: 平成 30 年度版情報通信白書,”
<http://www.soumu.go.jp/johotsusintokei/whitepaper/h30.html>
- [2] M. Alioto and M. Shahghasemi, “The Internet of Things on Its Edge: Trends Toward Its Tipping Point,” *IEEE Consumer Electronics Magazine*, vol. 7, no. 1, pp. 77–87, Jan. 2018.
- [3] 佐藤 証, “都市型農業のサービス産業化を実現する水耕栽培システム,” マルチメディア, 分散, 協調とモバイル (DICOMO 2018) シンポジウム論文集, pp. 925–928, Jul. 2018.
- [4] 倉地 亮, “自動車を取り巻く IoT セキュリティとその課題: 自動運転, 車車間, 路車間の V2X に関するセキュリティを中心に,” *情報管理*, vol. 60, no. 10, pp. 690–700, Jan. 2018.
- [5] “Smart America Challenge,” <http://smartamerica.org/>
- [6] “General Electronic,” <http://www.ge.com/digital/industrial-internet>
- [7] R. Drath and A. Horch, “Industrie 4.0: Hit or Hype?,” *IEEE Industrial Electronics Magazine*, vol. 8, no. 2, pp. 56–58, Jun. 2014.
- [8] 吉岡克成, “これからのネットワークセキュリティ,” *電子情報通信学会誌*, vol. 100, no. 3, pp. 208–213, Mar. 2017.
- [9] National Institute of Standards and Technology, “Specification for the Advanced Encryption Standard (AES),” *Federal Information Processing Standards Publication 197*, Nov. 2001.
<http://doi.org/10.6028/NIST.FIPS.197>
- [10] 鈴木大輔, 菅原 健, 佐伯 稔, “軽量/低遅延暗号のハードウェア実装性能について,” 2014 年暗号と情報セキュリティシンポジウム講演論文集, 2A2-2, pp. 1–6, Jan. 2014.
- [11] A. Bogdanav, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, and C. Vikkelse, “PRESENT: An Ultra-Lightweight Block Cipher,” *Proc. 9th Int. Workshop on Cryptographic Hardware and Embedded Systems (CHES 2007)*, LNCS vol. 4727, pp. 450–466, Springer-Verlag, Sep. 2007.
- [12] G. Leander, C. Paar, A. Poschmann, and K. Schramm, “New Lightweight DES Variants,”

- Proc. 14th Int. Workshop on Fast Software Encryption (FSE 2007), LNCS vol. 4593, pp. 196–210, Springer-Verlag, Mar. 2007.
- [13] T. Shirai, K. Shibutani, T. Akishita, S. Moriai, and T. Iwata, “The 128-bit Blockcipher CLEFIA,” Proc. 14th Int. Workshop on Fast Software Encryption (FSE 2007), LNCS vol. 4593, pp. 181–195, Springer-Verlag, Mar. 2007.
- [14] J. Guo, T. Peyrin, A. Poschmann, M. Robshaw, “The LED Block Cipher,” Proc. 13th Int. Workshop on Cryptographic Hardware and Embedded Systems (CHES 2011), LNCS vol. 6917, pp. 326–341, Springer, Sep. 2011.
- [15] K. Shibutani, T. Isobe, H. Hiwatari, A. Mitsuda, T. Akishita, and T. Shirai, “Piccolo: An Ultra-Lightweight Blockcipher,” Proc. 13th Int. Workshop on Cryptographic Hardware and Embedded Systems (CHES 2011), LNCS vol. 6917, pp. 342–357, Springer, Sep. 2011.
- [16] T. Suzaki, K. Minematsu, S. Morioka, and E. Kobayashi, “TWINE: A Lightweight, Versatile Blockcipher,” Proc. ECRYPT Workshop on Lightweight Cryptography (LC11), pp. 146–149, Nov. 2011.
- [17] J. Borghoff, A. Canteaut, T. Güneysu, E. B. Kavum, M. Knežević, L. R. Knudsen, G. Leander, V. Nikov, C. Paar, C. Rechberger, P. Rombouts, S. S. Thomsen, and T. Yalçin, “PRINCE - A Low-latency Block Cipher for Pervasive Computing Applications,” Proc. ASIACRYPT 2012, LNCS vol. 7658, pp. 208–225, Springer, Dec. 2012.
- [18] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers, “The SIMON and SPECK Families of Lightweight Block Ciphers,” IACR Cryptography ePrint Archive: Report 2013/404, Jun. 2013.
- [19] G. Yang, B. Zhu, V. Suder, M. D. Aagaard, and G. Gong, “The Simeck Family of Lightweight Block Ciphers,” Proc. 17th Int. Workshop on Cryptographic Hardware and Embedded Systems (CHES 2015), LNCS vol. 9293, pp. 307–329, Springer, Sep. 2015.
- [20] S. Banik, A. Bogdanov, T. Isobe, K. Shibutani, H. Hiwatari, T. Akishita, and F. Regazzoni, “Midori: A Block Cipher for Low Energy,” Proc. ASIACRYPT 2015, LNCS vol. 9453, pp. 411–436, Springer, Dec. 2015.
- [21] C. Beierle, J. Jean, S. Kölbl, G. Leander, A. Moradi, T. Peyrin, Y. Sasaki, P. Sasdrich, and S. M. Sim, “The SKINNY Family of Block Ciphers and Its Low-Latency Variant MANTIS,” Proc. 36th Annual Int. Cryptology Conf. (CRYPTO 2016), LNCS vol. 9815, pp. 123–153,

Springer, Aug. 2016.

- [22] S. Banik, S. K. Pandey, T. Peyrin, Y. Sasaki, S. M. Sim, Y. Todo, “GIFT: A Small Present - Towards Reaching the Limit of Lightweight Encryption,” Proc. 19th Int. Workshop on Cryptographic Hardware and Embedded Systems (CHES 2017), LNCS vol. 10529, pp. 321–345, Springer, Sep. 2017.
- [23] Y. Sasaki, Y. Todo, K. Aoki, Y. Naito, T. Sugawara, Y. Murakami, M. Matsui, and S. Hirose, “Minalpher v1.1,” Aug. 2015. <http://info.isl.ntt.co.jp/crypt/minalpher/files/minalpherv11.pdf>
- [24] K. Minematsu, “AES-OTR v3.1,” Sep. 2016.
<http://competitions.cr.yp.to/round3/aesotr31.pdf>
- [25] “European Network of Excellence in Cryptology II,”
<http://www.ecrypt.eu.org/ecrypt2/>
- [26] D.STVL.3, “Ongoing Research Areas in Symmetric Cryptography,” Jan. 2005.
<http://www.ecrypt.eu.org/ecrypt1/documents/D.STVL.3-2.5.pdf>
- [27] D.AZTEC.2, “Lightweight Asymmetric Cryptography and Alternatives to RSA,” Aug. 2005.
<http://www.ecrypt.eu.org/ecrypt1/documents/D.AZTEC.2-1.2.pdf>
- [28] D.VAM.11, “Suggested Algorithms for Light-Weight Cryptography,” Sep. 2006.
<http://www.ecrypt.eu.org/ecrypt1/documents/D.VAM.11-1.0.pdf>
- [29] D.SYM.5, “Lightweight Cryptographic Algorithms,” Jul. 2010.
<http://www.ecrypt.eu.org/ecrypt2/documents/D.SYM.5.pdf>
- [30] D.SYM.8, “Second Lightweight Cryptography Status Report,” Jan. 2012.
<http://www.ecrypt.eu.org/ecrypt2/documents/D.SYM.8.pdf>
- [31] D.SYM.12, “Final Lightweight Cryptography Status Report,” Oct. 2012.
<http://www.ecrypt.eu.org/ecrypt2/documents/D.SYM.12.pdf>
- [32] D.VAM.2, “Report on Lightweight Cryptography,” Jul. 2010.
<http://www.ecrypt.eu.org/ecrypt2/documents/D.VAM.2.pdf>
- [33] “NIST Lightweight Cryptography Workshop 2015,”
<http://www.nist.gov/news-events/events/2015/07/lightweight-cryptography-workshop-2015>
- [34] “NIST Lightweight Cryptography Workshop 2016,”
<http://www.nist.gov/news-events/events/2016/10/lightweight-cryptography-workshop-2016>
- [35] K. A. McKay, L. Bassham, M. S. Turan, and N. Mouha, “National Institute of Standards and

- Technology Internal Report 8114,” NIST, pp. 1–21, Mar. 2017.
<http://doi.org/10.6028/NIST.IR.8114>
- [36] “NIST Lightweight Cryptography,”
<http://csrc.nist.gov/projects/lightweight-cryptography>
- [37] “CAESAR: Competition for Authenticated Encryption: Security, Applicability, and Robustness,”
<http://competitions.cr.yp.to/caesar.html>
- [38] M. Dworki, “Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC,” National Institute of Standards and Technology Internal Special Publication 800-38D, Nov. 2007.
- [39] CRYPTREC, “CRYPTREC Cryptographic Technology Guideline (Lightweight Cryptography),” Mar. 2017.
<http://www.cryptrec.go.jp/report/cryptrec-gl-0001-2016-e.pdf>
- [40] O. Kömmerling and M. G. Kuhn, “Design Principles for Tamper-Resistant Smartcard Processors,” Proc. USENIX Workshop on Smartcard Technology (Smartcard’99), pp. 9–20, May 1999.
- [41] S. P. Skorobogatov, “Semi-Invasive Attacks – A New Approach to Hardware Security Analysis,” PhD thesis, University of Cambridge, Apr. 2005.
- [42] P. Kocher, J. Jaffe, and B. Jun, “Differential Power Analysis,” Proc. CRYPTO’99, LNCS vol. 1666, pp. 388–397, Springer-Verlag, Dec. 1999.
- [43] E. Brier, C. Clavier, and F. Olivier, “Correlation Power Analysis with a Leakage Model,” Proc. 6th Int. Workshop on Cryptographic Hardware and Embedded Systems (CHES 2004), LNCS vol. 3156, pp. 16–29, Springer-Verlag, Aug. 2004.
- [44] K. Gandolfi, C. Moutrel, and F. Olivier, “Electromagnetic Analysis: Concrete Results,” Proc. 3rd Int. Workshop on Cryptographic Hardware and Embedded Systems (CHES 2001), LNCS vol. 2162, pp. 251–261, Springer-Verlag, May 2001.
- [45] O. Meynard, S. Guilley, J.-L. Danger, and L. Sauvage, “Far Correlation-based EMA with a Precharacterized Leakage Model,” Proc. Design, Automation and Test in Europe Conference and Exhibition (DATE 2010), pp. 977–980, Mar. 2010.
- [46] D. Boneh, R. A. DeMillo, and R. J. Lipton, “On the Importance of Checking Cryptographic

- Protocols for Faults,” Proc. 14th Annual EUROCRYPT Conf. (EUROCRYPT’97), LNCS vol. 1233, pp. 37–51, Springer-Verlag, May 1997.
- [47] G. Piret, J. -J. Quisquater, “A Differential Fault Attack Technique against SPN Structures, with Application to the AES and Khazad,” Proc. 5th Int. Workshop on Cryptographic Hardware and Embedded Systems (CHES 2003), LNCS vol. 2779, pp. 77–88, Springer-Verlag, Sep. 2003.
- [48] National Institute of Standards and Technology, “Data Encryption Standard (DES),” Federal Information Processing Standards Publication 46-3, Oct. 1999.
<http://csrc.nist.gov/CSRC/media/Publications/fips/46/3/archive/1999-10-25/documents/fips46-3.pdf>
- [49] R. L. Rivest, A. Shamir, and L. Adleman, “A method for obtaining digital signatures and public-key cryptosystems,” Communications of the ACM, vol. 21, no. 2, pp. 120–126, Feb. 1978.
- [50] N. Koblitz, “Elliptic Curve Cryptosystems,” Mathematics of Computation, vol. 48, no. 177, pp. 203–209, Jan. 1987.

第 2 章

軽量暗号

2.1 緒言

IoT によって、面積などの制約を含む様々な機器で暗号技術が必要となっている。しかしながら、これらのデバイスでは一般的に広く使用されている AES 暗号 [1]を実装することが難しいことが指摘されている [2]。軽量暗号は、そのようなデバイスでも実装可能な暗号技術としてこれまで提案されてきた [3]-[10]。

まず 2.2 節では、基本的な暗号技術の概要について、暗号技術の分類・実装を中心に説明する。次に 2.3 節では、軽量暗号の優位性を示すための比較として用いられる AES 暗号について説明する。最後に 2.4 節では各種軽量暗号について概説する。

2.2 暗号技術の概要

2.2.1 暗号技術の分類

図 2.1 に暗号技術の分類を示す。図 2.1 に示すように、暗号技術は共通鍵暗号と公開鍵暗号に大別される。本論文では共通鍵暗号を対象とする。また、各方式について以下に示す。

共通鍵暗号

共通鍵暗号は、暗号化と復号に同一の秘密鍵を使用する暗号方式であり、ブロック暗号とストリーム暗号に大別される。本論文では、特にブロック暗号を対象とする。

ブロック暗号の概要を図 2.2 に示す。ブロック暗号は暗号化する入力データを、 n bit のブロックと呼ばれる単位に分割し、ブロックごとに処理を行う。このとき、ブロック暗号は暗号処理部と鍵スケジュール部に分かれており、鍵スケジュール部では秘密鍵から暗号

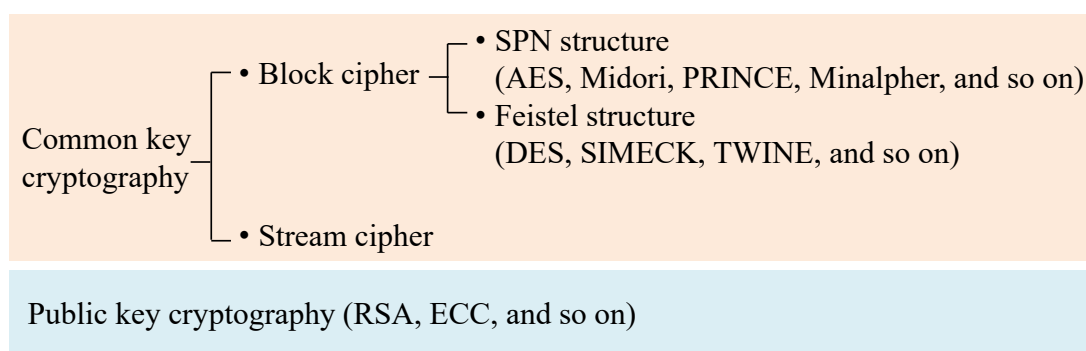


図 2.1 暗号技術の分類

Fig. 2.1 Classification for encryption techniques

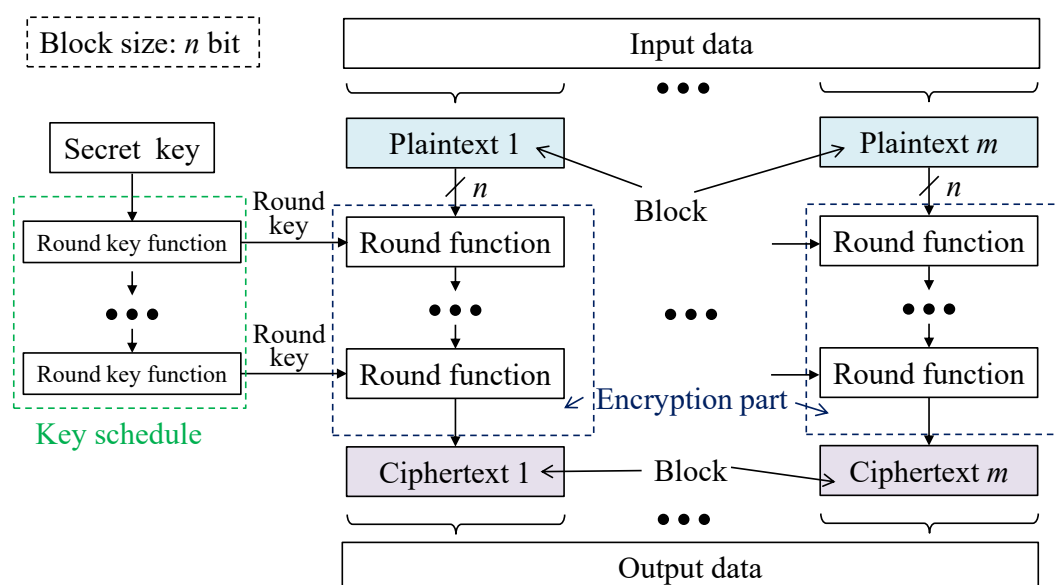


図 2.2 ブロック暗号の概要

Fig. 2.2 Outline of block cipher

処理部で使用するラウンド鍵を生成する。各処理では、ラウンド関数と呼ばれる処理を繰り返し適用し、得られた値を暗号文として出力する。このとき、ブロックの長さをブロック長、繰り返し処理の回数をラウンド数と呼ぶ。例えば、AES 暗号ではブロック長に 128bit が用いられ、鍵長が 128bit の場合、ラウンド数は 10 となる。また、主なブロック暗号には、AES 暗号 [1]や DES 暗号 [11], 本論文で対象とする軽量暗号 TWINE [4], SIMECK [5], Midori [6], PRINCE [7], Minalpher [8]などがある。

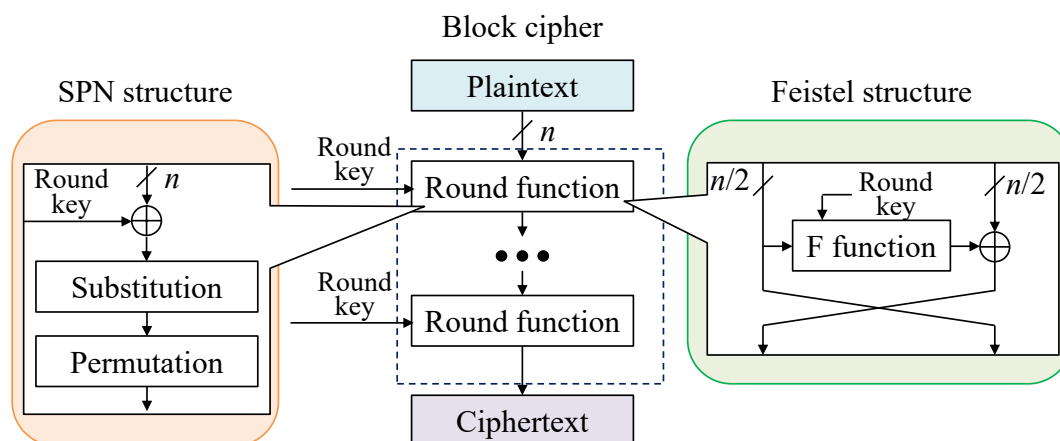


図 2.3 SPN 構造と Feistel 構造

Fig. 2.3 SPN structure and Feistel structure

ブロック暗号の構造に関して、主に Substitution Permutation Network (SPN) 構造と Feistel 構造がある。SPN 構造と Feistel 構造を図 2.3 に示す。SPN 構造は、図 2.3 に示すように 1 つのラウンド処理を換字処理 (Substitution) と転置処理 (Permutation) で構成する。主な SPN 構造のブロック暗号には、AES 暗号、Midori, PRINCE, Minalpher などがある。Feistel 構造は、図 2.3 に示すように、入力される n bit のブロックのデータを半分ずつに分割し、処理する。主な Feistel 構造のブロック暗号には、DES 暗号や SIMECK, TWINE (Feistel 構造を拡張した 16 分割一般化 Feistel 構造) などがある。

ストリーム暗号は、秘密鍵を入力とした乱数生成器の出力列を鍵として、平文と排他的論理和演算することで、暗号文を生成する。主なストリーム暗号には、SNOW [12]や MUGI [13]などがある。

公開鍵暗号

公開鍵暗号は、暗号化と復号に用いる鍵は異なり、暗号化には公開鍵を使用し、復号には秘密鍵を使用する。主な公開鍵暗号には、RSA 暗号 [14]や ECC [15]などがある。

2.2.2 暗号実装

暗号技術は通常、何らかのデバイスに実装され使用される。この暗号実装にはハードウェア実装とソフトウェア実装がある。本論文では主にハードウェア実装を対象とする。各実装について、以下に示す。

ハードウェア実装

ハードウェア実装は、暗号アルゴリズムを Application Specific Integrated Circuit (ASIC) や FPGA の回路上で実現する。ハードウェア実装は、暗号アルゴリズム専用の回路を使用することができるため、小回路規模で実装でき、後述するソフトウェア実装と比べ処理速度や消費電力の面で優れている。そのため、小回路規模・低消費電力・低遅延での実装が必要なデバイスでは、ハードウェア実装が必要とされる。

ソフトウェア実装

ソフトウェア実装は、暗号アルゴリズムを C 言語などのプログラミング言語を使用して記述し、Central Processing Unit (CPU) などの汎用プロセッサを用いて実現する。ソフトウェア実装は汎用の命令セットを使用するため、プログラムの変更・修正を柔軟に行うことができる。一方で、専用回路を使用しないため、ハードウェア実装と比べ処理速度や消費電力の面で劣る。

2.3 AES 暗号

AES 暗号 [1]は、NIST によって標準化された SPN 構造を持つブロック暗号である。また、AES 暗号は ISO/IEC 18033-3 でも標準化されている。ブロック長は 128bit であり、鍵長は 128bit, 196bit, 256bit の 3 種類から選択可能である。ここでは、特に鍵長 128bit の場合を例に説明する。

AES 暗号の概要を図 2.4 に示す。AES 暗号は 128bit (16byte) の平文を、4×4 の State と呼ばれる単位で処理する。主に、暗号処理部と鍵スケジュール部で構成しており、合計で 10 ラウンドの処理を行い、暗号化する。暗号処理部での各ラウンドでは、図 2.4 に示す SubBytes, ShiftRows, MixColumn, AddRoundKey で構成するラウンド関数 F を用いた処理を行う。10 ラウンド目 (10R) ではラウンド関数 F から MixColumn を除いた処理を行う。また、AddRoundKey で使用するラウンド鍵は鍵スケジュール部で生成する。

ラウンド関数 F の各処理に関して、AddRoundKey では State の各値に対してラウンド鍵との排他的論理和演算を行う。SubBytes では S-BOX と呼ばれる非線形な置換処理によって、8bit 単位で値が置換される。ShiftRows では State の各列に対して、それぞれ 0, 1, 2, 3 の左シフトが行われる。MixColumn では、State の各列に対して式(2.1)に示す行列 a による計算を行う。

$$a = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix}_{16} \quad (2.1)$$

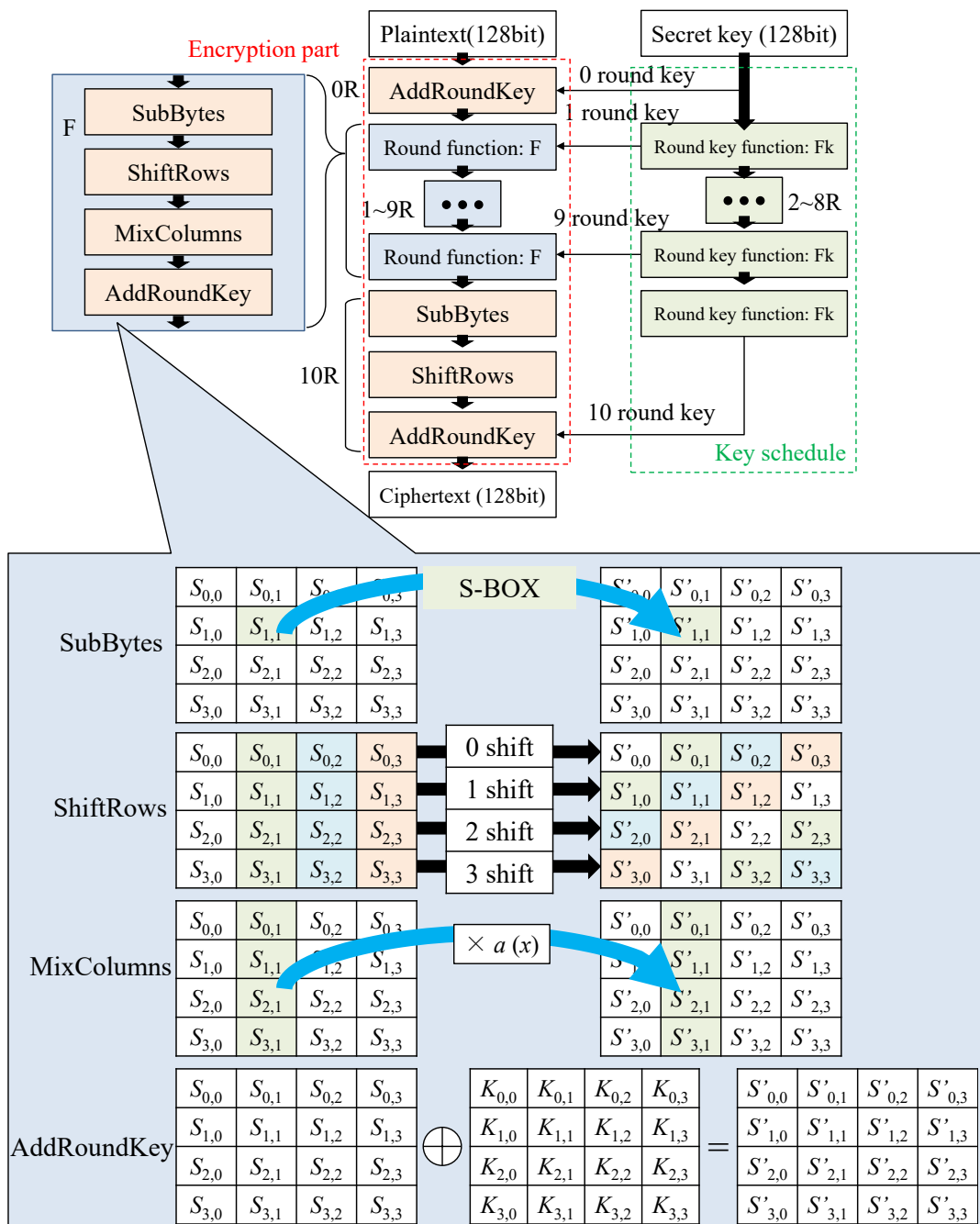


図 2.4 AES 暗号の概要

Fig. 2.4 Outline of AES encryption

また AES 暗号のハードウェア実装では、ラウンド関数のうち SubBytes が回路規模や消費電力の大半を占める [16]。そのため、S-BOX の実装方法に関して多くの研究が報告されている [16]–[19]。

2.4 軽量暗号の概要

軽量暗号は、AES 暗号では実装が難しい小型デバイスでも実装できるアルゴリズムとして提案されてきた。例えば、AES 暗号はブロック長が 128bit、鍵長は 128bit、192bit、256bit から選択可能で、S-BOX は 8bit 単位のものを使用しているのに対して、軽量暗号ではブロック長が 64bit、鍵長が 80bit、S-BOX は 4bit 単位を採用しているものが多く、小回路規模化を実現させている。また、軽量暗号には AES 暗号と比較して鍵スケジュール部を持たないものや、暗号化と復号を同一の回路で実現可能なものもあり、小型デバイスへの実装に向けて様々な工夫がなされている。

具体的に、これまでに ISO/IEC29192-2 に採択された PRESENT [3]をはじめ、様々な要求に合わせて多くのアルゴリズムが提案されてきた。本論文では、図 2.5 に示すように商用利用・小回路規模実装・低消費電力実装・低遅延実装・認証付暗号の 5 つを軽量暗号における重要な要素と位置付け、これら 5 大要素の代表的な軽量暗号 TWINE, SIMECK, Midori, PRINCE, Minalpher, TWINE-OTR を対象とする。

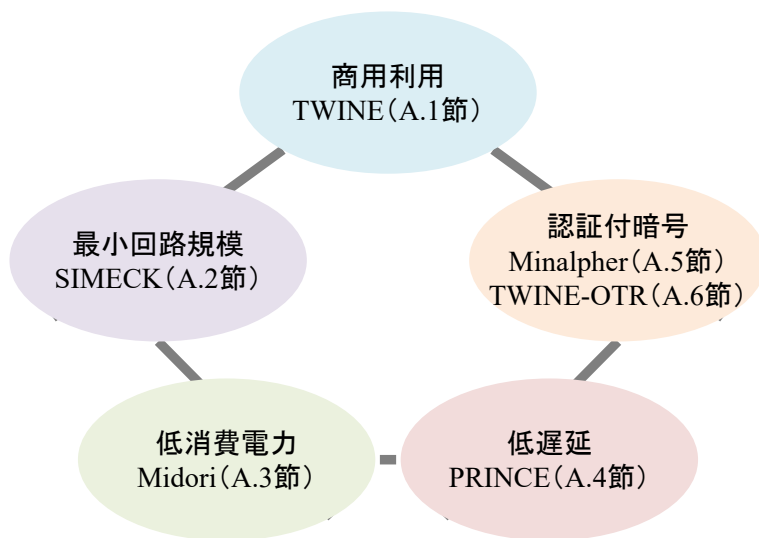


図 2.5 本論文で対象とする軽量暗号

Fig. 2.5 Lightweight ciphers targeted in this paper

2.4.1 各種軽量暗号アルゴリズム

本項では、本論文で対象とする代表的な軽量暗号アルゴリズムの概要について述べる。また、各アルゴリズムの詳細については、付録で説明する。

TWINE

TWINE [4]は、2011 年に日本電気株式会社によって発表された軽量暗号であり、CRYPTREC 軽量暗号 WG 発行の軽量暗号活用ガイドライン [20]においても評価対象となっている。また TWINE は、軽量暗号の 5 大要素の 1 つである商用利用がなされており、ICT プラットフォーム [21]や IoT 化した防犯カメラ [22]などのシステムで使用されている。TWINE は 16 分割一般化 Feistel 構造のブロック暗号であり、ブロック長は 64bit、鍵長は 80bit と 128bit の 2 種類から選択でき、ラウンド数は 36 である。TWINE のアルゴリズムの詳細は、付録（A.1 節）で説明する。

SIMECK

SIMECK [5]は 2015 年に CHES で提案された軽量暗号であり、軽量暗号の 5 大要素の 1 つである小回路規模での実装が可能であり、現在報告されている軽量暗号の中で最も小回路規模でハードウェア実装することができる。SIMECK は Feistel 構造のブロック暗号であり、ブロック長は 32, 48, 64bit、鍵長は 64, 96, 128bit から決められた組み合わせを選択し、この組み合わせによってラウンド数は変化する。

また SIMECK は、NSA によって公開された SIMON/SPECK をベースに設計されており、S-BOX を持たず非線形演算は AND 演算のみで構成している。また、回路規模を削減するためのチューニングを加えることで、SIMON/SPECK よりも小回路規模でのハードウェア実装に成功している。SIMECK のアルゴリズムの詳細は、付録（A.2 節）で説明する。

Midori

Midori [6]は、2015 年に ASIACRYPT で提案された軽量暗号であり、軽量暗号の 5 大要素の 1 つである低消費電力を指向して設計されている。また、Midori は CRYPTREC 軽量暗号 WG 発行の軽量暗号活用ガイドライン [20]において評価対象となっている。Midori は SPN 構造を持つブロック暗号で、鍵長は 128bit であり、ブロック長は 64bit と 128bit から選択でき、選択したブロック長によってラウンド数は異なる。

Midori は低消費電力を指向して設計されているため、暗号化と復号に必要な平均消費電

力は、AES 暗号と比較して約 0.15 倍、PRESENT と比較して約 0.78 倍となっており、低消費電力化に成功している [6]。このように、Midori は他の暗号アルゴリズムと比較して低消費電力で動作させることができるため、バッテリー寿命を長くさせることができ、センサ機器などへの利用が期待されている。Midori のアルゴリズムの詳細は、付録（A.3 節）で説明する。

PRINCE

PRINCE [7]は、2012 年に ASIACRYPT で発表された軽量暗号であり、軽量暗号の 5 大要素の 1 つである低遅延での利用が可能である。また、PRINCE は CRYPTREC 軽量暗号 WG 発行の軽量暗号活用ガイドライン [20]において評価対象となっている。ブロック長は 64bit、秘密鍵は 128bit であり、SPN 構造を持つブロック暗号である。PRINCE は、主に低遅延での動作を指向して設計されたアルゴリズムであるため、低遅延での動作に向いているアンロールド型アーキテクチャでの実装が想定されている。ここでアンロールド型アーキテクチャは、全ての処理を 1 クロックサイクルで行う、すなわち全ての処理を組み合わせ回路で実現する実装方式である。

また、PRINCE は鍵スケジュールを持たず、暗号化と復号を同一の回路で実現することができるため、小回路規模でハードウェア実装することができる。文献 [2]によると、アンロールド型アーキテクチャでの実装において、PRINCE は AES 暗号と比較して、100 倍以上小回路規模での実装が可能であり、遅延においては 2 倍以上の優位性があることが報告されている。また、PRESENT と比較しても小回路規模・低遅延で利用することができる。以上のように PRINCE は低遅延で動作させることができるため、メモリ暗号化や車載システムにおけるリアルタイム通信の暗号化などへの利用が期待されている。PRINCE のアルゴリズムの詳細は、付録（A.4 節）で説明する。

Minalpher

Minalpher [8][23][24]は 2014 年に発表された軽量暗号であり、軽量暗号の 5 大要素の 1 つである認証付暗号である。また、Minalpher は CRYPTREC 軽量暗号 WG 発行の軽量暗号活用ガイドライン [20]において評価対象となっている。Minalpher は SPN 構造を持つブロック暗号で、ブロック長は 256bit、鍵長は 128bit であり、処理のコアとなるラウンド関数のラウンド数は 17.5 である。また、CAESAR コンペティション [25]では 1 次選考を通過し、一般的に広く用いられている認証付暗号 AES-GCM [26]よりも、実装面や安全性で優

れている [8]。

Minalpher は 4bit 単位の S-BOX や, Involution 性を持つ構造を組み込むことで, 小回路規模での実装を実現しており, 暗号化と同時に認証が可能のため, センサネットワークなどのセンサ機器での利用が期待されている。Minalpher のアルゴリズムの詳細は, 付録 (A.5 節) で説明する。

TWINE-OTR

TWINE-OTR [9]は, 軽量暗号 TWINE [4]と認証付暗号方式 OTR [9]を組み合わせた暗号で, 軽量暗号の 5 大要素の 1 つである認証付暗号であり, OTR は CRYPTREC 軽量暗号 WG 発行の軽量暗号活用ガイドライン [20]において評価対象となっている。また, CAESAR コンペティションでは 2 次選考を通過している。

TWINE-OTR は小回路規模で実装することが可能であり, IoT デバイスでの利用が期待されている。また, 実際に IoT センサデバイスに搭載されており, 実用化がなされている [27]。TWINE-OTR のアルゴリズムの詳細は, 付録 (A.6 節) で説明する。

2.5 結言

軽量暗号は, 使用用途に合わせて様々なアルゴリズムが提案されている。本章では, まず暗号技術の基本的な分類や実装について説明した。次に, 軽量暗号の比較でよく用いられる AES 暗号について説明した。最後に, 軽量暗号の 5 大要素 (商用利用, 小回路規模, 低消費電力, 低遅延, 認証付暗号) の代表的なアルゴリズムとして, TWINE, SIMECK, Midori, PRINCE, Minalpher, TWINE-OTR について概説した。

参考文献

- [1] National Institute of Standards and Technology, “Specification for the Advanced Encryption Standard (AES),” Federal Information Processing Standards Publication 197, Nov. 2001.
<http://doi.org/10.6028/NIST.FIPS.197>
- [2] 鈴木大輔, 菅原 健, 佐伯 稔, “軽量/低遅延暗号のハードウェア実装性能について,” 2014 年暗号と情報セキュリティシンポジウム講演論文集, 2A2-2, pp. 1–6, Jan. 2014.

- [3] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, and C. Vikkelsoe, “PRESENT: An Ultra-Lightweight Block Cipher,” Proc. 9th Int. Workshop on Cryptographic Hardware and Embedded Systems (CHES 2007), LNCS vol. 4727, pp. 450–466, Springer-Verlag, Sep. 2007.
- [4] T. Suzaki, K. Minematsu, S. Morioka, and E. Kobayashi, “TWINE: A Lightweight, Versatile Blockcipher,” Proc. ECRYPT Workshop on Lightweight Cryptography (LC11), pp. 146–149, Nov. 2011.
- [5] G. Yang, B. Zhu, V. Suder, M. D. Aagaard, and G. Gong, “The Simeck Family of Lightweight Block Ciphers,” Proc. 17th Int. Workshop on Cryptographic Hardware and Embedded Systems (CHES 2015), LNCS vol. 9293, pp. 307–329, Springer, Sep. 2015.
- [6] S. Banik, A. Bogdanov, T. Isobe, K. Shibutani, H. Hiwatari, T. Akishita, and F. Regazzoni, “Midori: A Block Cipher for Low Energy,” Proc. ASIACRYPT 2015, LNCS vol. 9453, pp. 411–436, Springer, Dec. 2015.
- [7] J. Borghoff, A. Canteaut, T. Güneysu, E. B. Kavum, M. Knežević, L. R. Knudsen, G. Leander, V. Nikov, C. Paar, C. Rechberger, P. Rombouts, S. S. Thomsen, and T. Yalçin, “PRINCE - A Low-latency Block Cipher for Pervasive Computing Applications,” Proc. ASIACRYPT 2012, LNCS vol. 7658, pp. 208–225, Springer, Dec. 2012.
- [8] Y. Sasaki, Y. Todo, K. Aoki, Y. Naito, T. Sugawara, Y. Murakami, M. Matsui, and S. Hirose, “Minalpher v1.1,” Aug. 2015. <http://info.isl.ntt.co.jp/crypt/minalpher/files/minalpherv11.pdf>
- [9] K. Minematsu, “AES-OTR v3.1,” Sep. 2016. <http://competitions.cr.yp.to/round3/aesotr31.pdf>
- [10] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers, “The SIMON and SPECK Families of Lightweight Block Ciphers,” IACR Cryptography ePrint Archive: Report 2013/404, Jun. 2013.
- [11] P. Ekdahl and T. Johansson, “A New Version of the Stream Cipher SNOW,” Proc. 9th Annual Int. Workshop on Selected Areas in Cryptography (SAC 2002), LNCS vol. 2595, pp. 47–61, Springer-Verlag, Aug. 2002.
- [12] D. Watanabe, S. Furuya, H. Yoshida, K. Takaragi, and B. Preneel, “A New Key Stream Generator MUGI,” Proc. 9th Int. Workshop on Fast Software Encryption (FSE 2002), LNCS vol. 2365, pp. 179–194, Springer-Verlag, Feb. 2002.

- [13] National Institute of Standards and Technology, “Data Encryption Standard (DES),” Federal Information Processing Standards Publication 46-3, Oct. 1999.
<http://csrc.nist.gov/CSRC/media/Publications/fips/46/3/archive/1999-10-25/documents/fips46-3.pdf>
- [14] R. L. Rivest, A. Shamir, and L. Adleman, “A method for obtaining digital signatures and public-key cryptosystems,” *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, Feb. 1978.
- [15] N. Koblitz, “Elliptic Curve Cryptosystems,” *Mathematics of Computation*, vol. 48, no. 177, pp. 203–209, Jan. 1987.
- [16] 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, May 2003.
- [17] A. Satoh, S. Morioka, K. Takano, and S. Munetoh, “A Compact Rijndael Hardware Architecture with S-Box Optimization,” *Proc. ASIACRYPT 2001*, LNCS vol. 2248, pp. 239–254, Springer-Verlag, Dec. 2001.
- [18] 森岡澄夫, 佐藤 証, 高野光司, “ $GF(((2^2)^2)^2)$ 上の演算を用いた AES の S-Box 構成法,” *情報処理学会第 63 回全国大会講演論文集*, 3G-04, Sep. 2001.
- [19] T. Itoh and S. Tsujii, “A Fast Algorithm for Computing Multiplicative Inverses in $GF(2^m)$ using Normal Bases,” *Information and Computation*, vol. 78, no. 3, pp. 171–177, Sep. 1988.
- [20] CRYPTREC, “CRYPTREC Cryptographic Technology Guideline (Lightweight Cryptography),” Mar. 2017.
<http://www.cryptrec.go.jp/report/cryptrec-gl-0001-2016-e.pdf>
- [21] “NEC、IoT ソリューションを支える ICT プラットフォーム製品と体制を強化,”
http://jpn.nec.com/press/201511/20151109_01.html
- [22] “セブン-イレブン、ALSOK・NEC と連携し、クラウド型の防犯カメラサービスを導入,” http://jpn.nec.com/press/201604/20160421_02.html
- [23] 佐々木悠, 藤堂洋介, 青木和麻呂, 内藤祐介, 菅原 健, 村上ユミコ, 松井 充, 廣瀬勝一, 高橋克己, “改ざん検知暗号 Minalpher,” 2015 年暗号と情報セキュリティシンポジウム講演論文集, 2E-1, pp. 1–4, Jan. 2015.
- [24] 青木和麻呂, 藤堂洋介, “Minalpher の x86_64 最適化実装,” 2015 年暗号と情報セキュリティシンポジウム講演論文集, 1B1-4, pp. 1–8, Jan. 2015.

- [25] “CAESAR: Competition for Authenticated Encryption: Security, Applicability, and Robustness,”
<http://competitions.cr.yp.to/caesar.html>
- [26] M. Dworki, “Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC,” National Institute of Standards and Technology Internal Special Publication 800-38D, Nov. 2007.
- [27] “NEC プラットフォームズ セキュアセンサ BOX,”
<http://www.necplatforms.co.jp/product/securesensor/spec.html>

第 3 章

サイドチャネル解析

3.1 緒言

一般的に暗号技術は、計算量的にその安全性が保障されている。しかしながら、ソフトウェアやハードウェアで実装された暗号回路に対する様々な物理解析が報告されている。特に、サイドチャネル解析は比較的安価な機器で解析ができるため、非常に脅威とされている。

まず、3.2 節では暗号回路に対する物理解析について述べる。次に、3.3 節ではサイドチャネル解析の中でも消費電力や漏洩電磁波を利用した解析である電力・電磁波解析について説明する。そして、3.4 節では演算誤りを利用したフォールト解析について説明する。

3.2 暗号回路に対する物理解析

暗号回路に対する物理解析は、主に侵襲解析・半侵襲解析・非侵襲解析の 3 つに分類される [1]。各解析について、以下に示す。

3.2.1 侵襲解析

侵襲解析 [2]は、回路の物理的な改変を伴う解析である。まず、侵襲解析では特殊な薬品を用いることで、暗号回路のパッケージを開封する。そして、プロービング装置を使用し、直接内部の信号を読み取ることで秘密鍵を窃取する。他にも、レーザーカッターや集束イオンビーム (Focused Ion Beam: FIB) を利用した物理的な改変を加えることで、回路の解析を行う。

3.2.2 半侵襲解析

半侵襲解析 [3]は，侵襲解析と同様に暗号回路のパッケージを開封するが，侵襲解析のように回路に物理的な改変は加えない。具体的に，半侵襲解析では X 線や電磁波，光を照射することで，暗号回路に対して意図的に演算誤りを引き起こす。そして，この演算誤りを内部の秘密鍵の推定に利用する。半侵襲解析は侵襲解析ほど高価な装置を必要としないため，解析に必要なコストは侵襲解析よりも低い。

3.2.3 非侵襲解析

非侵襲解析はデバイスに対して解析の痕跡を残さない手法であり，回路から漏洩する情報を観測することで解析を行う。サイドチャネル解析は，非侵襲解析に分類され，比較的安価な装置で実施できるため非常に強力な解析である。また，本論文でのサイドチャネル解析の分類を図 3.1 に示す。図 3.1 に示すように，サイドチャネル解析は大きく電力・電磁波解析（3.3 節で説明する）とフォールト解析（3.4 節で説明する）に分類される。

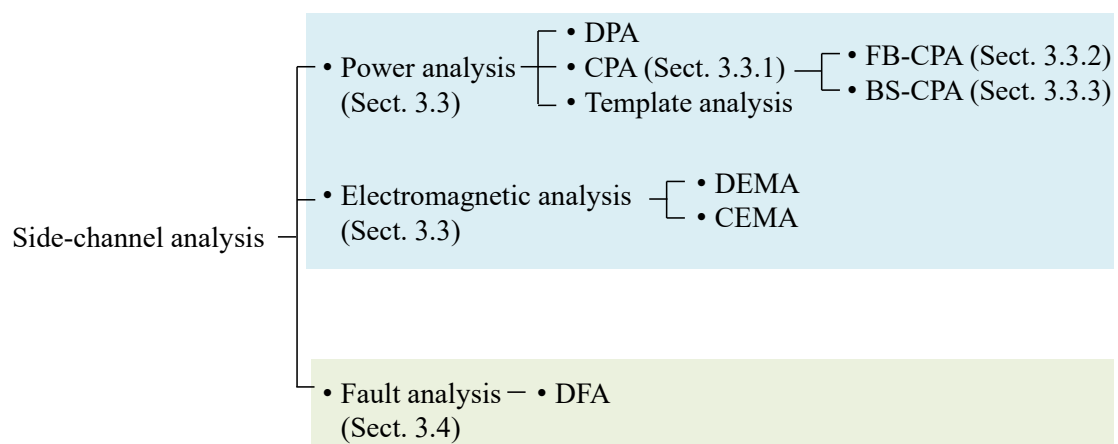


図 3.1 サイドチャネル解析の分類

Fig. 3.1 Classification of side-channel analysis

3.3 電力・電磁波解析

電力・電磁波解析では、暗号回路動作時に発生する消費電力や漏洩電磁波を利用した統計処理を行うことで、秘密鍵を推定する。消費電力を利用した解析を電力解析、漏洩電磁波を利用した解析を電磁波解析と呼ぶ。電力・電磁波解析は大きく分けて、ハミング重み型解析とハミング距離型解析の2種類に分類される。ハミング重み型解析は、暗号中間値のハミング重みによって暗号回路で発生する消費電力・漏洩電磁波に偏りが生じることに着目した解析手法である。ハミング距離型解析は、ループ型アーキテクチャで実装した暗号回路において、データレジスタ間のハミング距離によって、暗号回路で発生する消費電力に偏りが生じることを利用する解析手法である。

代表的な電力・電磁波解析には、差分電力解析 (Differential Power Analysis: DPA) [4]や相関電力解析 (Correlation Power Analysis: CPA) [5], 差分電磁波解析 (Differential ElectroMagnetic Analysis: DEMA) [6], 相関電磁波解析 (Correlation ElectroMagnetic Analysis: CEMA) [7], テンプレート解析 [8]–[10]などがある。特に、本論文で提案する電力・電磁波解析のベースとなる CPA について説明する。ここで CEMA に関しては、CPA で使用する消費電力の代わりに電磁波を使用することで、同様に解析を行うことができる。また、CPA・CEMA はハミング距離型解析の1つである。

3.3.1 CPA

CPA の概要について、AES 暗号を対象とする場合を例に図 3.2 を用いて説明する。CPA では、まず、データレジスタ間のハミング距離と消費電力との間に線形な関係が成り立つことを仮定する。この線形相関の例を図 3.3 に示す。図 3.3 は AES 暗号処理の 10R 目計算時の消費電力について、ある 8bit のレジスタに着目し、ハミング距離別の平均を計算したものである。図の横軸は時間を、縦軸は電圧値を示しており、消費電力として $1[\Omega]$ シャント抵抗の電圧降下をオシロスコープで測定している。図 3.3 に示すように、ハミング距離が増加するほど、消費電力が大きくなっており、ハミング距離と消費電力の間に線形な相関があることが確認できる。

CPA ではこの線形相関を解析に利用するために、データレジスタ間のハミング距離を計算する。このとき、図 3.2 に示すようにデータレジスタの値のうち、10R 計算終了後の値である暗号文は既知の値である。そのため、解析者にとって未知の値である 9R 計算終了後の値 (暗号中間値) を算出する。この計算では、既知の暗号文に対して `AddRoundKey`

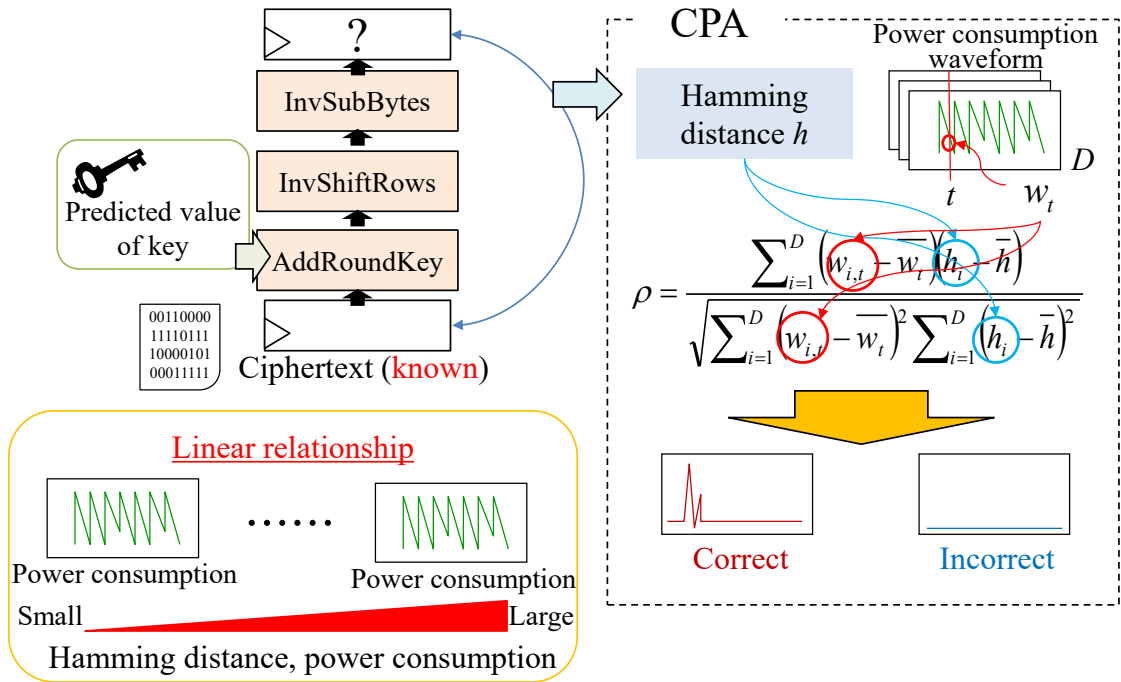


図 3.2 CPA の概要

Fig. 3.2 Outline of CPA

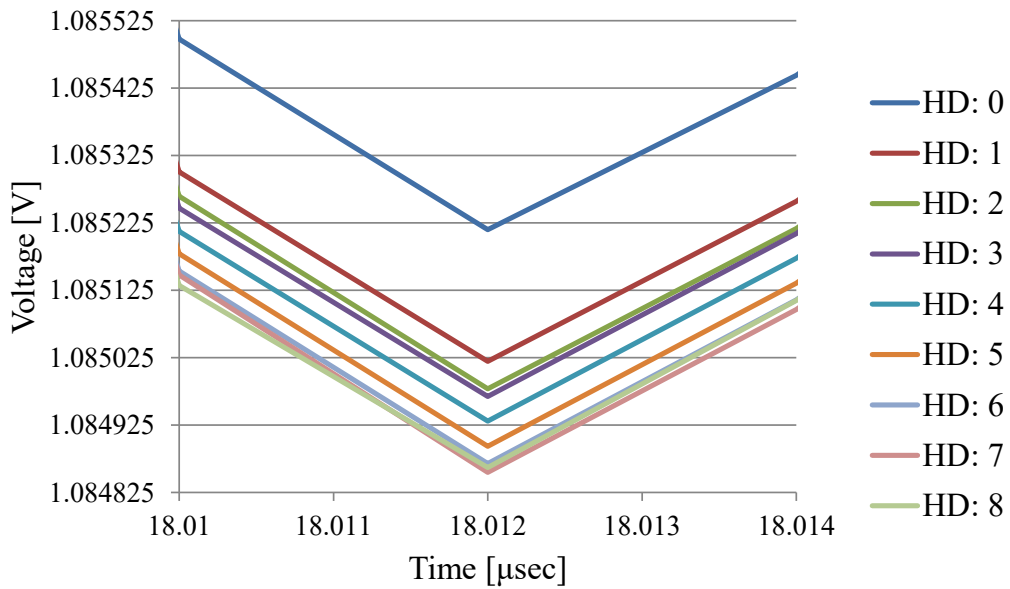


図 3.3 消費電力とハミング距離の線形な関係

Fig. 3.3 Linear relationship between power consumption and Hamming distance

やラウンド関数の各処理の逆関数、InvSubByte、InvShiftRows を適用する。ここで、AddRoundKey では 10R のラウンド鍵 RK^{10} の予測値を使用する。また、AES 暗号では 4×4 の State で 8bit 毎に処理を行っているため、8bit ずつ暗号中間値を計算することができる。そのため AddRoundKey では、128bit のラウンド鍵 RK^{10} を 8bit ずつの 16 個の部分鍵 RK_i^{10} ($0 \leq i \leq 15$) に分割し、この 8bit の部分鍵の予測値を使用する。すなわち、 $2^8 = 256$ 通りの鍵候補を全て試し、暗号中間値を導出する。そして、導出した暗号中間値と既知の暗号文を利用することで、データレジスタ間のハミング距離 h を求め、ハミング距離 h と消費電力 w とのピアソンの相関係数 ρ を計算する。この計算式を式(3.1)に示す。

$$\rho = \frac{\sum_{i=1}^D (w_{i,t} - \bar{w}_t)(h_i - \bar{h})}{\sqrt{\sum_{i=1}^D (w_{i,t} - \bar{w}_t)^2 \sum_{i=1}^D (h_i - \bar{h})^2}} \quad (3.1)$$

ここで、 t は消費電力波形の時間軸上のサンプル点、 $\bar{w}_t$ は消費電力 $w_{i,t}$ の平均値、 $\bar{h}$ はハミング距離 h の平均値、 D は計算に使用した消費電力波形の数をそれぞれ表している。このとき、鍵の予測値が正しい場合、ハミング距離と消費電力との間の線形な相関関係がピアソンの相関係数にピークとなって現れる。一方で、鍵の予測値が間違っている場合、相関は表われない。そのため、このピアソンの相関係数 ρ を最大とする鍵の予測値を正解鍵として推定する。この計算を残りの部分鍵に合わせて 15 回行うことで、AES の 10R のラウンド鍵 RK^{10} を全て求めることができる。

また、これまでに CPA の解析効率を高める方法として、3.3.2 項で説明する周波数領域での解析手法 (Frequency Based CPA: FB-CPA) [11]–[13]や、3.3.3 項で説明する解析済の部分鍵を組み込む解析手法 (Built-in determined Sub-key CPA: BS-CPA) [14][15]などが提案されている。

3.3.2 FB-CPA

電力解析では、複数の消費電力波形の任意の時間軸上のサンプル点を利用して解析を行うため、位相ずれ誤差は解析効率を低下させる要因となる。FB-CPA では、周波数領域で解析を行うことで、この時間軸方向での位相ずれ誤差の影響を抑え、解析効率を向上させる [11]–[13]。また、周波数領域で解析を行うことで、測定誤差による平均電圧の変動による解析への影響も抑えることができる [10]。FB-CPA では、解析対象の消費電力波形のサ

ンプル点に対して、離散フーリエ変換 (Discrete Fourier Transform: DFT) を適用し、パワースペクトル波形を算出する。そして、時間領域の消費電力波形の代わりに、算出したパワースペクトル波形を用いて CPA を行う。

また、周波数領域での電力解析は、第 4 章で提案する TWINE に対する電力解析手法や、第 7 章で提案する Minalpher に対する電力解析手法においても、解析効率を向上させるアプローチとして使用する。

3.3.3 BS-CPA

CPA はあるデータレジスタ間のハミング距離に着目して解析を行うため、着目しているハミング距離以外のデータ遷移で発生する消費電力情報は、解析を行う際にノイズ成分(アルゴリズムノイズとも呼ばれる)となる。そのため、CPA では多くの波形データを使用することで、このノイズ成分を平準化する。BS-CPA では、着目するハミング距離に付随して生じる消費電力(信号成分)を増やし、未知情報(ノイズ成分)を減らすことで、信号対ノイズ比 (Signal to Noise Ratio: SNR) を改善させる [14][15]。具体的に、BS-CPA では CPA での部分鍵の推定において、ある閾値を超えたものを推定済の部分鍵とみなす。そして、推定済の部分鍵を他の部分鍵の推定時に組み込む。この部分鍵の組み込みでは、推定済の部分鍵を利用して、通常ノイズ成分となるハミング距離を計算する。そして、新たに算出したハミング距離を相関係数の計算に利用することで、SNR を改善する。これまでに、BS-CPA は DES 暗号に対して提案 [14]されて以降、AES 暗号に対しても提案 [15]されており、その有効性が示されている。

また、組み込み処理によってノイズ成分の影響を軽減する考えは、第 4 章で提案する TWINE に対する電力解析手法や、第 5 章で提案する SIMECK に対する電磁波解析手法でも、解析効率を向上させるアプローチとして使用する。

3.4 フォールト解析

フォールト解析は、暗号回路に対して故意に演算誤り (以後、フォールトと呼ぶ) を発生させ、誤った暗号文 (以後、フォールト入り暗号文と呼ぶ) を出力させる。そして、フォールト入り暗号文と通常の暗号文を用いた解析を行うことで、秘密鍵を推定する。フォールト解析の研究は、解析手法に関する研究とフォールト混入手法に関する研究に大別される。

3.4.1 解析手法に関する研究

解析手法に関する研究に関して、フォールト解析は 1997 年に中国人剰余定理（Chinese Remainder Theorem: CRT）を用いた RSA 暗号を対象として初めて提案された [16]。その後、DES 暗号に対して差分フォールト解析（Differential Fault Analysis: DFA）[17]が提案され、AES 暗号に対しても暗号処理部を対象とした DFA [18][19]や鍵スケジュール部を対象とした DFA [20][21]など、数多く提案されている。DFA では、通常の暗号文とフォールト入り暗号文のペアを複数個用いて、秘密鍵を推定する。一方で、DFA は特定のタイミング（ラウンド）で、特定の位置に、特定の数のフォールトの混入を必要とする。

3.4.2 フォールト混入手法に関する研究

フォールト混入手法の研究に関しては、代表的な手法としてレーザーを照射する方法 [22]や、異常電圧を印加する方法 [23]、不正クロックを挿入する方法 [24][25]などがある。

レーザー照射 [22]は、解析対象のレイアウト情報を参照することで、フォールトを混入させるタイミングや位置などを精密に制御することができる。そのため、レーザー照射は強力なフォールト混入手法の 1 つである。一方で、レーザー照射では高価なレーザー装置や電子顕微鏡などを必要とするため、解析のためのコストは増加する。

一方で、異常電圧の印加 [23]や不正クロックの挿入 [24][25]は、安価な機器で実現可能である。特に、不正クロックを挿入する方法は、暗号回路に供給するクロックに、周期の短いクロックパルス（クロックグリッチ）を挿入するだけであり、容易に実現することができる。具体的に、クロックグリッチを挿入することで、フリップフロップに対してセットアップタイム制約違反を引き起こし、フォールトを混入させる。しかし、クロックグリッチは暗号回路全体、すなわち全てのフリップフロップに供給されるため、フォールトの混入位置やフォールトの混入数などを正確に制御することは難しい。クロックグリッチを用いたフォールト混入に関して、これまでに 1Byte のフォールトを必要とする AES 暗号の DFA [18]を行うための詳細な評価も行われている [26]。文献 [26]では、0.1 [nsec]単位でクロックグリッチを制御することで、ランダムな 10 種類の平文のうち 3 種類の平文で DFA が適用可能なことが示されている。一方で、クロックグリッチを 0.1[nsec]単位を制御できた場合でも、1Byte のフォールトではない複数のフォールトが混入し、DFA の適用困難な平文が存在することが指摘されている。

3.5 結言

本章では、まず暗号回路に対する物理解析について概説した。次に、強力な解析手法であるサイドチャネル解析について述べた。そして、電力・電磁波解析については、提案する電力・電磁波解析のベースとなる CPA を中心に説明した。また、フォールト解析については解析手法やフォールト混入手法について概説した。

参考文献

- [1] S. Mangard, E. Oswald, and T. Popp, “Power Analysis Attacks,” p.338, Springer US, 2007.
- [2] O. Kömmerling and M. G. Kuhn, “Design Principles for Tamper-Resistant Smartcard Processors,” Proc. USENIX Workshop on Smartcard Technology (Smartcard’99), pp. 9–20, May 1999.
- [3] S. P. Skorobogatov, “Semi-Invasive Attacks – A New Approach to Hardware Security Analysis,” PhD thesis, University of Cambridge, Apr. 2005.
- [4] P. Kocher, J. Jaffe, and B. Jun, “Differential Power Analysis,” Proc. CRYPTO’99, LNCS vol. 1666, pp. 388–397, Springer-Verlag, Dec. 1999.
- [5] E. Brier, C. Clavier, and F. Olivier, “Correlation Power Analysis with a Leakage Model,” Proc. 6th Int. Workshop on Cryptographic Hardware and Embedded Systems (CHES 2004), LNCS vol. 3156, pp. 16–29, Springer-Verlag, Aug. 2004.
- [6] K. Gandolfi, C. Mourtel, and F. Olivier, “Electromagnetic Analysis: Concrete Results,” Proc. 3rd Int. Workshop on Cryptographic Hardware and Embedded Systems (CHES 2001), LNCS vol. 2162, pp. 251–261, Springer-Verlag, May 2001.
- [7] O. Meynard, S. Guilley, J. -L. Danger, and L. Sauvage, “Far Correlation-based EMA with a Precharacterized Leakage Model,” Proc. Design, Automation and Test in Europe Conference and Exhibition (DATE 2010), pp. 977–980, Mar. 2010.
- [8] S. Chari, J. R. Rao, and P. Rohatgi, “Template Attacks,” Proc. 4th Int. Workshop on Cryptographic Hardware and Embedded Systems (CHES 2002), LNCS vol. 2523, pp. 13–28, Springer-Verlag, Aug. 2002.

- [9] 金 用大, 菅原 健, 本間尚文, 青木孝文, 佐藤 証, “AES のハードウェア実装に対するテンプレート攻撃,” 第 8 回情報科学技術フォーラム講演論文集, vol. 8, no.4, pp. 139–146, Aug. 2009.
- [10] 野崎佑典, 旭 健作, 藤野 毅, 吉川雅弥, “周波数領域における調節平文を用いたテンプレート攻撃とその評価,” 電気学会論文誌 C, vol. 134, no. 12, pp. 1775–1782, Dec. 2014.
- [11] C. H. Gebotys, S. Ho, and C. C. Tiu, “EM Analysis of Rijndael and ECC on a Wireless Java-Based PDA,” Proc. 7th Int. Workshop on Cryptographic Hardware and Embedded Systems (CHES 2005), LNCS vol. 3659, pp. 250–264, Springer-Verlag, Aug. 2005.
- [12] 菅原 健, 本間尚文, 林 優一, 水木敬明, 青木孝文, 曾根秀昭, 佐藤 証, “周波数領域での暗号モジュールの電力解析,” 第 8 回情報科学技術フォーラム講演論文集, vol. 8, no. 4, pp. 135–138, Aug. 2009.
- [13] 佐藤隆亮, 松島大祐, 汐崎 充, 藤野 毅, 吉川雅弥, “周波数領域における部分鍵推定を用いたハイブリッド電力解析攻撃とその評価,” 電気学会論文誌 C, vol. 133, no. 7, pp. 1322–1330, Jul. 2013.
- [14] Y. Komano, H. Shimizu, and S. Kawamura, “BS-CPA: Built-In Determined Sub-Key Correlation Power Analysis,” IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences, vol. E93-A, no. 9, pp. 1632–1638, Sep. 2010.
- [15] 駒野雄一, 清水秀夫, 野崎華恵, 新保 淳, 川村信一, “組込型相関電力解析 BS-CPA の解析効率の評価,” 2011 年暗号と情報セキュリティシンポジウム講演論文集, 3D4-4, pp. 1–8, Jan. 2011.
- [16] D. Boneh, R. A. DeMillo, and R. J. Lipton, “On the Importance of Checking Cryptographic Protocols for Faults,” Proc. 14th Annual EUROCRYPT Conf. (EUROCRYPT’97), LNCS vol. 1233, pp. 37–51, Springer-Verlag, May 1997.
- [17] E. Biham and A. Shamir, “Differential Fault Analysis of Secret Key Cryptosystems,” Proc. CRYPTO’97, LNCS vol. 1294, pp. 513–525, Springer-Verlag, Aug. 1997.
- [18] G. Piret, J. -J. Quisquater, “A Differential Fault Attack Technique against SPN Structures, with Application to the AES and Khazad,” Proc. 5th Int. Workshop on Cryptographic Hardware and Embedded Systems (CHES 2003), LNCS vol. 2779, pp. 77–88, Springer-Verlag, Sep. 2003.

- [19] A. Moradi, M. T. M. Shalmani, and M. Salmasizadeh, “A Generalized Method of Differential Fault Attack Against AES Cryptosystem,” Proc. 8th Int. Workshop on Cryptographic Hardware and Embedded Systems (CHES 2006), LNCS vol. 4249, pp. 91–100, Springer-Verlag, Oct. 2006.
- [20] C. -N. Chen and S. -M. Yen, “Differential Fault Analysis on AES Key Schedule and Some Countermeasures,” Proc. 8th Australasian Conf. Information Security and Privacy (ACISP 2003), LNCS vol. 2727, pp. 118–129, Springer-Verlag, Jul. 2003.
- [21] 高橋順子, 福永利徳, “AES の鍵スケジュール部に対するフォールト攻撃,” 2008 年暗号と情報セキュリティシンポジウム講演論文集, 2A1-2, pp. 1–6, Jan. 2008.
- [22] S. P. Skorobogatov and R. J. Anderson, “Optical Fault Induction Attacks,” Proc. 4th Int. Workshop on Cryptographic Hardware and Embedded Systems (CHES 2002), LNCS vol. 2523, pp. 2–12, Springer-Verlag, Aug. 2003.
- [23] N. Selmane, S. Bhasin, S. Guilley, T. Graba, and J. -L. Danger, “WDDL is Protected Against Setup Time Violation Attacks,” Proc. 6th Int. Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC 2009), pp. 73–83, Sep. 2009.
- [24] 福永利徳, 高橋順子, “暗号 LSI に対するフォールト解析攻撃実験,” 2009 年暗号と情報セキュリティシンポジウム講演論文集, 2A3-5, pp. 1–6, Jan. 2009.
- [25] S. Endo, T. Sugawara, N. Homma, T. Aoki and A. Satoh, “A Configurable On-Chip Glitchy-Clock Generator for Fault Injection Experiments,” IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences, vol. E95-A, no. 1, pp. 263–266, Jan. 2012.
- [26] 前島一仁, 五百旗頭健吾, 籠谷裕人, 渡辺哲史, 豊田啓孝, “差分フォールト解析においてクリティカルなグリッチ注入タイミングの測定,” 2015 年暗号と情報セキュリティシンポジウム講演論文集, 3A3-3, pp. 1–5, Jan. 2015.

第 4 章

TWINE に対する電力解析手法

4.1 緒言

本章では，TWINE の耐タンパ性評価に必要な電力解析手法を提案する [1]。TWINE [2] は，軽量暗号の 5 大要素の 1 つである商用利用されている暗号アルゴリズムであり，ICT プラットフォーム [3]や IoT 防犯カメラ [4]，IoT センサデバイス [5]などの実製品で利用されている。TWINE のアルゴリズムの詳細については，付録（A.1 節）で述べる。

電力解析の研究に関して，TWINE を対象としたものは見当たらないが，これまでに AES 暗号を対象としたものは数多く報告されている。AES 暗号に対する電力解析では，主に最終ラウンドのみを解析対象とする。なぜなら AES 暗号では，1 つのラウンドに用いる全ての部分鍵の合計ビット数は鍵長と等しいからである。そのため，1 つのラウンドの全ての部分鍵を解析することで，鍵スケジュール部の逆スケジューリング処理から，全ての秘密鍵を解析することができる。一方で，複数のラウンドを対象とする解析も報告されており，複数のラウンドの多くの消費電力情報を利用することで，解析効率を向上させている。例えば，文献 [6]で提案されている手法では，AES 暗号の 9 ラウンド目（9R）と 10R に対して CPA を行い，AES の鍵スケジュール部の特性を利用することで，効率的に 10R の部分鍵を解析している。他にも，解析効率を向上させるためのアプローチとして，3.3.2 項で述べた FB-CPA [7]–[9]や，3.3.3 項で述べた BS-CPA [10][11]が報告されている。

しかしながら，TWINE を対象とした電力解析の研究はこれまでに報告されていない。本章で提案する電力解析手法では，一般的な AES 暗号に対する電力解析とは異なり，複数の

ラウンドを対象とする。また、周波数領域での解析や部分鍵の組込み処理を組み合わせることで解析効率を向上させる。

4.2 提案する電力解析手法

提案する電力解析手法では、TWINE の 3 つのラウンドに対して CPA [12] を改良した解析を行う。これは、TWINE の鍵長は 80bit であるのに対して、1 つのラウンドの部分鍵は 32bit であるからである。具体的には、34 ラウンド目 (34R)、35R、36R (最終ラウンド) を対象とする。更に、以下の 2 つの処理を導入することで解析効率を向上させる。

- (1) 周波数領域での電力解析手法
- (2) 複数ラウンドに対する解析における部分鍵の組込み処理

まず、(1)については 3.3.2 項で述べた通り、電力解析に使用する消費電力波形における時間軸方向での位相ずれ誤差は、解析効率を低下させる原因となる。また、測定誤差による平均電圧の変動も解析効率を低下させる要因となる。そこで、周波数領域で電力解析を行うことで、これらの影響を抑え解析効率を向上させる。次に、(2)については、複数のラウンドに対して解析を行う際に、解析済の部分鍵を組込む処理を行うことで、解析効率を向上させる。

最初に 4.2.1 項で、ベースとなる時間領域での電力解析手法について説明する。次に 4.2.2 項では、電力解析手法の周波数領域での適用方法について説明する。最後に 4.2.3 項では、複数ラウンドに対する解析手法について説明する。

4.2.1 ベースとなる時間領域での電力解析手法

まず、TWINE に対する CPA をベースとした時間領域での電力解析手法について説明する。提案する電力解析手法では、3 つのラウンドの部分鍵を求めるため、34R、35R、36R (最終ラウンド) を対象に解析を行う。

ここでは、36R を対象とした場合を例に説明する。解析方法の概要を図 4.1 に示す。図 4.1 (i)に示すように、35R と 36R 計算終了後のデータレジスタ間のハミング距離を利用するものとする。このとき、35R の計算終了後の暗号中間値 x^{36} は未知、36R の計算終了後の値は暗号文であるため、既知である。そのため、図 4.1 (ii)に示すように x^{36} を既知の暗号文 c と部分鍵 RK_j^{36} の予測値を用いた逆算処理によって計算する。

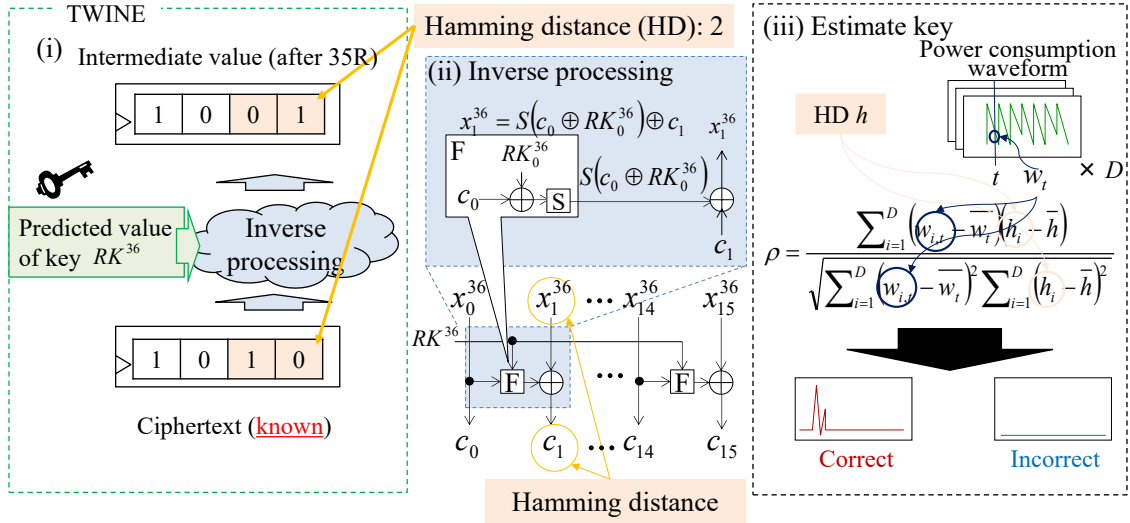


図 4.1 CPA をベースとした電力解析手法

Fig. 4.1 Power analysis method based on CPA

ここで、暗号中間値は各 4bit の 8 個の部分鍵を用いてそれぞれ計算する。すなわち、 x_{2j+1}^{36} ($0 \leq j \leq 7$) をそれぞれ求める。この計算は式(A.1)、(A.2)を用いて、式(4.1)で表すことができる。

$$x_{2j+1}^{36} = S(x_{2j}^{36} \oplus RK_j^{36}) \oplus c_{2j+1} \quad (4.1)$$

ここで、式(A.1)、(A.2)から、 $c_{2j} = x_{2j}^{36}$ であるため、式(4.1)は式(4.2)に示す計算式で表すことができる。

$$x_{2j+1}^{36} = S(c_{2j} \oplus RK_j^{36}) \oplus c_{2j+1} \quad (4.2)$$

このとき、部分鍵 RK_j^{36} は 4bit であるため、 $2^4 = 16$ 通りの鍵候補を試す。そして、暗号中間値 x_{2j+1}^{36} を求めた後、暗号文 c_{2j+1} とのハミング距離 h を計算する。ハミング距離を算出後、図 4.1 (iii)に示すように、式(3.1)を用いてハミング距離 h と消費電力 w からピアソンの相関係数 ρ を計算する。最後に、相関係数 ρ を最大とする鍵候補を正解鍵として推定する。

4.2.2 周波数領域での電力解析手法

提案する電力解析手法では、4.2.1 項で説明した電力解析手法を周波数領域で適用することで、解析効率を向上させる。周波数領域での電力解析手法の概要を図 4.2 に示す。図 4.2

は 36R を対象とした場合を示したものである。図 4.2 に示すように、周波数領域での電力解析では導出したハミング距離とパワースペクトル波形を利用する。このパワースペクトル波形は、取得した消費電力波形を DFT して作成したものを利用する。周波数領域での電力解析で使用するパワースペクトル波形の算出方法を図 4.3 に示す。図 4.3 に示すように、時間領域での電力解析で解析対象となる消費電力波形のサンプル点を DFT することで、パワースペクトル波形を算出する。このパワースペクトル波形は、消費電力値を DFT して取得した振幅スペクトルの二乗である両側パワースペクトルを、ナイキスト周波数で折り返した片側パワースペクトルである。

また、時間領域での電力解析では、消費電力とハミング距離を利用するのに対して、周波数領域での電力解析では、パワースペクトルとハミング距離を利用する。すなわち、パワースペクトルとハミング距離のピアソンの相関係数を計算する。このピアソンの相関係

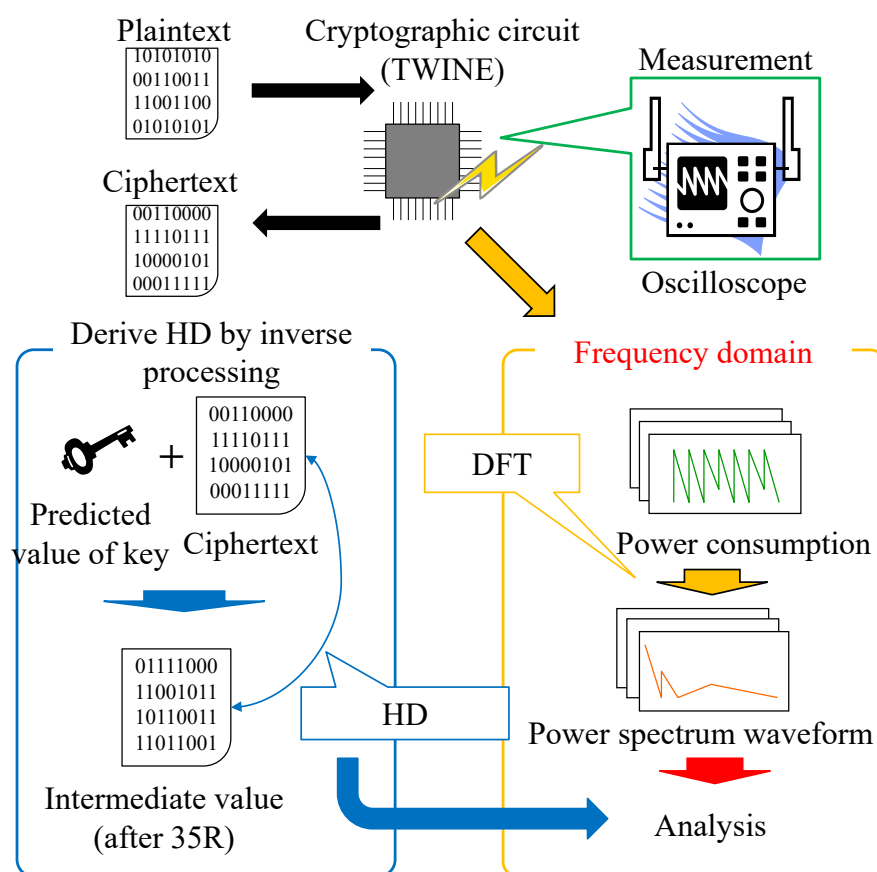


図 4.2 周波数領域での電力解析手法

Fig. 4.2 Power analysis method in frequency domain

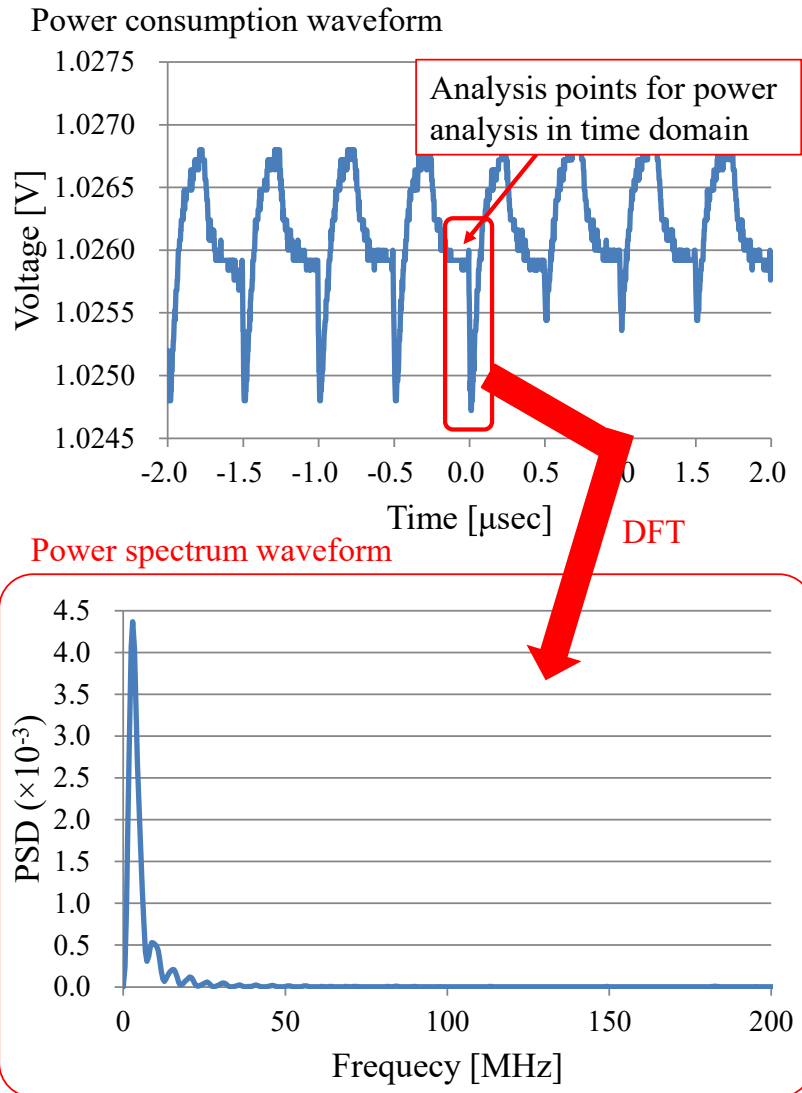


図 4.3 パワースペクトル波形の生成方法

Fig. 4.3 Power spectrum generation method

数の計算では、式(3.1)において使用していた消費電力 w の代わりに、パワースペクトル s を使用する。すなわち、式(4.3)で計算する。

$$\rho = \frac{\sum_{i=1}^D (s_{i,f} - \bar{s}_f)(h_i - \bar{h})}{\sqrt{\sum_{i=1}^D (s_{i,f} - \bar{s}_f)^2 \sum_{i=1}^D (h_i - \bar{h})^2}} \quad (4.3)$$

ここで、 f は任意の周波数帯域、 $\overline{s_f}$ はパワースペクトル s の平均値である。そして、提案する電力解析手法では、式(4.3)から得られたピアソンの相関係数 ρ を最大とする鍵の予測値を正解鍵として推定する。

4.2.3 複数ラウンドに対する解析手法

TWINE の全ての秘密鍵を解析するためには、鍵スケジュール部での 80bit 分の部分鍵(以後、必要部分鍵と呼ぶ)を求める必要がある。そして、この 80bit の必要部分鍵に対して、鍵スケジュール部の逆スケジューリング処理を適用することで、全ての秘密鍵を特定することができる。ここで、36R と 35R の暗号処理で使用する部分鍵 (RK^{36} , RK^{35}) を生成する鍵スケジュール部を図 4.4 に示す。図 4.4 に示す必要部分鍵を求めることで、全ての秘密鍵を特定することができる。

図 4.4 に示すように、必要部分鍵は $RK_2^{36} \oplus S(RK_7^{35})$, RK_0^{36} , RK_3^{35} , RK_1^{36} , RK_2^{36} , RK_4^{34} , RK_3^{36} , RK_6^{34} , RK_7^{34} , RK_4^{35} , RK_5^{35} , RK_6^{35} , RK_7^{35} , RK_4^{36} , RK_5^{36} , RK_6^{36} , RK_7^{36} , RK_3^{34} , RK_1^{35} , $S^{-1}(RK_0^{35} \oplus RK_7^{36})$ の合計 20 個 (80bit) である。これらの必要部分鍵のうち、 RK_0^{36}

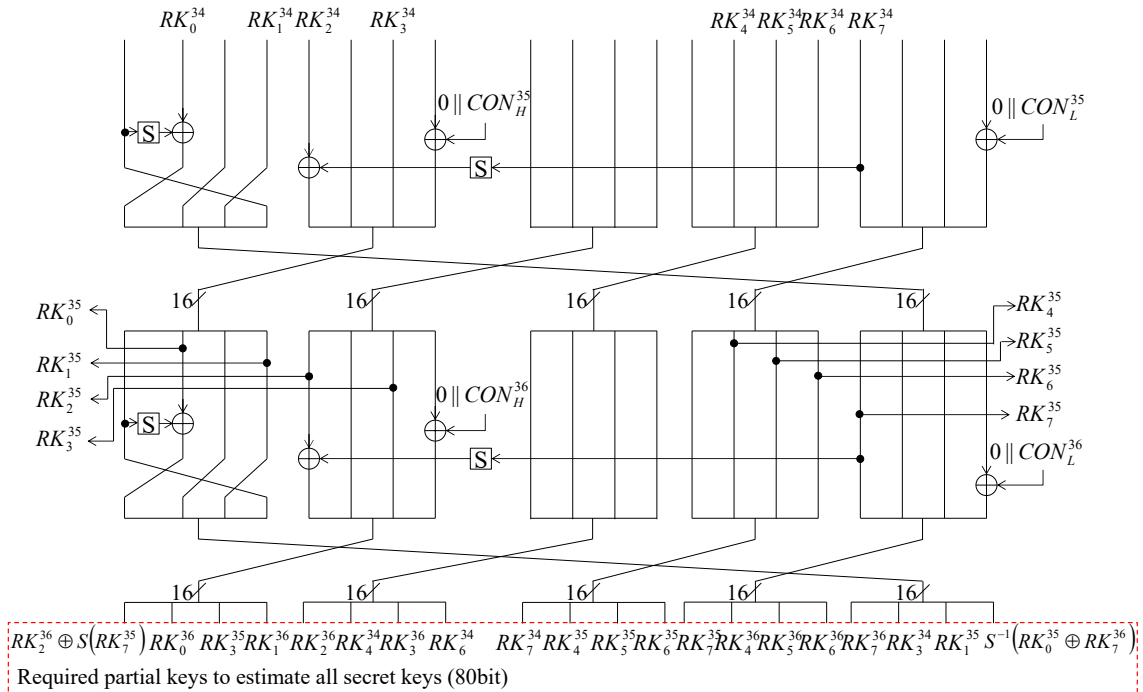


図 4.4 秘密鍵の解析に必要な部分鍵

Fig. 4.4 Keys required analysis for secret key

から RK_7^{36} の 8 個の部分鍵 (32bit) は、4.2.1 項、4.2.2 項の 36R に対する電力解析から求めることができる。20 個の必要部分鍵を求めるためには、残りの 12 個の必要部分鍵 (48bit) を求める必要がある。そこで、提案する電力解析手法では、複数のラウンド (34R と 35R) に対して電力解析を行うことで、残りの 12 個の必要部分鍵 (48bit) を推定する。

35R に対する電力解析では、35R の部分鍵 (RK_0^{35} から RK_7^{35} の 8 個の部分鍵) を対象とする。35R の部分鍵 (RK_0^{35} から RK_7^{35}) を求めることで、 $RK_2^{36} \oplus S(RK_7^{35})$, RK_3^{35} , RK_4^{35} , RK_5^{35} , RK_6^{35} , RK_7^{35} , RK_1^{35} , $S^{-1}(RK_0^{35} \oplus RK_7^{36})$ の 8 個の必要部分鍵を求めることができる。この解析では、まず、36R に対する電力解析によって、既知となる部分鍵 (RK_0^{36} から RK_7^{36} の 8 個の部分鍵) を使用して、35R 計算終了後の暗号中間値 x_{2j}^{36} を求める。そして、この暗号中間値 x_{2j}^{36} を利用して、35R に対して周波数領域での電力解析を行う。このとき、使用するパワースペクトル波形は、消費電力波形のうち 35R の処理を行っている範囲を対象に DFT することで取得する。また、提案する電力解析手法では、解析効率を向上させるために、部分鍵の組込み処理を行う。この部分鍵の組込み処理を図 4.5 に示す。

部分鍵の組込み処理では、36R の電力解析で既知となる部分鍵 (RK_0^{36} から RK_7^{36} の 8 個の部分鍵) を利用する。ここで、3.3.3 項でも述べた通り、電力解析ではあるデータレジスタ間のハミング距離に注目して解析を行う際、注目するハミング距離に関連して発生する消費電力は信号成分、注目外のデータ遷移に関連して発生する消費電力はノイズ成分となる [10][11]。すなわち、 x_1^{35} と x_1^{36} のハミング距離 (4bit) に注目した場合、それ以外の 60bit 分のデータ遷移 (x_0^{35} と x_0^{36} のハミング距離と、 x_2^{35} と x_2^{36} のハミング距離から x_7^{35} と x_7^{36} のハミング距離) はノイズ成分となる。また、暗号中間値 x^{35} と x^{36} には、式(A.1), (A.2)によって式(4.4)に示す関係が成り立つ。

$$\begin{aligned} x_{\pi(2j)}^{36} &= x_{2j}^{35} \\ x_{\pi(2j+1)}^{36} &= S(x_{2j}^{35} \oplus RK_j^{35}) \oplus x_{2j+1}^{35} \end{aligned} \quad (4.4)$$

このとき、 $\pi()$ は拡散層によって置換されたビット位置を表しており、表 A.3 に示す通り、奇数番目の位置の値 ($2j+1$ ($j=0, 1, \dots, 7$)) は、必ず偶数番目の位置に置換される。そのため、34R 計算終了後の暗号中間値の偶数番目の値は、35R 計算終了後の暗号中間値の奇数番目の値で表すことができる。したがって、図 A.2 に示すように、36R の部分鍵 (RK_0^{36} から RK_7^{36} の 8 個の部分鍵) が既知であることを利用して、 x_{2j}^{35} と x_{2j+1}^{36} のハミング距離 (図 4.5 の例では、 x_0^{35} と x_0^{36} , x_2^{35} と x_2^{36} , x_4^{35} と x_4^{36} のハミング距離) を求めることができる。す

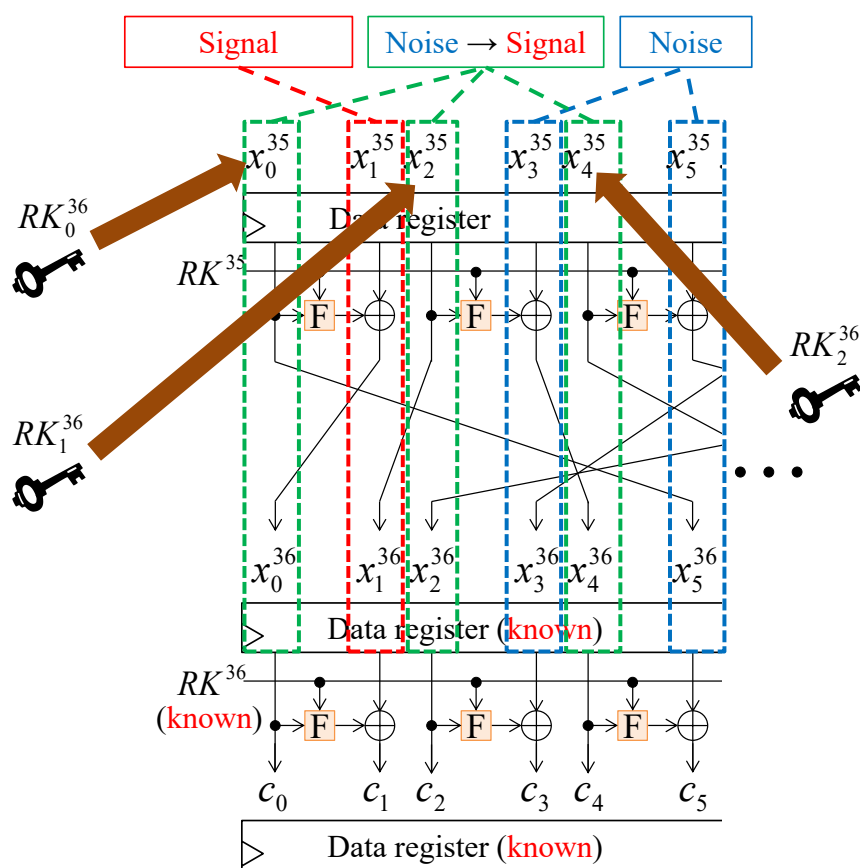


図 4.5 部分鍵の組込み処理

Fig. 4.5 Built-in processing

なわち, 8 個の部分鍵 ($8 \times 4 = 32\text{bit}$) を組込むことでデータレジスタ間のハミング距離 32bit が既知となり, 32bit 分のノイズ成分が 32bit 分の信号成分に変化するため, ノイズ成分を削減できる。また, この部分鍵の組込み処理では, 3.3.3 項で説明した BS-CPA [10][11] のように, あるラウンドで特定した部分鍵を, 同じラウンドの別の部分鍵の解析に組込む処理は行わない。

34R に対する電力解析では, 残りの 4 個の必要部分鍵 RK_3^{34} , RK_4^{34} , RK_6^{34} , RK_7^{34} を対象とする。具体的には, 35R に対する電力解析によって, 既知となる部分鍵 (RK_0^{35} から RK_7^{35} の 8 個の部分鍵) を使用して, 34R に対して周波数領域での電力解析を行う。このとき, 使用するパワースペクトル波形は, 消費電力波形のうち 34R の処理を行っている範囲を対象として, DFT することで取得する。

また, 35R に対する電力解析と同様に, 34R に対する電力解析においても, 部分鍵の組込み処理を行う。34R に対する部分鍵の組込み処理では, 既知の 35R の部分鍵を組込むだ

けでなく, 鍵スケジュール部の性質上, 既知となる 34R の別の部分鍵の組込み処理も行う。具体的には, 図 4.4 の鍵スケジュール部と式(4.5)に示すように, RK_0^{34} , RK_1^{34} , RK_5^{34} が推定済の部分鍵 RK^{36} , RK^{35} によって既知となることを利用する。

$$\begin{aligned} RK_0^{34} &= S(RK_6^{36} \oplus (0 \parallel CON_L^{35})) \oplus RK_7^{35} \\ RK_1^{34} &= RK_5^{36} \\ RK_5^{34} &= RK_3^{36} \end{aligned} \quad (4.5)$$

したがって, 35R の部分鍵 (RK_0^{35} から RK_7^{35} の 8 個の部分鍵) と, 34R の部分鍵 (RK_0^{34} , RK_1^{34} , RK_5^{34} の 3 個の部分鍵) の合計 11 個の部分鍵の組込み処理を行う。すなわち, 1 つの部分鍵は 4bit であるため, $11 \times 4 = 44\text{bit}$ 分のノイズ成分を削減できる。

4.3 評価実験

4.3.1 実験環境

実験環境を図 4.6 と表 4.1 に示す。実験では, 鍵長が 80bit(必要部分鍵が 20 個)の TWINE

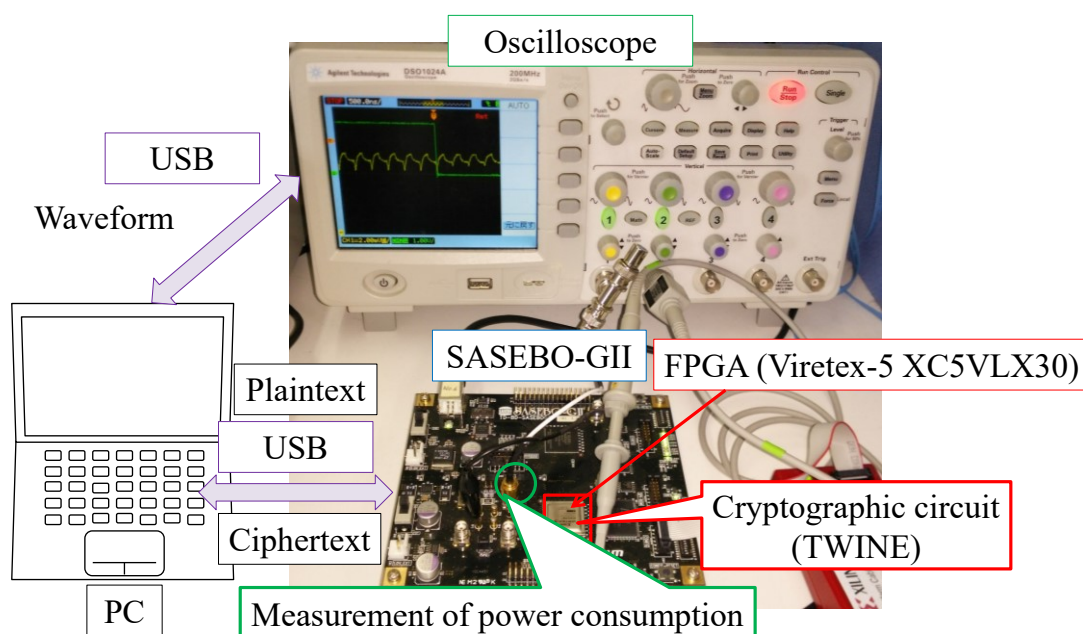


図 4.6 評価システム

Fig. 4.6 Evaluation system

表 4.1 実験環境

Table 4.1 Experimental condition

Encryption algorithm	TWINE
Key size [bit]	80
Evaluation board	SASEBO-GII [13]
FPGA	Virtex-5 XC5VLX30
Implementation tool	Xilinx ISE Design Suite 14.1
Oscilloscope	Agilent DSO 1024A
Sampling rate [Gsa/sec]	2

を SASEBO-GII [13] 上の FPGA に実装した。また, TWINE の S-BOX は合成体方式 [1][14][15] で実装した。そして, TWINE の暗号処理を 2,000 回行い, 2,000 個の暗号文と消費電力波形を取得した。消費電力波形は, 暗号回路の消費電流をシャント抵抗で測定した電圧波形である。

4.3.2 実験結果

評価実験では, まず, 36R (最終ラウンド) を対象に提案する電力解析を行った。この解析では, 20 個の必要部分鍵のうち 8 個の部分鍵を求める。ここで周波数領域での電力解

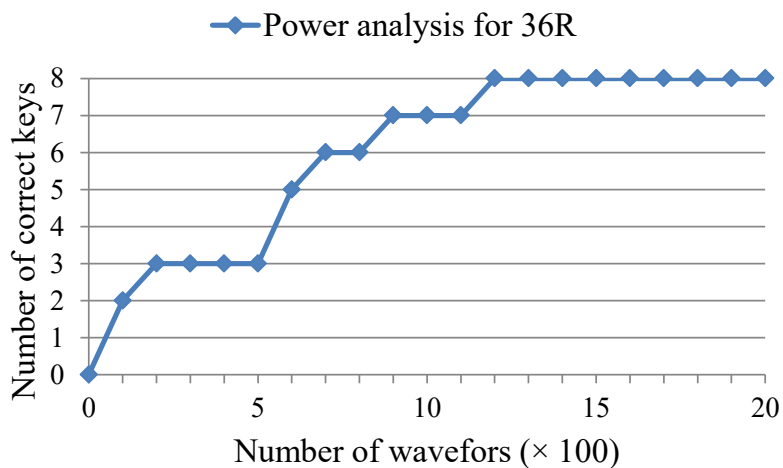


図 4.7 実験結果 (36 ラウンド目)

Fig. 4.7 Experimental result in the 36th round

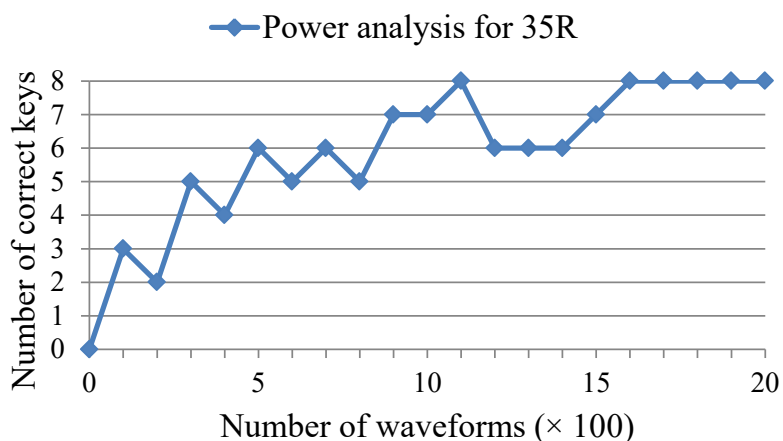


図 4.8 実験結果 (35 ラウンド目)

Fig. 4.8 Experimental result in the 35th round

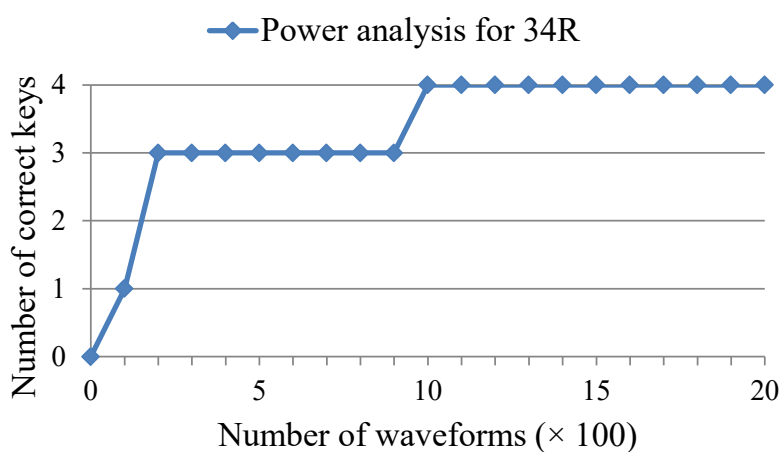


図 4.9 実験結果 (34 ラウンド目)

Fig. 4.9 Experimental result in the 34th round

析では、低周波数帯域を使用することで、解析効率が向上することが知られている [8]。そこで本実験では、予備実験（詳細は 4.3.3 項で説明する）で最も高い相関が観測された低周波数帯域（3–10 [MHz]）を使用した。実験結果を図 4.7 に示す。図の横軸は解析に使用した消費電力波形の数を、縦軸は解析に成功した部分鍵の数を示している。図 4.7 に示すように、1,200 波形を用いることで全ての部分鍵の解析に成功した。したがって、提案する電力解析手法が有効であることが分かる。

次に、35R に対して電力解析を行った。この解析では、20 個の必要部分鍵のうち 8 個の部分鍵を求める。また解析では、36R に対する電力解析で特定した部分鍵を組み込んだ処理を行う。実験結果を図 4.8 に示す。図 4.8 に示すように、1,600 波形を用いることで全ての部分鍵の特定に成功していることが確認できる。

最後に、34R に対して電力解析を行った結果を図 4.9 に示す。この解析では、20 個の必要部分鍵のうち 4 個の部分鍵を求める。また、35R に対する解析で特定した部分鍵を組み込んだ処理を行う。図 4.9 に示すように、1,000 波形を用いることで全ての部分鍵の解析に成功していることが確認できる。

以上より、TWINE の必要部分鍵 20 個 (80bit) を全て特定することに成功した。この必要部分鍵 80bit から鍵スケジュール部の逆スケジューリング処理によって、秘密鍵 80bit を特定することができる。したがって、提案する電力解析手法によって全ての秘密鍵を推定することができ、TWINE の耐タンパ性を評価できることが分かる。

4.3.3 解析に優位な周波数帯域の検証

周波数領域での電力解析において、解析に有意な周波数帯域について検証した。まず、36R に対する電力解析での正解鍵と不正解鍵の相関係数を比較した。また、この計算では

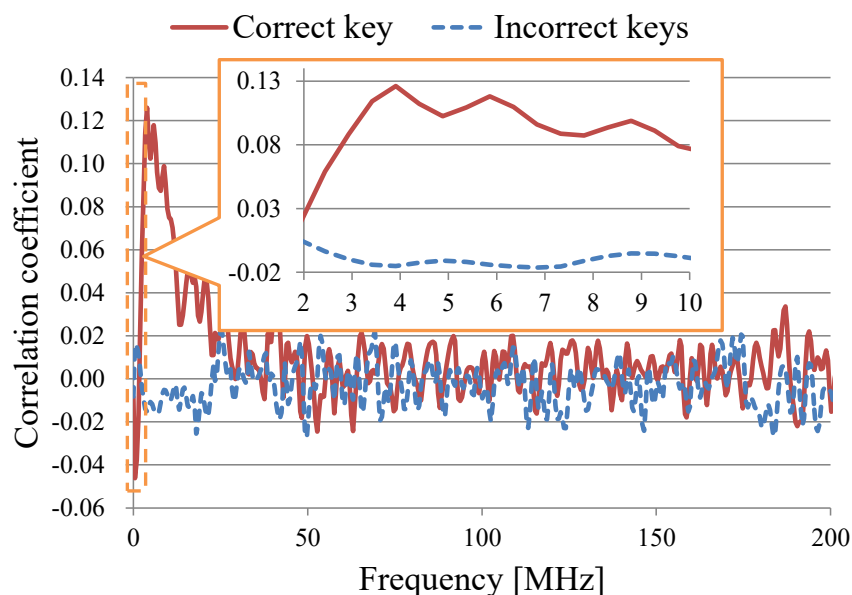


図 4.10 正解鍵と不正解鍵における相関係数の比較

Fig. 4.10 Comparison of correlation coefficient

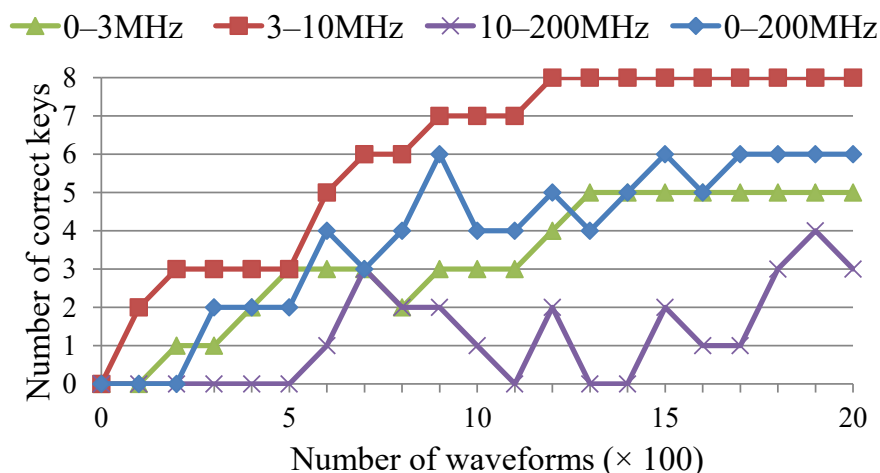


図 4.11 使用する周波数帯域の違いによる比較結果

Fig. 4.11 Comparison results of different frequency bands

10,000 波形のデータを使用した。比較結果を図 4.10 に示す。図の横軸は周波数帯域を、縦軸は相関係数を、赤色は正解鍵の結果を、青色は不正解鍵での結果をそれぞれ示している。図 4.10 の結果から正解鍵において、3-10[MHz]の低周波数帯域で高い相関が表れていることが確認できる。したがって、3-10[MHz]の周波数帯域を使用することで高効率な解析が期待される。

また、解析に使用する周波数帯域を 0-3 [MHz], 3-10 [MHz], 10-200 [MHz], 0-200 [MHz] と変化した場合の解析効率の違いを検証した。この結果を図 4.11 に示す。図 4.11 の結果から、3-10 [MHz]を用いた解析では、1,200 波形で全ての部分鍵の推定に成功している。一方で、3-10 [MHz]以外の周波数帯域では、2,000 波形を用いた場合でも、全ての部分鍵を解析することができなかった。したがって、3-10 [MHz]の周波数帯域を使用することで、最も高効率な解析が可能だと考えられる。

4.3.4 周波数領域での解析効率の検証

周波数領域の電力解析の解析効率への影響について検証した。この検証では、36R を対象とした解析において、電力解析を時間領域で行う場合と周波数領域で行う場合の実験結果を比較した。比較結果を図 4.12 に示す。図 4.12 に示すように、時間領域での電力解析では、2,000 波形を用いた解析で推定できた部分鍵が 3 個であるのに対して、周波数領域での電力解析では、1,200 波形で全ての部分鍵の推定に成功した。したがって、周波数領域に展

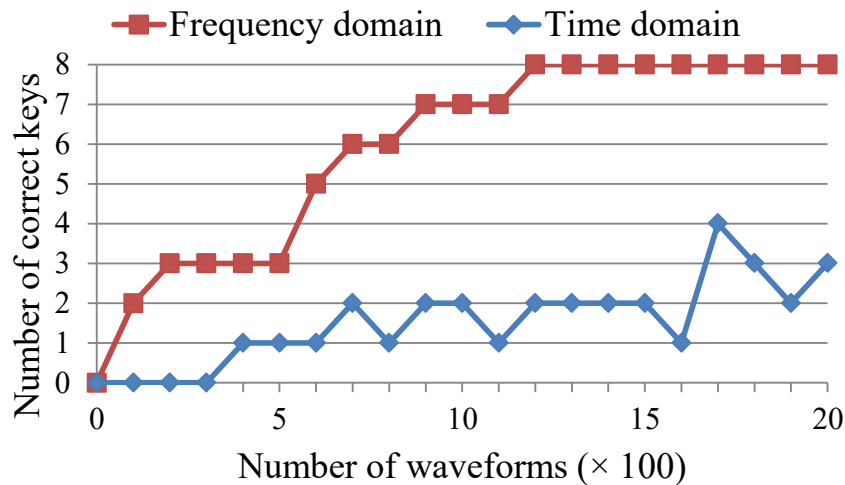


図 4.12 周波数領域と時間領域での解析効率の比較

Fig. 4.12 Comparison between frequency domain and time domain

開することで解析効率が向上しており，周波数領域での電力解析が有効であることが分かる。

4.3.5 部分鍵の組込み処理の解析効率への影響

34R と 35R に対する電力解析における，部分鍵の組込み処理の解析効率への影響について検証した。部分鍵の組込み処理を行う場合と行わない場合について，35R の解析結果と 34R の解析結果をそれぞれ図 4.13 と図 4.14 に示す。図 4.13 と図 4.14 は，周波数領域の電力解析で部分鍵の組込み処理を行う場合と行わない場合，時間領域の電力解析で部分鍵の組込み処理を行う場合と行わない場合の解析結果をそれぞれ示している。

図 4.13 に示すように，周波数領域での電力解析では，部分鍵の組込み処理を行う場合，1,600 波形で 8 個全ての部分鍵の解析に成功している。一方で，部分鍵の組込み処理を行わない場合，2,000 波形を用いた場合でも，全ての部分鍵を解析することはできなかった。また，時間領域での電力解析では，部分鍵の組込み処理を行う場合，2,000 波形で 8 個中 4 個の部分鍵の解析に成功した。一方で，部分鍵の組込み処理を行わない場合，2,000 波形で 8 個中 3 個の部分鍵の解析に成功した。したがって，周波数領域と時間領域の電力解析のどちらの場合においても，部分鍵の組込み処理を導入することで解析効率が向上しており，提案する部分鍵の組込み処理が有効であることが分かる。

次に，34R に対する解析結果について述べる。図 4.14 の結果から，周波数領域での電力

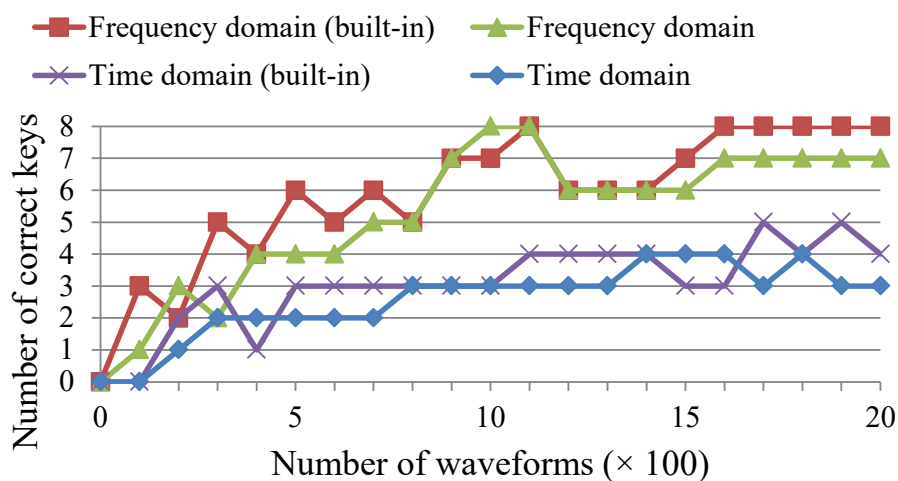


図 4.13 部分鍵の組込み処理による解析効率の違い（35 ラウンド目）

Fig. 4.13 Comparison with and without built-in processing at the 35th round

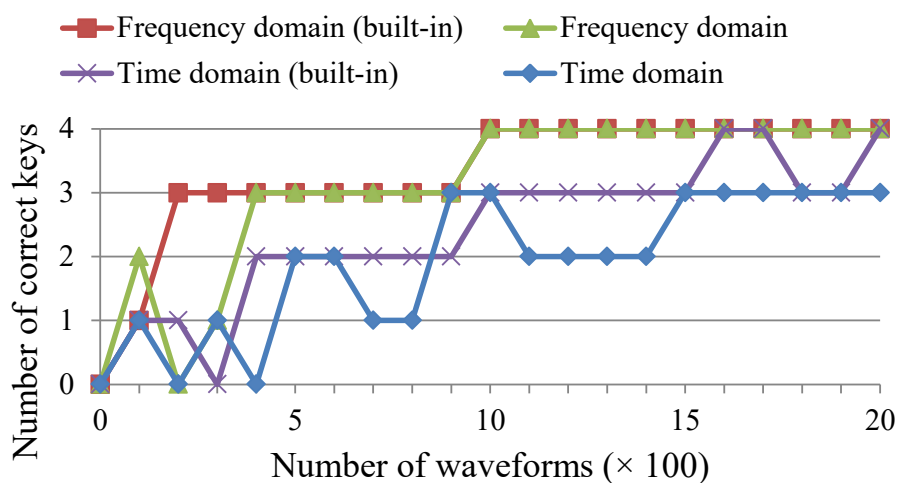


図 4.14 部分鍵の組込み処理による解析効率の違い（34 ラウンド目）

Fig. 4.14 Comparison with and without built-in processing at the 34th round

解析では、部分鍵の組込み処理を行う場合と行わない場合のどちらにおいても、1,000 波形で 4 個全ての部分鍵の解析に成功した。このとき、部分鍵の組込み処理を行う場合では、行わない場合と比較して、200 波形少ない波形数で 3 個の部分鍵の解析に成功していることが確認できる。時間領域での電力解析では、部分鍵の組込み処理を行う場合、2,000 波形

で全ての部分鍵の解析に成功した。一方で、部分鍵の組込み処理を行わない場合では、2,000 波形を使用した場合でも、全ての部分鍵を解析することができなかった。したがって、部分鍵の組込み処理を行うことで解析効率が向上しており、提案する部分鍵の組込み処理が有効であると考えられる。

4.4 結言

本章では、TWINE に対する電力解析手法を提案し、FPGA を用いた評価実験によってその有効性を示した。提案する電力解析手法では、TWINE のデータレジスタ間のハミング距離と消費電力との線形相関に着目する。また、周波数領域で解析を適用することで、時間軸方向での位相ずれ誤差の影響や、測定誤差による平均電圧の変動による解析効率への影響を抑え、高効率な解析を実現させた。更に、複数のラウンドに対する電力解析では、解析済の部分鍵を組込むことで、ノイズ成分を削減し、解析効率を向上させた。

以上より、提案した電力解析手法を用いることで、TWINE の耐タンパ性を評価できることを示した。

参考文献

- [1] 野崎佑典, 藤野 毅, 吉川雅弥, “軽量暗号 TWINE に対する周波数領域での電力解析とその評価,” 電子情報通信学会論文誌 B, vol. J99-B, no. 10, pp. 881–892, Oct. 2016.
- [2] T. Suzaki, K. Minematsu, S. Morioka, and E. Kobayashi, “TWINE: A Lightweight, Versatile Blockcipher,” Proc. ECRYPT Workshop on Lightweight Cryptography (LC11), pp. 146–149, Nov. 2011.
- [3] “NEC、IoT ソリューションを支える ICT プラットフォーム製品と体制を強化,” http://jpn.nec.com/press/201511/20151109_01.html
- [4] “セブン-イレブン、ALSOK・NEC と連携し、クラウド型の防犯カメラサービスを導入,” http://jpn.nec.com/press/201604/20160421_02.html
- [5] “NEC プラットフォームズ セキュアセンサ BOX,” <http://www.necplatforms.co.jp/product/securesensor/spec.html>

- [6] 中津大介, 太田和夫, 崎山一男, “AES-128 に対する複数ラウンド CPA,” 2011 年暗号と情報セキュリティシンポジウム講演論文集, 3D4-2, pp. 1–6, Jan. 2011.
- [7] C. H. Gebotys, S. Ho, and C. C. Tiu, “EM Analysis of Rijndael and ECC on a Wireless Java-Based PDA,” Proc. 7th Int. Workshop on Cryptographic Hardware and Embedded Systems (CHES 2005), LNCS vol. 3659, pp. 250–264, Springer-Verlag, Aug. 2005.
- [8] 菅原 健, 本間尚文, 林 優一, 水木敬明, 青木孝文, 曾根秀昭, 佐藤 証, “周波数領域での暗号モジュールの電力解析,” 第 8 回情報科学技術フォーラム講演論文集, vol. 8, no. 4, pp. 135–138, Aug. 2009.
- [9] 佐藤隆亮, 松島大祐, 汐崎 充, 藤野 毅, 吉川雅弥, “周波数領域における部分鍵推定を用いたハイブリッド電力解析攻撃とその評価,” 電気学会論文誌 C, vol. 133, no. 7, pp. 1322–1330, Jul. 2013.
- [10] Y. Komano, H. Shimizu, and S. Kawamura, “BS-CPA: Built-In Determined Sub-Key Correlation Power Analysis,” IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences, vol. E93-A, no. 9, pp. 1632–1638, Sep. 2010.
- [11] 駒野雄一, 清水秀夫, 野崎華恵, 新保 淳, 川村信一, “組込型相関電力解析 BS-CPA の解析効率の評価,” 2011 年暗号と情報セキュリティシンポジウム講演論文集, 3D4-4, pp. 1–8, Jan. 2011.
- [12] E. Brier, C. Clavier, and F. Olivier, “Correlation Power Analysis with a Leakage Model,” Proc. 6th Int. Workshop on Cryptographic Hardware and Embedded Systems (CHES 2004), LNCS vol. 3156, pp. 16–29, Springer-Verlag, Aug. 2004.
- [13] “The National Institute of Advanced Industrial Science and Technology Research Institute for Secure Systems: Evaluation Environment for Side-channel Attacks,”
<http://www.risec.aist.go.jp/project/sasebo>
- [14] A. Satoh, S. Morioka, K. Takano, and S. Munetoh, “A Compact Rijndael Hardware Architecture with S-Box Optimization,” Proc. ASIACRYPT 2001, LNCS vol. 2248, pp. 239–254, Springer-Verlag, Dec. 2001.
- [15] 森岡澄夫, 佐藤 証, 高野光司, “ $GF(((2^2)^2)^2)$ 上の演算を用いた AES の S-Box 構成法,” 情報処理学会第 63 回全国大会講演論文集, 3G-04, Sep. 2001.

第 5 章

SIMECK に対する電磁波解析手法

5.1 緒言

本章では、SIMECK の耐タンパ性評価に必要な電磁波解析手法を提案する [1]。SIMECK [2]は、軽量暗号の 5 大要素の 1 つである小回路規模実装が可能であり、これまで報告されているものの中で最も小回路規模でハードウェア実装できる。そのため、小回路規模での実装が求められる RFID タグなどでの利用が期待されている。SIMECK のアルゴリズムの詳細については、付録 (A.2 節) で述べる。

電磁波解析の研究に関して、これまでに SIMECK を対象とした手法は報告されていない。本章で提案する電磁波解析手法では、CEMA [3]を改良した解析を行う。また、SIMECK の複数のラウンド情報や既知情報を利用することで、解析効率を向上させる。

5.2 提案する電磁波解析手法

提案する電磁波解析手法では、CEMA [3]を改良した解析を行う。また、SIMECK の構造に着目した複数ラウンドを利用した解析を導入することで、解析効率を向上させる。更に、既知情報の組み込み処理を行うことで、ノイズ成分の影響を抑え、高効率な解析を実現させる。

まず 5.2.1 項では、ベース電磁波解析手法について説明する。次に 5.2.2 項では、複数ラウンドを利用した解析について説明する。最後に 5.2.3 項では、既知情報の組み込み処理について説明する。

本章は[1]で発表された内容から構成される。

5.2.1 ベース電磁波解析手法

提案する電磁波解析手法では、SIMECK のデータレジスタ間のハミング距離と、電磁波との間に線形相関があることを仮定し、この関係を解析に利用する。ベース電磁波解析手法について、SIMECK の最終ラウンド ($T-1$ ラウンド目) を対象とした場合を例に図 5.1 を用いて説明する。

ベース電磁波解析手法では、SIMECK の右半分の値 R^{T-1} と R^T のハミング距離を利用するものとする。このとき、 R^T は既知の暗号文であり、 R^{T-1} は未知の値である。そして、暗号中間値 R^{T-1} は式(5.1)で求めることができる。

$$R^{T-1} = L^T \oplus f(R^T) \oplus RK^{T-1} \quad (5.1)$$

また、式(5.1)の計算例を図 5.2 に示す。図 5.2 はブロック長が 32bit (ワード長 $m = 16$, ラウンド数 $T = 33$) の場合を例に示したものである。図 5.2 に示すように、対象とする暗号中間値 (図 5.2 の例では R^{32}) は既知の暗号文 (図 5.2 の例では L^{33} , R^{33}) とラウンド鍵 (図 5.2 の例では RK^{32}) の予測値 (未知) を用いて計算する。また、SIMECK はビット単位で処理を行っているため、図 5.2 に示すように、この計算はビット単位で行うことがで

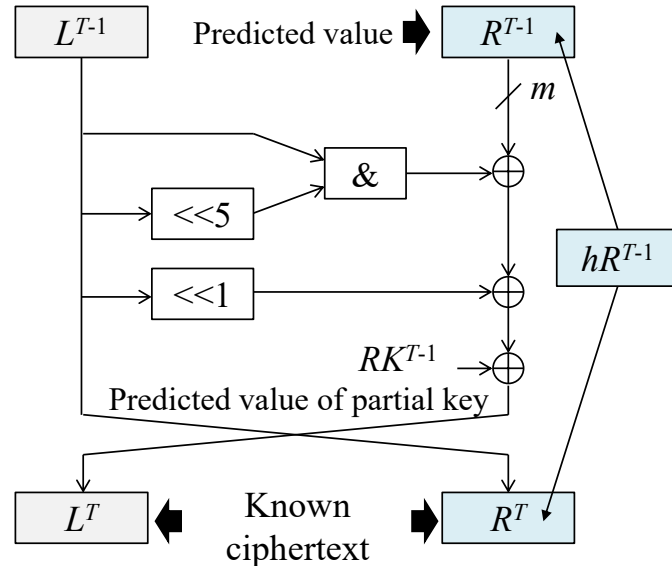


図 5.1 ベース電磁波解析手法

Fig. 5.1 Based electromagnetic analysis method

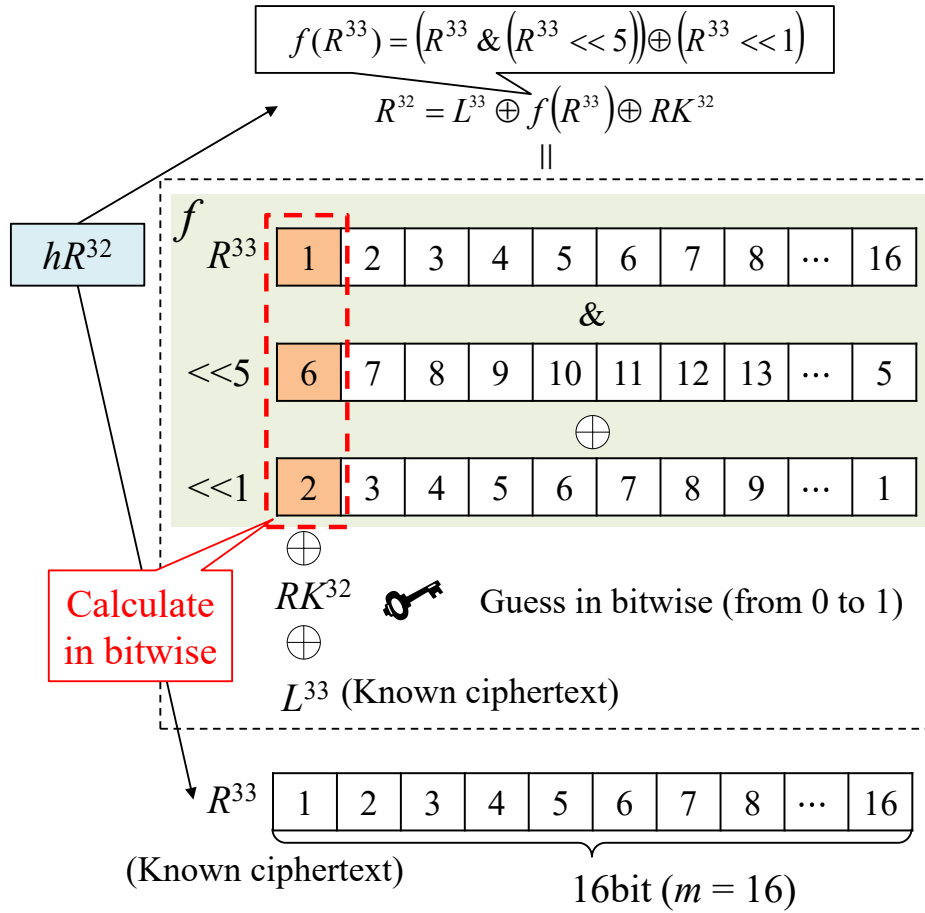


図 5.2 ベース解析手法の計算例

Fig. 5.2 Calculation example for the based analysis method

きる。更に、この計算ではラウンド鍵 RK^{T-1} の予測値はワード長である m bit ではなく、1bit ずつの部分鍵に分割して適用する。そのため、計算量を 2^m 通りから $2^1 \times m$ 通りまで削減することができる。

そして、暗号中間値 R^{T-1} と既知の暗号文 R^T とのハミング距離 hR^{T-1} を式(5.2)から計算する。このとき $HD(a, b)$ は、ある値 a と b のハミング距離を求める関数である。

$$hR^{T-1} = HD(R^{T-1}, R^T) \quad (5.2)$$

最後に式(3.1)を利用して、ハミング距離 hR^{T-1} と電磁波 w から相関係数 phR^{T-1} を計算する。そして、この相関係数 phR^{T-1} を最大とする鍵の予測値を正解鍵として推定する。

5.2.2 複数ラウンドを利用した解析

提案する電磁波解析手法では、解析効率を向上させるために、SIMECK の構造に着目した複数ラウンドを利用した解析を導入する。SIMECK では、その構造から 1 つのラウンド鍵の予測値によって、2 つのラウンドのハミング距離を導出することが可能である。この性質について図 5.3 を用いて説明する。

SIMECK では、式(A.11)から $L^r = R^{r+1}$ が成り立つ。そのため、図 5.3 に示すように、 $T-2$ ラウンド目と $T-1$ ラウンド目では、それぞれ $L^{T-2} = R^{T-1}$, $L^{T-1} = R^T$ が成り立つ。したがって、 $T-2$ ラウンド目の左半分の値のハミング距離 hL^{T-2} と、 $T-1$ ラウンド目の右半分の値のハミング距離 hR^{T-1} の間には、式(5.3)に示す関係が成り立つ。

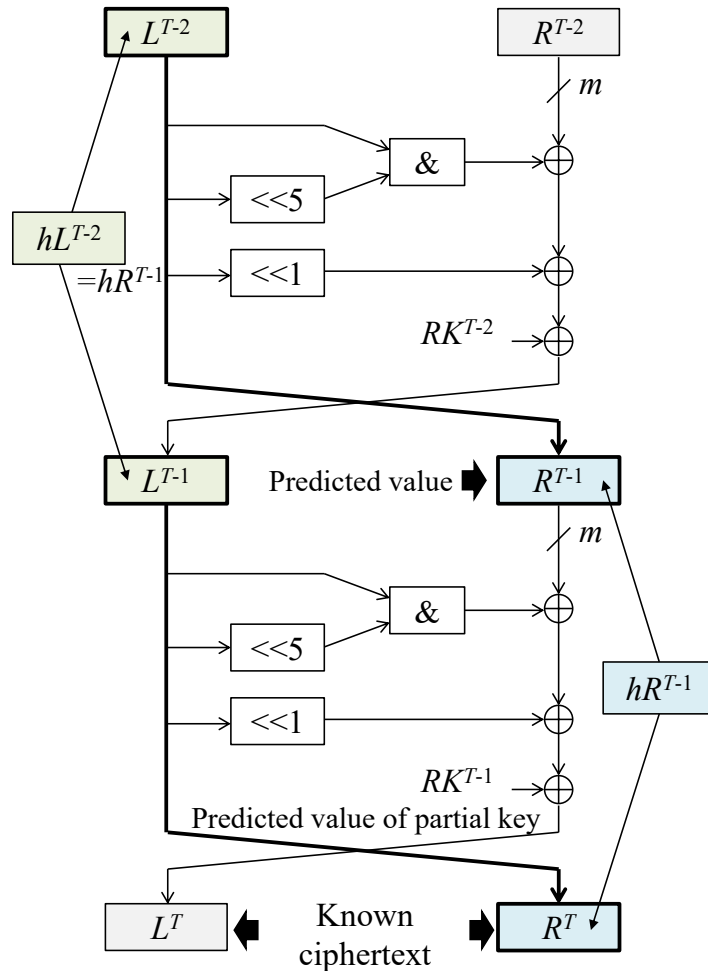


図 5.3 複数ラウンドを利用した解析で使用するハミング距離

Fig. 5.3 HD required by the analysis method using double rounds

$$hL^{T-2} = hR^{T-1} \quad (5.3)$$

以上より、1つのラウンド鍵 RK^{T-1} の予測値から、2つのラウンド ($T-2$ ラウンドと $T-1$ ラウンド) のハミング距離 hL^{T-2} , hR^{T-1} を求めることができる。したがって、式(5.3)が成り立つことから、正解鍵における相関係数には、図 5.4 (i)に示すような複数のラウンドでの複数のピークが観測される。

提案する電磁波解析手法では、この複数のラウンドで観測されるピークを解析に利用する。具体的には、図 5.4 (ii)に示すように、 $T-1$ ラウンド目で観測されるピーク $peak^{T-1}$ と、 $T-2$ ラウンド目で観測されるピーク $peak^{T-2}$ の合計値を解析に利用する。このように、提案する電磁波解析手法では複数のラウンドのより多くの情報を利用することで、解析効率を向上させる。

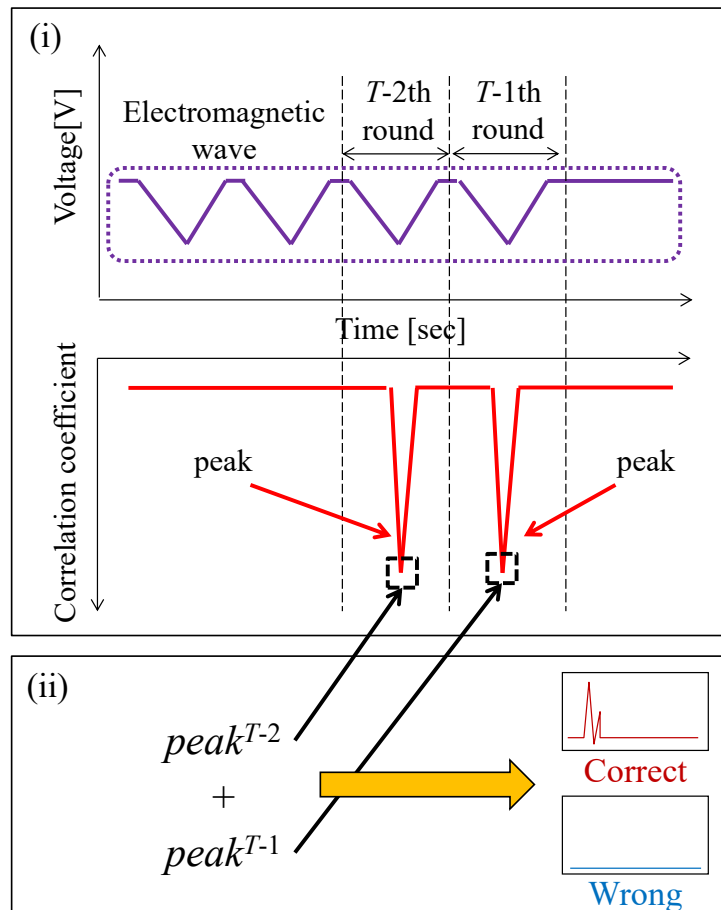


図 5.4 2つのラウンドでのピーク

Fig. 5.4 Peaks in double rounds

5.2.3 既知情報の組み込み処理

提案する電磁波解析手法では、既知情報の組み込み処理を導入することで、解析効率を向上させる。3.3.3 項でも述べた通り、サイドチャネル解析は、あるデータレジスタ間のハミング距離に着目して解析を行うため、着目しているハミング距離以外のデータ遷移で発生するサイドチャネル情報は、解析を行う際にノイズ成分となる。そのため、既知情報（信号成分）を増やし、未知情報（ノイズ成分）を減らすことで SNR を改善させ、効率的な解析が可能となる [4][5]。

既知情報の組み込み処理の概要を図 5.5 に示す。図 5.5 (i) に示すように、SIMECK では 1bit のハミング距離に着目して解析を行うため、残りの $2m-1$ bit 分のノイズが発生する。図 5.5 (ii) に示すように、既知情報の組み込み処理では、 m bit のノイズ成分を既知情報（信号成分）

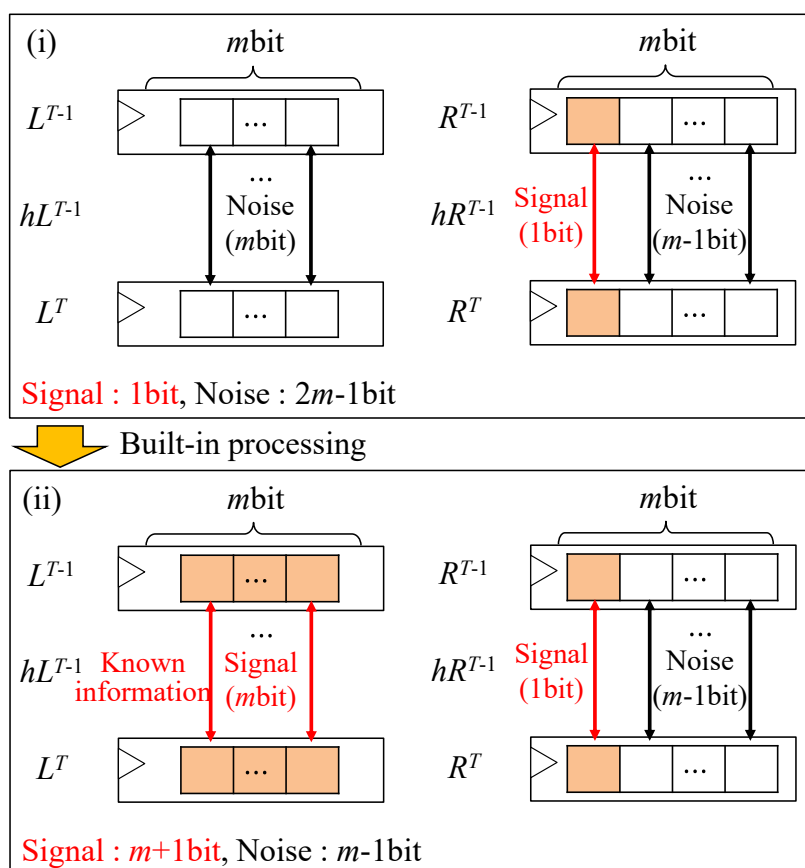


図 5.5 組み込み処理の概要

Fig. 5.5 Outline of the built-in processing

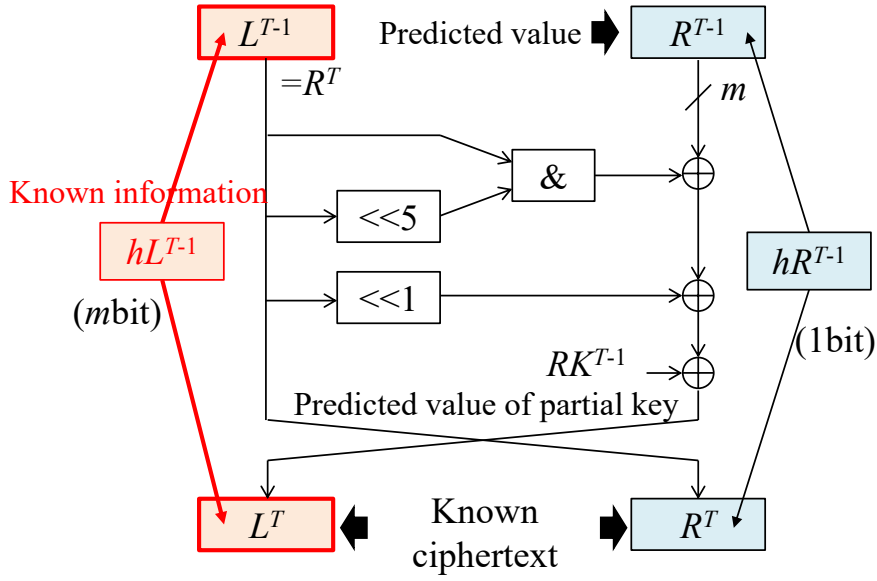


図 5.6 組込み処理で利用する既知情報

Fig. 5.6 Known information for the built-in processing

として扱うことで解析効率を向上させる。この既知情報には、図 5.6 に示すように暗号中間値 L^{T-1} と既知の暗号文 L^T とのハミング距離 hL^{T-1} を利用する。SIMECK では、その構造から $L^{T-1} = R^T$ が成り立つため、 $T-1$ ラウンド目の左半分値 L^{T-1} は既知である。したがって、式(5.4)によって既知情報 hL^{T-1} を計算することができる。

$$hL^{T-1} = HD(L^{T-1}, L^T) \quad (5.4)$$

そして、 $T-1$ ラウンド目を対象として相関係数を求める際に、既知情報 hL^{T-1} を組み込む。具体的には、式(3.1)を利用した相関係数の計算を行う際に、ハミング距離 h の代わりに既知情報を組み込んだハミング距離 $h_{built-in}^{T-1}$ を用いる。この $h_{built-in}^{T-1}$ の計算式を式(5.5)に示す。そして、 $h_{built-in}^{T-1}$ を用いて計算した相関係数 $\rho h_{built-in}^{T-1}$ を解析に利用する。

$$h_{built-in}^{T-1} = hL^{T-1} + hR^{T-1} \quad (5.5)$$

このように計算を行うことで、 m bit のノイズ成分が m bit の信号成分に変化するため、ノイズ成分を大幅に削減することができる。

また、提案する電磁波解析手法では複数のラウンド情報を利用するため、既知情報の組込み処理を、5.2.2 項の複数ラウンドを利用した解析と組み合わせて適用する。このとき、

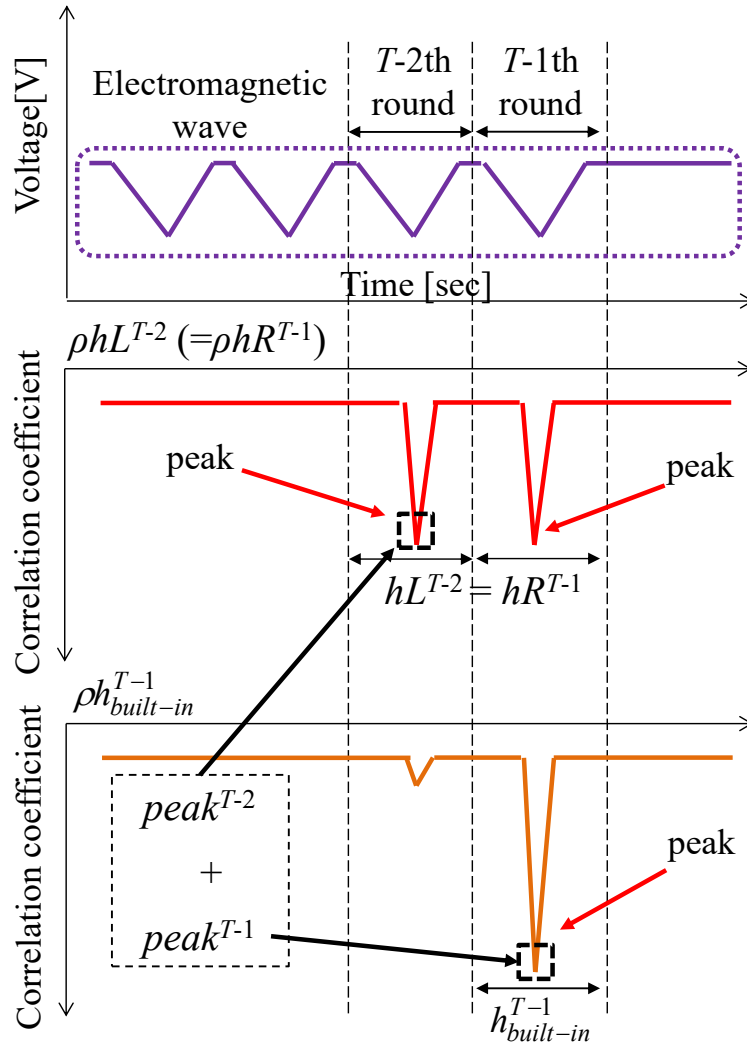


図 5.7 提案する電磁波解析手法でのピーク

Fig. 5.7 Peaks for the proposed electromagnetic analysis method

$T-1$ ラウンド目のラウンド鍵 RK^{T-1} を対象とした解析において、 $T-2$ ラウンド目のハミング距離 hL^{T-2} を求める際には、既知情報の組み込み処理は行わない。これは、ラウンド鍵 RK^{T-1} の予測値だけでは、 $T-2$ ラウンド目の残りのハミング距離 (hL^{T-2} の着目していない $m-1$ bit や hR^{T-2} に相当する m bit の値) を求めることができないからである。

したがって、図 5.7 に示すように、提案する電磁波解析手法ではハミング距離 $hL^{T-2}(=hR^{T-1})$ を利用して求めた相関係数 $phL^{T-2} (=phR^{T-1})$ のピーク $peak^{T-2}$ と、既知情報 hL^{T-1} を組込んだハミング距離 $h_{built-in}^{T-1}$ を利用して求めた相関係数 $ph_{built-in}^{T-1}$ のピーク $peak^{T-1}$ の合計値を解析に利用する。

5.3 評価実験

5.3.1 実験環境

実験環境を図 5.8 と表 5.1 にそれぞれ示す。評価ボードには SASEBO-W [6]を使用した。SIMECK は、ブロック長が 32bit，鍵長が 64bit のもの ($m = 16$, $T = 33$) を，SASEBO-W 上の FPGA に実装した。また，FPGA には Spartan-6 XC6SLX150-FGG484 を使用した。電磁波波形の取得に関して，電磁波測定用プローブにはシールドディッドループアンテナを，電磁波測定用 RF アンプには MITEQ AU-3A-0150 (0.3-500MHz, 52dBmin) を使用した。実験で使用したシールドディッドループアンテナを図 5.9 に示す。図 5.9 に示すように，ループ開口面の大きさは直径約 10mm である。電磁波波形の測定では，ループ開口面が FPGA に対して水平になるようにシールドディッドループアンテナを固定した。平文はランダムに生成し，40,000 回の暗号処理を行い，40,000 個の電磁波波形を取得した。

また，実際に取得した電磁波波形の例を図 5.10 に示す。評価実験では，図 5.10 に示す電磁波波形を用いて提案する電磁波解析手法による解析を行う。

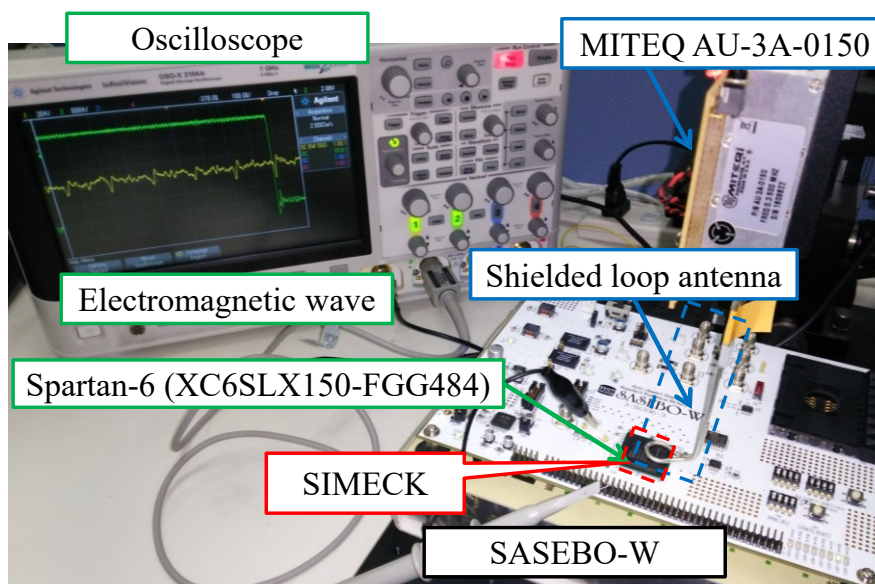


図 5.8 評価システム

Fig. 5.8 Evaluation system

表 5.1 実験環境

Table 5.1 Experimental condition

Encryption algorithm	SIMECK
Block size [bit]	32
Key size [bit]	64
Evaluation board	SASEBO-W [6]
FPGA	Xilinx Spartan-6 XC6SLX140
Implementation tool	Xilinx ISE Design Suite 14.7
Near field magnetic probe	Shielded loop antenna
Oscilloscope	Agilent DSO-X 3104A
Sampling rate [GSa/sec]	5
PC	HP ProBook 6570b
Memory [GB]	8
CPU	Core i7-3570M
Programming language	MATLAB 2013b

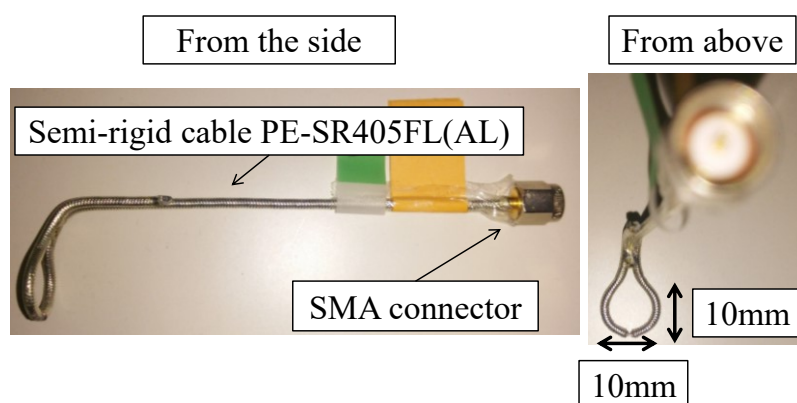


図 5.9 シールドイットドループアンテナ

Fig. 5.9 Shielded loop antenna

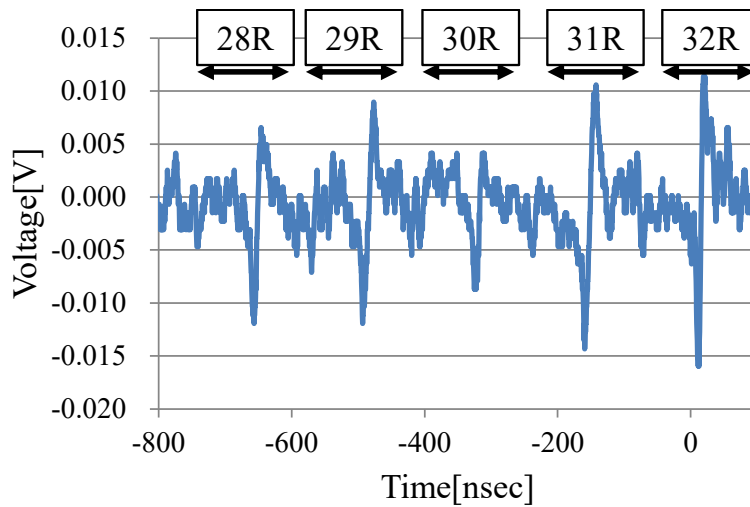


図 5.10 取得波形の例

Fig. 5.10 Example of the measured electromagnetic waveform

5.3.2 実験結果

まず、SIMECK の最終ラウンド（32 ラウンド目）のラウンド鍵 RK^{32} を対象に解析を行った。このとき、解析では既知情報の組込み処理と、複数ラウンドを利用した解析を適用した。実験結果を図 5.11 に示す。図の横軸は解析に使用した電磁波波形の数を、縦軸は解析に成功した部分鍵の bit 数を示している。この実験では SIMECK の最終ラウンドで使用するラウンド鍵 RK^{32} （16bit）を対象としているため、縦軸の最大値は 16 である。図 5.11 に示すように、13,000 波形のデータを用いた解析によって全ての部分鍵の解析に成功していることが分かる。したがって、SIMECK は電磁波解析に対して脆弱であり、提案する電磁波解析手法は有効であると考えられる。

また、解析対象の SIMECK（ブロック長 32bit、鍵長 64bit）の秘密鍵を全て求めるためには、残りの 3 つのラウンドのラウンド鍵（ RK^{29} , RK^{30} , RK^{31} ）を求める必要がある。そこで、この残りの 3 つのラウンド鍵を対象に解析を行った。実験結果を図 5.12 に示す。図 5.12 に示すように、31,000 波形を用いた解析によって残りのラウンド鍵（ RK^{29} , RK^{30} , RK^{31} ）の推定に成功していることが分かる。すなわち、64bit の秘密鍵を全て求めることが可能である。以上より、提案する電磁波解析手法によって SIMECK の耐タンパ性を検証することができる。

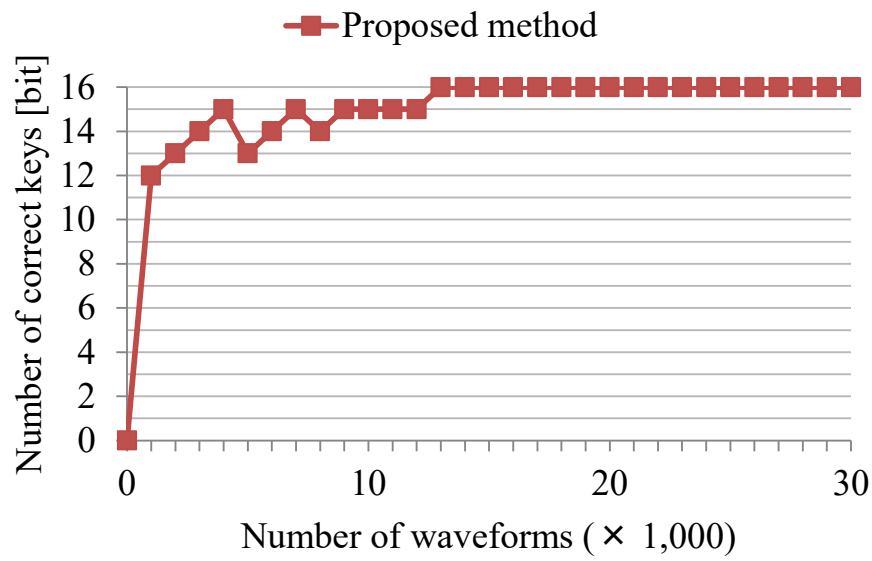


図 5.11 実験結果

Fig. 5.11 Experimental result

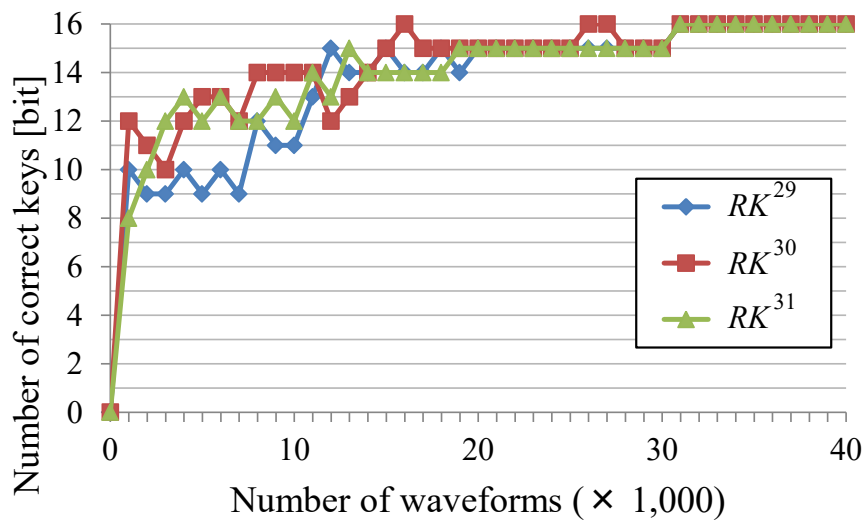


図 5.12 ラウンド鍵 RK^{29} , RK^{30} , RK^{31} の解析結果

Fig. 5.12 Experimental results for round keys RK^{29} , RK^{30} , and RK^{31}

また、解析に必要な処理時間を表 5.2 に示す。表 5.2 は、31,000 個の波形データを使用した場合における、64bit の全ての秘密鍵の解析に必要な処理時間を示したものである。表 5.2 に示すように、解析に必要な合計の処理時間は 583 分であり、現実的な時間で解析を行うことが可能であり、提案する電磁波解析手法は有効であると考えられる。

5.3.3 複数ラウンドを利用した解析の有効性評価

複数ラウンドを利用した解析の有効性について検証した。この検証では、31 ラウンド目と 32 ラウンド目の複数ラウンドを利用した解析と、31 ラウンド目のみを利用した解析、32 ラウンド目のみを利用した解析をそれぞれ適用した。また、この実験では既知情報の組み込み処理は適用していない。比較結果を図 5.13 に示す。

図 5.13 に示すように、複数ラウンドを利用した解析では 19,000 波形を用いることで全ての部分鍵を解析することに成功した。一方で、31 ラウンド目のみを利用した解析では、解析に 30,000 波形のデータが必要であることが分かる。また、32 ラウンド目のみを利用した解析は、30,000 波形のデータを使用しても、全ての部分鍵を推定することはできなかった。したがって、解析効率が向上していることから、複数ラウンドを利用した解析は有効であると考えられる。

また、正解鍵における相関係数を図 5.14 に示す。図 5.14 は、30,000 波形のデータを用いて相関係数を計算した結果であり、図の横軸は SIMECK の処理時間を、縦軸は相関係数を示している。図 5.14 の結果から、31 ラウンド目と 32 ラウンド目の複数のラウンドで、複数の相関係数のピークが表れていることが確認できる。

表 5.2 解析時間

Table 5.2 Analytical time

	Time [min]
Analysis	173
Acquisition of waveforms	410
Total	583

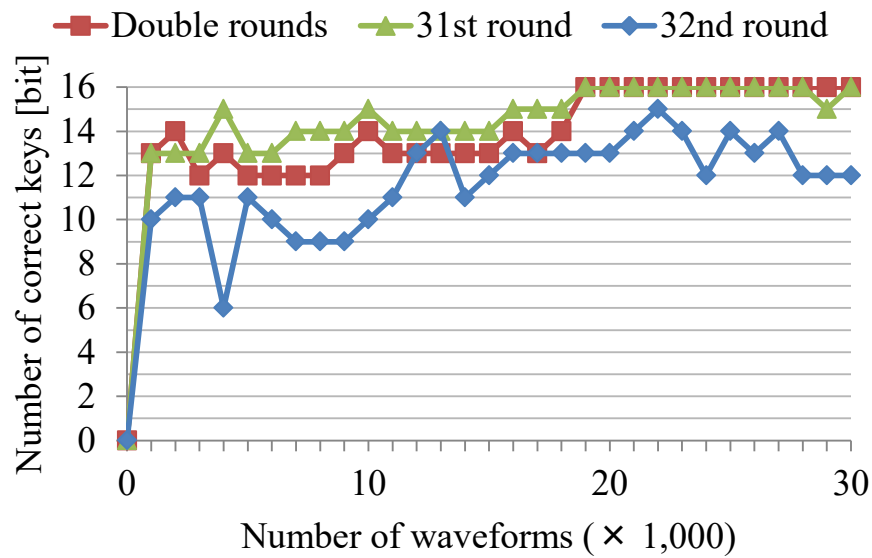


図 5.13 複数ラウンドを利用した解析を行う場合と行わない場合での比較結果

Fig. 5.13 Comparison of experimental results with and without double rounds

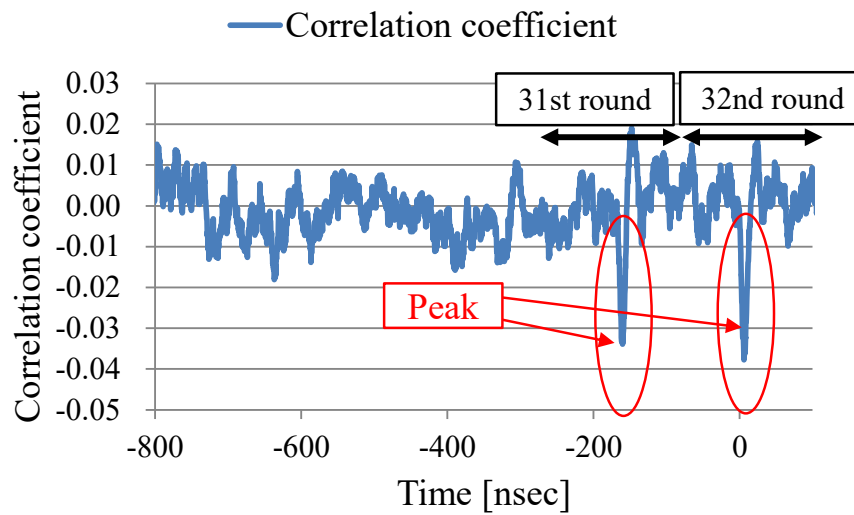


図 5.14 正解鍵における相関係数の結果

Fig. 5.14 Result of the correlation coefficient for the correct key

5.3.4 既知情報の組込み処理の有効性評価

既知情報の組込み処理の有効性について検証した。この検証では、32 ラウンド目を対象に既知情報の組込み処理を行う場合と、行わない場合で比較実験を行った。比較結果を図 5.15 に示す。

図 5.15 に示すように、既知情報の組込み処理を行う場合、24,000 波形のデータを使用することで全ての部分鍵の解析に成功していることが分かる。一方で、既知情報の組込み処理を行わない場合、30,000 波形のデータを使用した場合でも、全ての部分鍵を推定することはできなかった。したがって、解析効率が向上していることから、既知情報の組込み処理は有効であると考えられる。

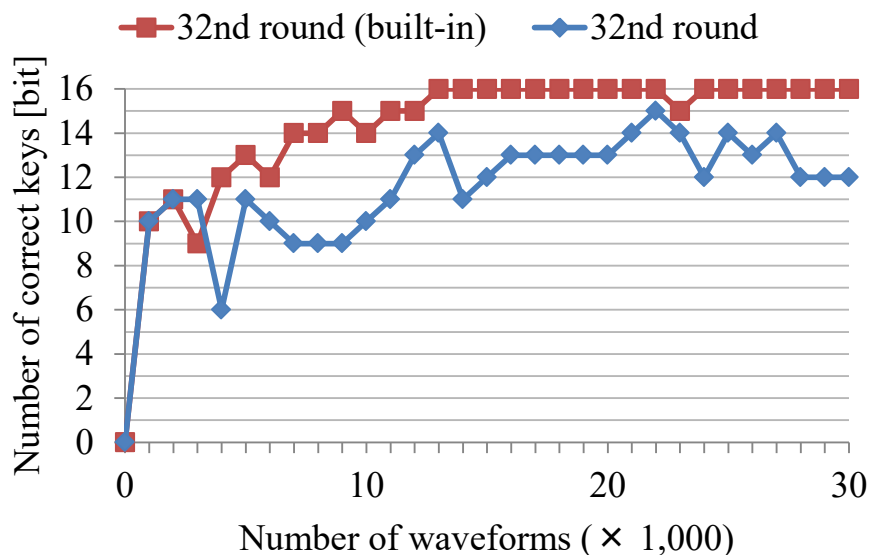


図 5.15 既知情報の組込み処理を行う場合と行わない場合での比較結果

Fig. 5.15 Comparison of with and without built-in processing

5.4 結言

本章では、最も小回路規模でハードウェア実装が可能な軽量暗号である SIMECK に対する電磁波解析手法を提案し、FPGA を用いた評価実験によってその有効性を示した。提案した電磁波解析手法では、SIMECK の構造に着目した複数ラウンドを利用した解析を導入することで、解析効率を向上させた。更に、既知情報の組み込み処理を導入することで、ノイズ成分の影響を抑え、高効率な解析を実現させた。

以上より、提案した電磁波解析手法を利用することで、SIMECK の耐タンパ性を評価できることを示した。

参考文献

- [1] 野崎佑典, 池崎良哉, 吉川雅弥, “軽量暗号 Simeck に対する二重電磁波解析攻撃とその評価,” 電気学会論文誌 C, vol. 136, no. 12, pp. 1613–1620, Dec. 2016.
- [2] G. Yang, B. Zhu, V. Suder, M. D. Aagaard, and G. Gong, “The Simeck Family of Lightweight Block Ciphers,” Proc. 17th Int. Workshop on Cryptographic Hardware and Embedded Systems (CHES 2015), LNCS vol. 9293, pp. 307–329, Springer, Sep. 2015.
- [3] O. Meynard, S. Guilley, J.-L. Danger, and L. Sauvage, “Far Correlation-based EMA with a Precharacterized Leakage Model,” Proc. Design, Automation and Test in Europe Conference and Exhibition (DATE 2010), pp. 977–980, Mar. 2010.
- [4] Y. Komano, H. Shimizu, and S. Kawamura, “BS-CPA: Built-In Determined Sub-Key Correlation Power Analysis,” IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences, vol. E93-A, no. 9, pp. 1632–1638, Sep. 2010.
- [5] 駒野雄一, 清水秀夫, 野崎華恵, 新保 淳, 川村信一, “組込型相関電力解析 BS-CPA の解析効率の評価,” 2011 年暗号と情報セキュリティシンポジウム講演論文集, 3D4-4, pp. 1–8, Jan. 2011.
- [6] “The National Institute of Advanced Industrial Science and Technology Research Institute for Secure Systems: Evaluation Environment for Side-channel Attacks,”
<http://www.risec.aist.go.jp/project/sasebo>

第 6 章

Midori に対する電力解析手法

6.1 緒言

本章では, Midori の耐タンパ性評価に必要な電力解析手法を提案する [1]。Midori [2]は, 軽量暗号の 5 大要素の 1 つである低消費電力で利用可能な暗号アルゴリズムであり, CRYPTREC 軽量暗号 WG の暗号技術ガイドライン (軽量暗号) [3]の評価対象になっている。また, Midori は低消費電力で動作するため, バッテリ寿命を長くさせることができ, センサ機器などへの利用が期待されている [2]。Midori のアルゴリズムの詳細については, 付録 (A.3 節) で述べる。

電力解析の研究に関して, これまでに Midori を対象とした手法は報告されていない。本章で提案する電力解析手法では, CPA [4]を階層的に適用することで, 秘密鍵を推定する。また, Midori の構造に着目することで, 耐タンパ性評価に必要な計算量を大幅に削減する。

6.2 提案する電力解析手法

提案する電力解析手法では, 2 段階の階層的な電力解析を導入することで, Midori の秘密鍵を全て推定する。まず 6.2.1 項では, 提案する階層的な電力解析手法の概要について説明する。そして 6.2.2 項と 6.2.3 項では, それぞれ提案手法を構成する 1 段階目の電力解析と 2 段階目の電力解析について述べる。一方で, 詳しくは 6.2.3 項で述べるが, 2 段階目の電力解析では計算量が多くなる。そのため提案する電力解析手法では, 6.2.4 項で述べる解析の効率化手法を導入する。

本章は[1]で発表された内容から構成される。

6.2.1 階層的な電力解析手法

階層的な電力解析手法の概要を図 6.1 に示す。この解析では、1 段階目の電力解析と 2 段階目の電力解析をそれぞれ行うことで、Midori の秘密鍵 K を全て推定する。

1 段階目の電力解析では、最終ラウンド (16R) を対象に電力解析を行い、鍵 WK を推定する。次に、2 段階目の電力解析では、2R を対象に電力解析を行うことで、鍵 K_0 を推定する。このとき、2 段階目の電力解析では、1 段階目の電力解析で解析済の鍵 WK を用いて解析を行う。

6.2.2 1 段階目の電力解析手法

1 段階目の電力解析では、最終ラウンド (16R) を対象に電力解析を行い、鍵 WK を推定する。具体的には、図 6.2 に示すように、1 段階目の電力解析は既知の暗号文 C と 15R 終了後の暗号中間値 x^{15} とのハミング距離を解析に用いるものとする。このとき、対象とする暗号中間値 x^{15} は未知の値であり、式(6.1)を用いて計算する。ここで、 $S(\cdot)$ は SubCell の S-BOX による処理を表している (A.3 節, 表 A.5 を参照)。また、S-BOX は 4bit 単位で処理を行うため、鍵 WK には 4bit の値を試す。すなわち、 $2^4 = 16$ 通りの鍵候補を全て試行する。

$$x^{15} = S(C \oplus WK) \quad (6.1)$$

次に、計算した暗号中間値 x^{15} と暗号文 C とのハミング距離 h を計算する。この計算式を式(6.2)に示す。ここで、 $HW(A)$ はある値 A のハミング重みを計算する関数である。

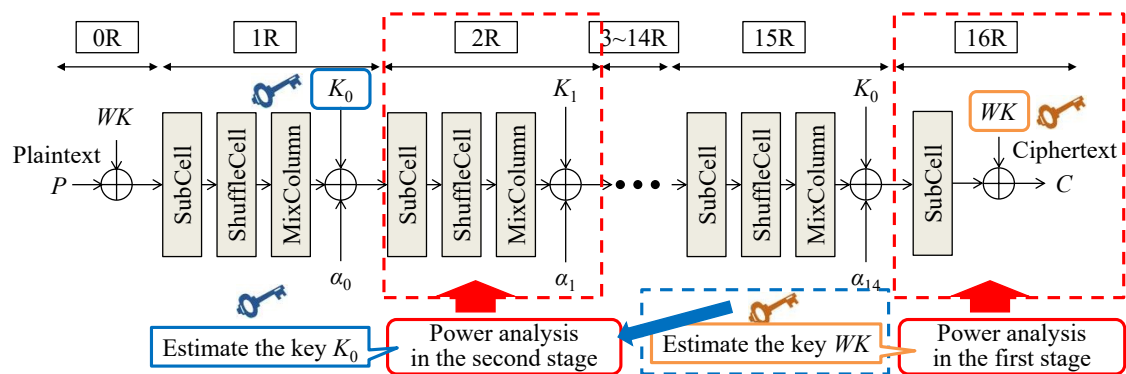


図 6.1 提案する電力解析手法の概要

Fig. 6.1 Outline of the proposed power analysis method

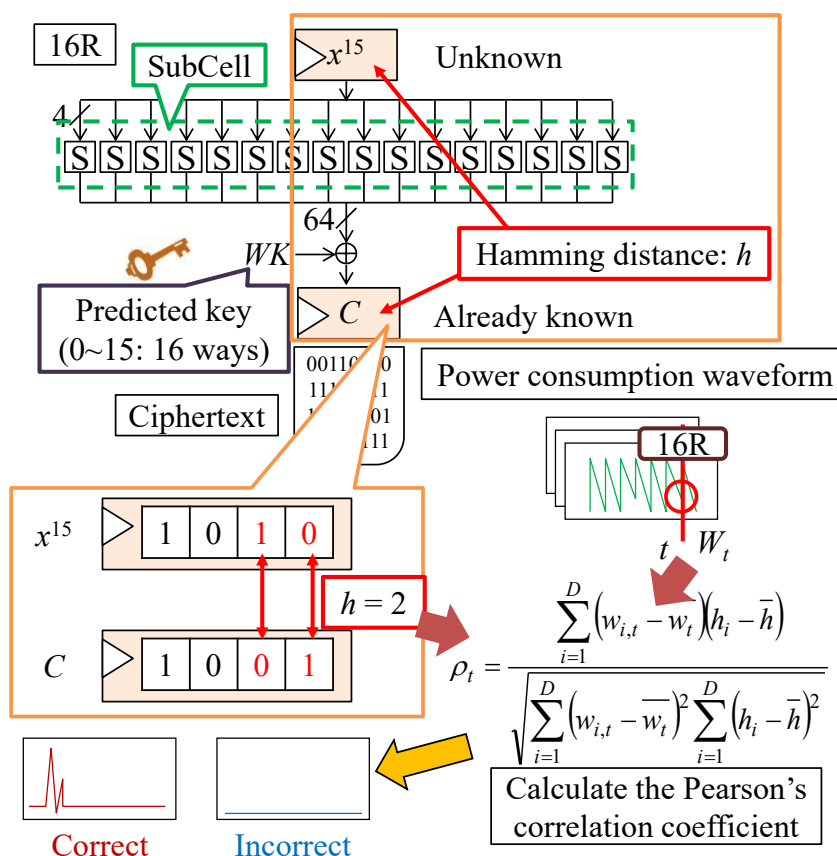


図 6.2 1 段階目の電力解析手法

Fig. 6.2 Power analysis method in the first stage

$$h = HW(x^{15} \oplus C) \quad (6.2)$$

鍵 WK の解析では、算出したハミング距離 h を用いて CPA [4] をベースとした解析を行う。具体的には、式(3.1)を用いてピアソンの相関係数を算出する。このとき、計算に用いる消費電力は、最終ラウンド（16R）処理時に対応する部分の値を使用する。そして、ピアソンの相関係数を最大とする鍵 WK の予測値を正解鍵として推定する。また、1 段階目の電力解析では、この計算を各 S-BOX に対応させて残り 15 回行い、全ての WK （64bit）を推定する。すなわち、合計で $2^4 \times 16 = 256$ 通りの計算を行う。

6.2.3 2 段階目の電力解析手法

2 段階目の電力解析では、2R を対象に電力解析を行い、鍵 K_0 を推定する。具体的には、図 6.3 に示すように、2 段階目の電力解析は、1R 終了後の暗号中間値 x^1 と 2R 終了後の暗号中間値 x^2 とのハミング距離を解析に用いるものとする。このとき、対象とする暗号中間

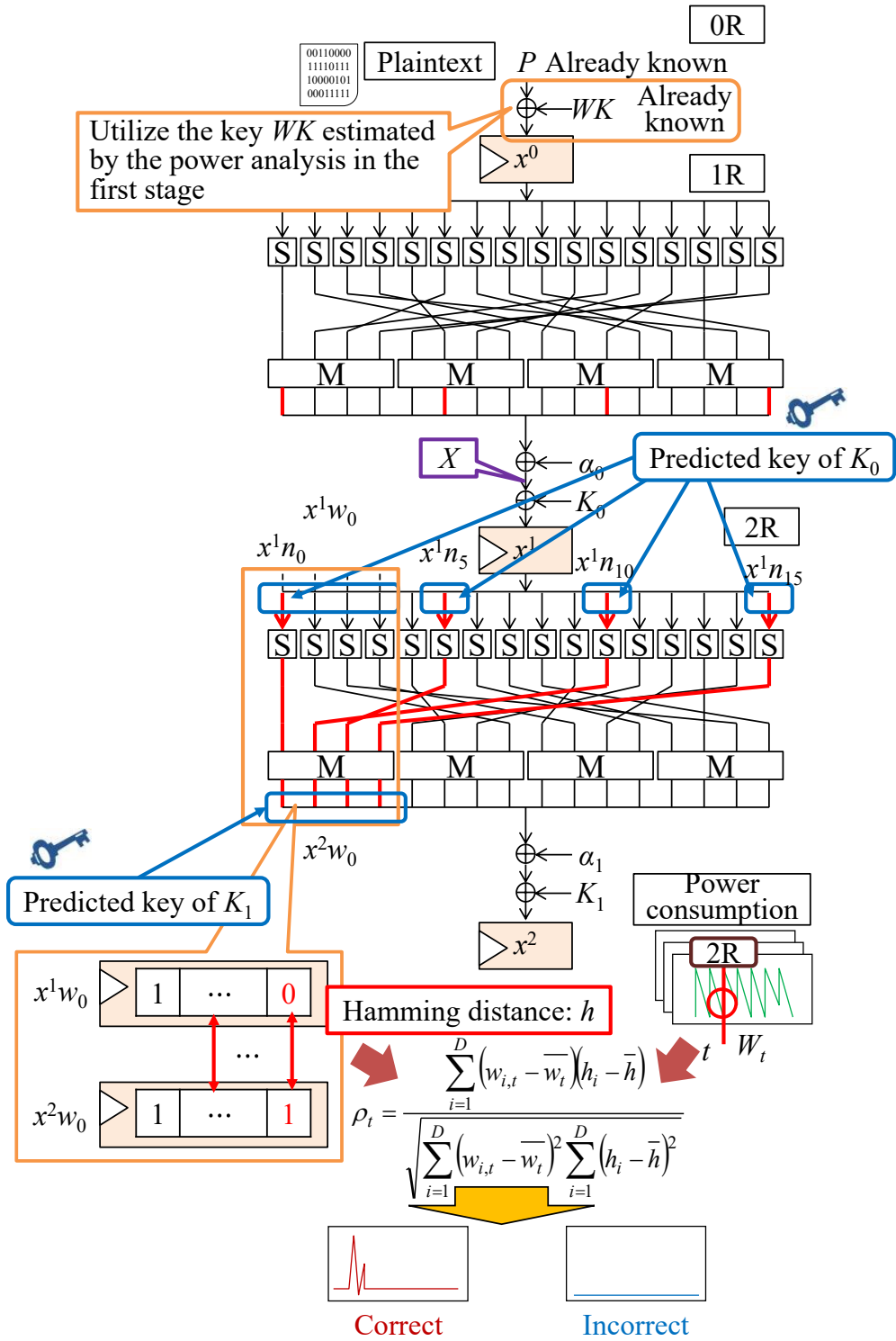


図 6.3 2 段階目の電力解析

Fig. 6.3 Power analysis in the second stage

値 x^1 と x^2 はどちらも未知の値である。 x^1 と x^2 の計算方法について、64bit の暗号中間値 x を4個のワード単位 (16bit 単位) xw_0, xw_1, xw_2, xw_3 ($x = xw_0 \parallel xw_1 \parallel xw_2 \parallel xw_3$) に分割したときの xw_0 に着目した場合を例に説明する。また、 x はニブル単位 (4bit 単位) では、16個の値 $xn_0, \dots, xn_{15}$ を用いて、 $x = xn_0 \parallel \dots \parallel xn_{15}$ で表すことができる。

まず、1R 終了後の暗号中間値 x^1w_0 の導出方法について説明する。 x^1w_0 は0R 終了後の暗号中間値 x^0 ($x^0 = x^0n_0 \parallel \dots \parallel x^0n_{15}$) を用いて式(6.3)で計算することができる。ここで、式(6.3)の $M()$ は式(A.13)の MixColumn による行列演算を、 K_0w_0 と α_0w_0 はそれぞれ K_0 と α_0 の0番目のワードの値を表している。また、 x^0 は既知の平文 P と 6.2.2 項の1段階目の電力解析で推定した鍵 WK を用いて計算することができる ($x^0 = P \oplus WK$)。このとき、式(6.3)の MixColumn は 16bit 単位で計算を行っているため、鍵 K_0w_0 の予測値には 16bit の値を全て試行する。

$$x^1w_0 = M(S(x^0n_0) \parallel S(x^0n_{10}) \parallel S(x^0n_5) \parallel S(x^0n_{13})) \oplus K_0w_0 \oplus \alpha_0w_0 \quad (6.3)$$

次に、2R 終了後の暗号中間値 x^2w_0 の導出方法について説明する。図 6.3 に示すように、 x^2w_0 は $x^1n_0, x^1n_{10}, x^1n_5, x^1n_{15}$ を用いて計算する。まず、 $x^1n_0, x^1n_{10}, x^1n_5, x^1n_{15}$ の計算式を式(6.4)に示す。このとき、式(6.4)の X は 1R で SubCell, ShuffleCell, MixColumn, ラウンド定数 α_0 による排他的論理和を適用した後の値である。また、 K_0n は鍵 K_0 をニブル単位で分割したときの値であり、 $K_0n_0, K_0n_{10}, K_0n_5, K_0n_{15}$ はそれぞれ、0番目、10番目、5番目、15番目の鍵 K_0n の値を示している。

$$\begin{aligned} x^1n_0 &= X_0 \oplus K_0n_0 \\ x^1n_{10} &= X_{10} \oplus K_0n_{10} \\ x^1n_5 &= X_5 \oplus K_0n_5 \\ x^1n_{15} &= X_{15} \oplus K_0n_{15} \end{aligned} \quad (6.4)$$

そして、 x^2w_0 は $x^1n_0, x^1n_{10}, x^1n_5, x^1n_{15}$ を用いて式(6.5)で計算することができる。ここで、式(6.5)の K_1w_0 と α_1w_0 は、それぞれ K_1 と α_1 の0番目のワードの値を表している。また、式(6.5)の MixColumn は 16bit 単位で計算を行っているため、鍵 K_1w_0 の予測値には 16bit の値を全て試行する。

$$x^2w_0 = M(S(x^1n_0) \parallel S(x^1n_{10}) \parallel S(x^1n_5) \parallel S(x^1n_{15})) \oplus K_1w_0 \oplus \alpha_1w_0 \quad (6.5)$$

更に、各暗号中間値 x^1w_0 と x^2w_0 のハミング距離 h は式(6.6)で計算することができる。このとき、ハミング距離 h の計算において、各暗号中間値 x^1w_0 と x^2w_0 の導出(式(6.3), 式(6.4), 式(6.5)) には、11 個の鍵 $K_0n_0, K_0n_1, K_0n_2, K_0n_3, K_0n_5, K_0n_{10}, K_0n_{15}, K_1n_0, K_1n_1, K_1n_2, K_1n_3$ の合計で 44bit の値が必要である。すなわち、 $2^{44} \approx 1.8 \times 10^{13}$ 通りの計算を行う必要がある。そこで提案する電力解析手法では、6.2.4 項で説明する解析の効率化手法を導入することで、計算量を削減する。

$$h = HW(x^1w_0 \oplus x^2w_0) \quad (6.6)$$

そして、2 段階目の電力解析では、導出したハミング距離 h と式(3.1)を用いて、CPA をベースとした解析を行い、鍵 K_0 を推定する。

最後に、式(A.14)によって $WK = K_0 \oplus K_1$ が成り立つため、鍵 K_1 は式(6.7)を用いて導出することができる。すなわち、128bit の秘密鍵 K ($K = K_0 \parallel K_1$) を全て求めることができる。

$$K_1 = WK \oplus K_0 \quad (6.7)$$

以上のように、1 段階目の電力解析では 16R におけるハミング距離を、2 段階目の電力解析では 2R におけるハミング距離を解析に用いることで、最終的に Midori の秘密鍵を全て推定することができる。

6.2.4 解析の効率化手法

6.2.3 項で述べた通り、2 段階目の電力解析では、44bit の鍵の予測値を必要とするため、本項で述べる解析の効率化手法を導入することで、計算量を削減する。解析の効率化手法では、解析済の鍵 WK を用いた効率化手法と、MixColumn における効率化手法を行う。

まず、解析済の鍵 WK を用いた効率化手法について、図 6.4 を用いて説明する。解析済の鍵 WK を用いた効率化手法では、1 段階目の電力解析で鍵 WK が既知となることを利用して計算量を削減する。

ここでは、図 6.4 に示すように、式(6.3)の $M(S(x^0n_0) \parallel S(x^0n_{10}) \parallel S(x^0n_5) \parallel S(x^0n_{13})) \oplus \alpha_0w_0$ を Y^1 に置き換える。このとき、暗号中間値 x^1w_0 は Y^1 を用いて式(6.8)で計算することができる。

$$x^1w_0 = Y^1 \oplus K_0w_0 \quad (6.8)$$

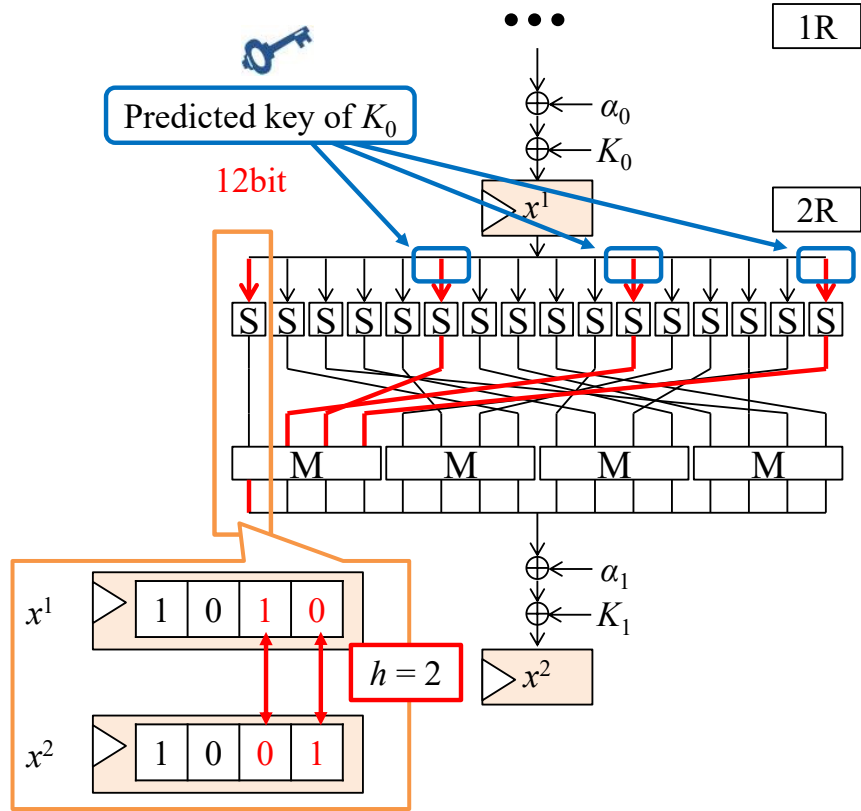


図 6.5 MixColumn における解析の効率化手法

Fig. 6.5 Efficiency improvement method of MixColumn

次に、MixColumn における効率化手法について、図 6.5 を用いて説明する。MixColumn における効率化手法では、MixColumn の計算をビット単位で行うことができる点に着目する。具体的には、MixColumn の 16bit の入力値を $mi_0 \parallel \dots \parallel mi_{15}$ 、出力値を $mo_0 \parallel \dots \parallel mo_{15}$ とすると、4bit の出力値 $mo_0 \parallel mo_1 \parallel mo_2 \parallel mo_3$ は、式(A.13)の行列 M から式(6.11)で計算することができる。

$$\begin{aligned}
 mo_0 &= mi_4 \oplus mi_8 \oplus mi_{12} \\
 mo_1 &= mi_5 \oplus mi_9 \oplus mi_{13} \\
 mo_2 &= mi_6 \oplus mi_{10} \oplus mi_{14} \\
 mo_3 &= mi_7 \oplus mi_{11} \oplus mi_{15}
 \end{aligned} \tag{6.11}$$

そして、図 6.5 と式(6.11)に示すように、4bit の出力値 ($mo_0 \parallel mo_1 \parallel mo_2 \parallel mo_3$) は 12bit の入力値 ($mi_4, mi_5, mi_6, mi_7, mi_8, mi_9, mi_{10}, mi_{11}, mi_{12}, mi_{13}, mi_{14}, mi_{15}$) を用いて計算できる。したがって、2 段階目の電力解析では、12bit の鍵 ($K_{0n_5}, K_{0n_{10}}, K_{0n_{15}}$) の予測値を用いて計算を行うことができる。また、16bit の鍵 ($K_{0n_0}, K_{0n_5}, K_{0n_{10}}, K_{0n_{15}}$) の残り

4bit の鍵 (K_{0n_0}) は、解析済の 12bit の値 (K_{0n_5} , $K_{0n_{10}}$, $K_{0n_{15}}$) を用いて同様の電力解析を行うことで、求めることができる。すなわち、計算量を 2^{16} から $2^{12} + 2^4 = 4,112$ 通りまで削減することができる。

また、2 段階目の電力解析では、この計算を各 MixColumn に対応させて、4 回行う。すなわち、合計で $(2^{12} + 2^4) \times 4 = 16,448$ 通りの計算を行う。

6.3 評価実験

6.3.1 実験環境

実験環境を図 6.6 と表 6.1 に示す。評価ボードには、SASEBO-GII [5] を使用し、SASEBO-GII 上に搭載されている FPGA Virtex-5 にブロック長が 64bit の Midori を FPGA 実装した。

実験では、乱数を用いて平文を 10,000 個作成し、10,000 回の暗号処理を行った。そして、暗号処理時の消費電力を、オシロスコープを用いて測定した。この消費電力波形は、暗号

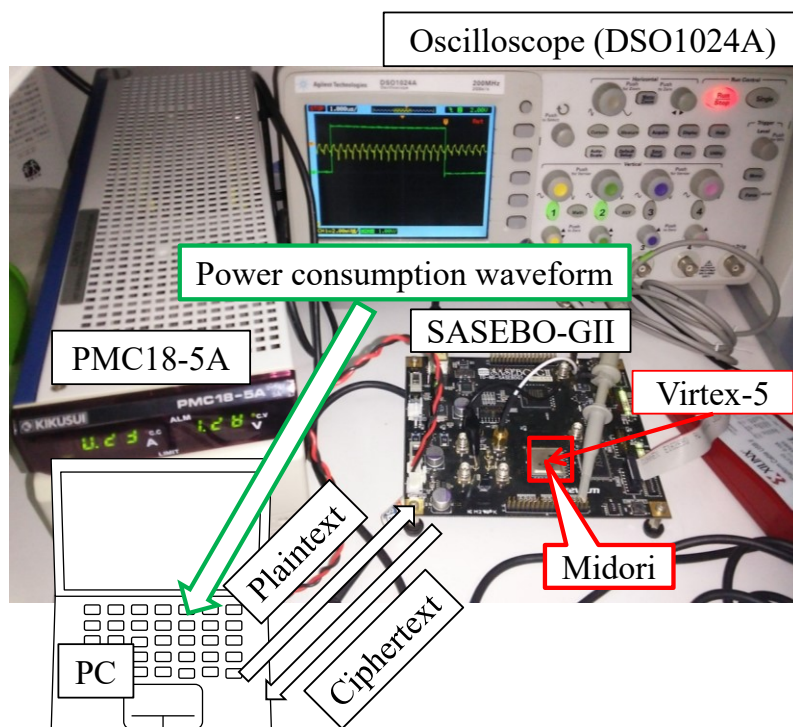


図 6.6 評価システム

Fig. 6.6 Evaluation system

表 6.1 実験環境

Table 6.1 Experimental condition

Encryption algorithm	Midori
Block size [bit]	64
Key size [bit]	128
Evaluation board	SASEBO-GII
FPGA	Xilinx Virtex-5 XC5VLX30
Implementation tool	Xilinx ISE Design Suite 14.1
Power supply	KIKUSUI PMC 18-5A
Oscilloscope	Agilent DSO 1024A
Sampling rate [GSa/sec]	2
PC	HP ProBook 6570b
Memory [GB]	8
CPU	Core i7-3570M
Programming language	MATLAB 2013b

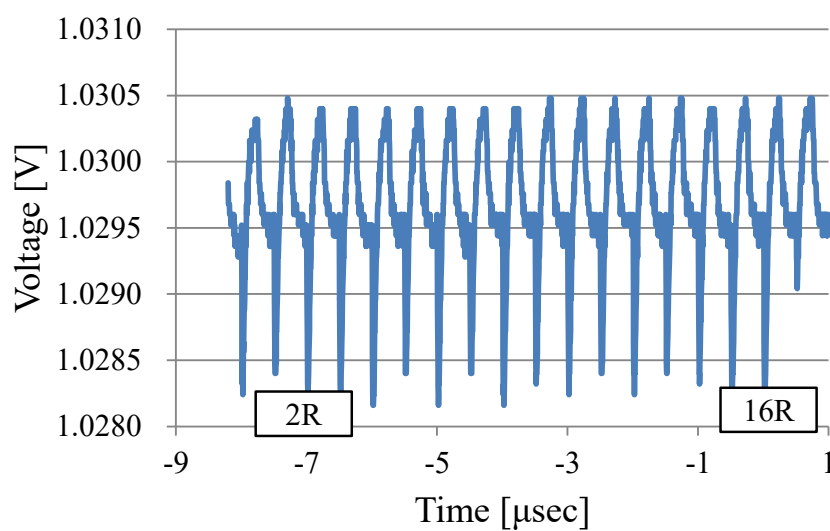


図 6.7 取得した消費電力波形の例

Fig. 6.7 Example of the acquired power consumption waveform

ハードウェアの消費電流を $1[\Omega]$ のシャント抵抗で測定した電圧波形である。取得した消費電力波形の例を図 6.7 に示す。評価実験では、図 6.7 に示す消費電力波形の 2R と 16R の処理部分を解析に使用する。

6.3.2 実験結果

まず、1 段階目の電力解析を行った。1 段階目の電力解析では、最終ラウンド (16R) を対象に解析を行い、鍵 WK を推定する。実験結果を図 6.8 に示す。図 6.8 の縦軸は解析に成功した鍵のビット数を、横軸は解析に使用した消費電力波形の数を示している。図 6.8 に示すように、5,000 個の波形データを用いることで全ての鍵 WK の解析に成功した。したがって、1 段階目の電力解析が有効であることが分かる。

次に、2 段階目の電力解析を行った。2 段階目の電力解析では、2R を対象に解析を行い、鍵 K_0 を推定する。実験結果を図 6.9 に示す。図 6.9 に示すように、2,000 個の波形データを使用することで、全ての鍵 K_0 の推定に成功した。したがって、2 段階目の電力解析は有効であることが分かる。以上より、提案する電力解析手法によって鍵 WK と K_0 を全て推定し、128bit の秘密鍵 K を全て求めることに成功した。したがって、提案する電力解析手法を用いることで、Midori の耐タンパ性を評価することができる。

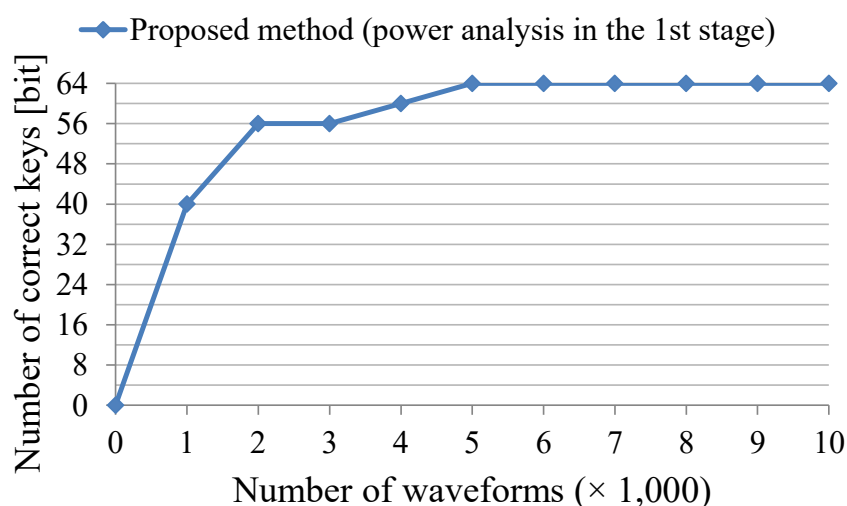


図 6.8 1 段階目の電力解析の結果

Fig. 6.8 Experimental result of the power analysis in the first stage

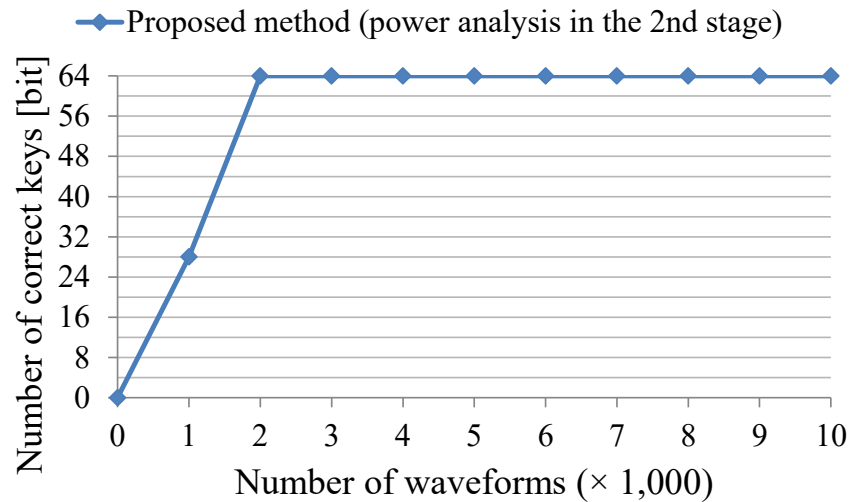


図 6.9 2 段階目の電力解析の結果

Fig. 6.9 Experimental result of the power analysis in the second stage

6.3.3 相関係数の比較

提案する電力解析手法における正解鍵と不正解鍵の相関係数について比較した。比較結果を図 6.10 と図 6.11 に示す。図 6.10 と図 6.11 は、それぞれ 1 段階目の電力解析と 2 段階目の電力解析における結果を示したものである。図 6.10 と図 6.11 の縦軸は相関係数を、横軸は時間を示している。また、赤色は正解鍵、灰色は不正解鍵を示している。

図 6.10 と図 6.11 に示すように、正解鍵ではそれぞれの解析ポイント（1 段階目の電力解析では 16R, 2 段階目の電力解析では 2R）において、ピークが確認できる。このように、正解鍵でのピークを確認できることから、高い相関が表れていることが分かる。一方で、不正解鍵では相関係数のピークが表れていないことが確認できる。

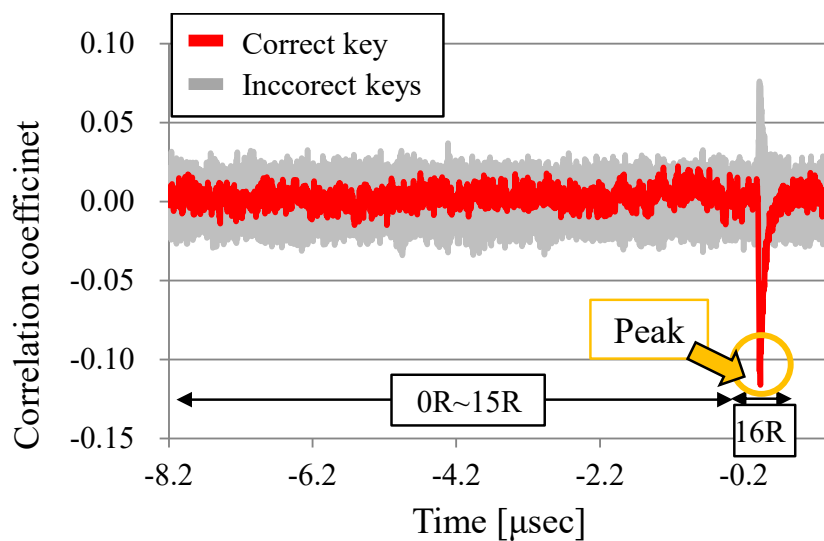


図 6.10 相関係数の比較（1段階目の電力解析）

Fig. 6.10 Comparison of the correlation coefficient (power analysis in the first stage)

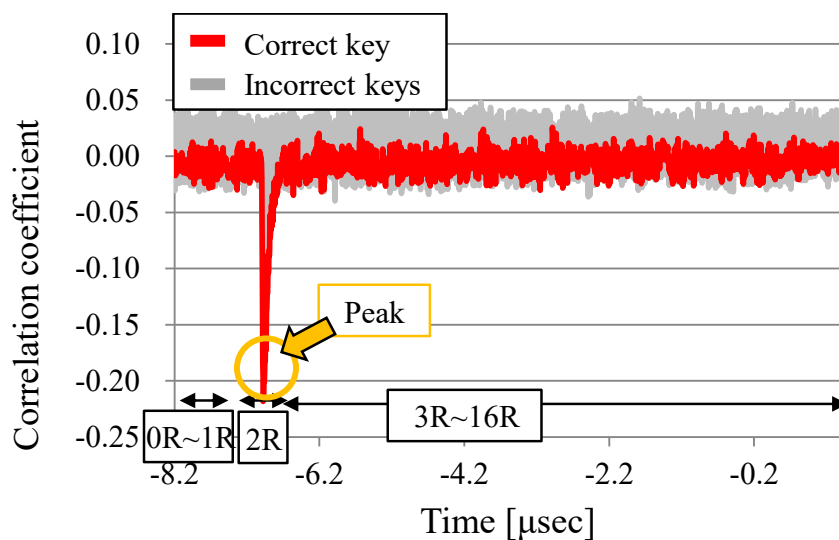


図 6.11 相関係数の比較（2段階目の電力解析）

Fig. 6.11 Comparison of the correlation coefficient (power analysis in the second stage)

6.3.4 処理時間・計算量の比較

解析の効率化手法の有効性を検証するために、処理時間と計算量について比較した。比較結果を表 6.2 に示す。ここでは、6.3.2 項の実験結果（図 6.9）において、2 段階目の電力解析では 2,000 個の波形データで全ての鍵 K_0 の推定に成功していることから、2,000 個の波形データを用いた場合の処理時間を測定した。表 6.2 に示すように、解析の効率化手法を用いた場合、処理時間は 288 分であることが確認できる。一方で、解析の効率化手法を用いない場合（表 6.2 の(i)）は、計算量が多く処理時間を測定することができなかった。具体的には、表 6.2 に示すように、1 つの MixColumn に関して 2^{44} 通りの計算が必要であるため、各 MixColumn に対応させて、合計で $2^{44} \times 4 \approx 7.0 \times 10^{13}$ 通りの計算を必要とする。一方で、解析の効率化手法を用いる場合では、その計算量は 16,448 通りである。以上のように、解析の効率化手法を導入することで、計算量の削減に成功しており、提案手法が有効であると考えられる。

6.4 結言

本章では、Midori に対する電力解析手法を提案し、FPGA を用いた評価実験によってその有効性を示した。提案した電力解析手法では、16R と 2R を対象とした 2 段階の階層的な電力解析手法を導入することで、Midori の秘密鍵を全て推定する。また、提案した電力解析手法では Midori の構造に着目した解析の効率化手法を導入することで、計算量を大幅に削減した。

以上より、提案する電力解析手法を用いることで、Midori の耐タンパ性を評価できることを示した。

表 6.2 処理時間と計算量の比較

Table 6.2 Comparison of the processing time and the computational amount

	With the improvement method		Without the improvement method	
	Time [min]	Computational amount	Time [min]	Computational amount
Power analysis in the second stage	268	$(2^{12} + 2^4) \times 4 = 16,448$	(i)	$2^{44} \times 4 \approx 7.0 \times 10^{13}$

参考文献

- [1] 野崎佑典, 吉川雅弥, “低消費電力軽量暗号 Midori に対する階層的電力解析とその評価,” 電気学会論文誌 C, vol. 138, no. 12, pp. 1455–1463, Dec. 2018.
- [2] S. Banik, A. Bogdanov, T. Isobe, K. Shibutani, H. Hiwatari, T. Akishita, and F. Regazzoni, “Midori: A Block Cipher for Low Energy,” Proc. ASIACRYPT 2015, LNCS vol. 9453, pp. 411–436, Springer, Dec. 2015
- [3] CRYPTREC, “CRYPTREC Cryptographic Technology Guideline (Lightweight Cryptography),” Mar. 2017.
<http://www.cryptrec.go.jp/report/cryptrec-gl-0001-2016-e.pdf>
- [4] E. Brier, C. Clavier, and F. Olivier, “Correlation Power Analysis with a Leakage Model,” Proc. 6th Int. Workshop on Cryptographic Hardware and Embedded Systems (CHES 2004), LNCS vol. 3156, pp. 16–29, Springer-Verlag, Aug. 2004.
- [5] “The National Institute of Advanced Industrial Science and Technology Research Institute for Secure Systems: Evaluation Environment for Side-channel Attacks,”
<http://www.risec.aist.go.jp/project/sasebo>

第 7 章

Minalpher に対する電力解析手法

7.1 緒言

本章では，Minalpher の耐タンパ性評価に必要な電力解析手法を提案する [1]。Minalpher [2][3]は，軽量暗号の 5 大要素の 1 つである暗号化と認証を同時に実現可能なアルゴリズムである。また，Minalpher は CRYPTREC 軽量暗号 WG の暗号技術ガイドライン（軽量暗号）[4]においても評価対象となっている暗号アルゴリズムであり，スマート農業で利用されるセンサなどへの利用が期待されている [4]。Minalpher のアルゴリズムの詳細については，付録（A.5 節）で述べる。

認証付暗号を対象とした電力解析に関して，これまでに AES-GCM [5]を対象とした電力解析が報告されている [6]。一方で，Minalpher に対する電力解析に関する研究は報告されていない。本章で提案する電力解析手法では，Minalpher のアルゴリズムを考慮し，階層的に CPA [7]を適用する。また，周波数領域での解析を導入することで解析効率を向上させる。

7.2 提案する電力解析手法

提案する電力解析手法では，Minalpher がオフセット値生成部分で秘密鍵を用いた処理を行っているため，オフセット値生成部分を対象とする。この電力解析では，秘密鍵の推定に複数のオフセット値が必要となる。しかし，Minalpher の暗号アルゴリズムの構成上，オフセット値は外部に出力されない。すなわち，解析において，オフセット値を直接取得することはできない。そこで提案する電力解析手法では，オフセット値を推定するための電力解析を導入する。すなわち，2 段階での電力解析を行う。

本章は[1]で発表された内容から構成される。

一方で、3.2.2 項でも述べた通り、電力解析では時間領域ではなく周波数領域で電力解析を行うことで、解析効率が向上することが報告されている[8]–[10]。そこで提案する電力解析手法では、2 段階での電力解析を周波数領域で適用することで、時間軸上での位相ずれ誤差による影響などを抑え、高効率な解析を実現させる。

まず 7.2.1 項から 7.2.3 項では、2 段階での電力解析手法について、時間領域で適用する場合を例に説明する。そして 7.2.4 項では、2 段階での電力解析手法の周波数領域での適用方法について説明する。

7.2.1 2 段階での電力解析手法

2 段階での電力解析手法は、図 7.1 に示すように 1 段階目の電力解析と 2 段階目の電力解析で構成する。1 段階目の電力解析では、暗号文生成部分を対象として、オフセット値を推定する。このオフセット値の推定では、既知の暗号文と消費電力波形のペアを D_1 個使用して、電力解析を行う。

次に、2 段階目の電力解析では、1 段階目の電力解析を D_2 回行い、 D_2 個のオフセット値を取得する。そして、 D_2 個の消費電力波形とオフセット値のペアを使用して、オフセット値生成部分に対して電力解析を行い、秘密鍵 K を推定する。

7.2.2 1 段階目の電力解析手法

Minalpher の暗号文生成部分を対象とした 1 段階目の電力解析について、図 7.2 を用いて説明する。図 7.2 に示すように、P 関数の 17.5R を対象として CPA [7]をベースとした解析を行う。具体的には、17R 計算終了後の暗号中間値 x を計算する。この計算は、既知の暗号文 $C[1]$ とオフセット値 ϕ_1 の予測値を利用して、式(7.1)を用いて行う。ここで、 $S()$ は S 関数による処理 (A.5 節, 表 A.10 を参照) を、 $T()$ は T 関数による処理 (A.5 節, 表 A.11 を参照) を表している。

$$x = S(T(C[1] \oplus \phi_1)) \quad (7.1)$$

また、 $S()$ 、 $T()$ はそれぞれ 4bit 単位で計算できる。したがって、オフセット値の予測値には $2^4 = 16$ 通りの候補を試す。そして、式(7.2)から暗号中間値 x と暗号文 $C[1]$ とのハミング距離 h を計算する。ここで、 $HD(a, b)$ は 2 つの値 a と b のハミング距離を計算する関数である。

$$h = HD(x, C[1]) \quad (7.2)$$

オフセット値の推定では、ハミング距離 h と消費電力値 w からピアソンの相関係数 ρ を式(3.1)から計算する。そして、ピアソンの相関係数 ρ を最大とするオフセット値 φ_1 の予測値を正解値として推定する。また、オフセット値の導出に関して、全てのオフセット値である 256bit の値ではなく、秘密鍵 K に関連する部分である 128bit の値のみを導出する。

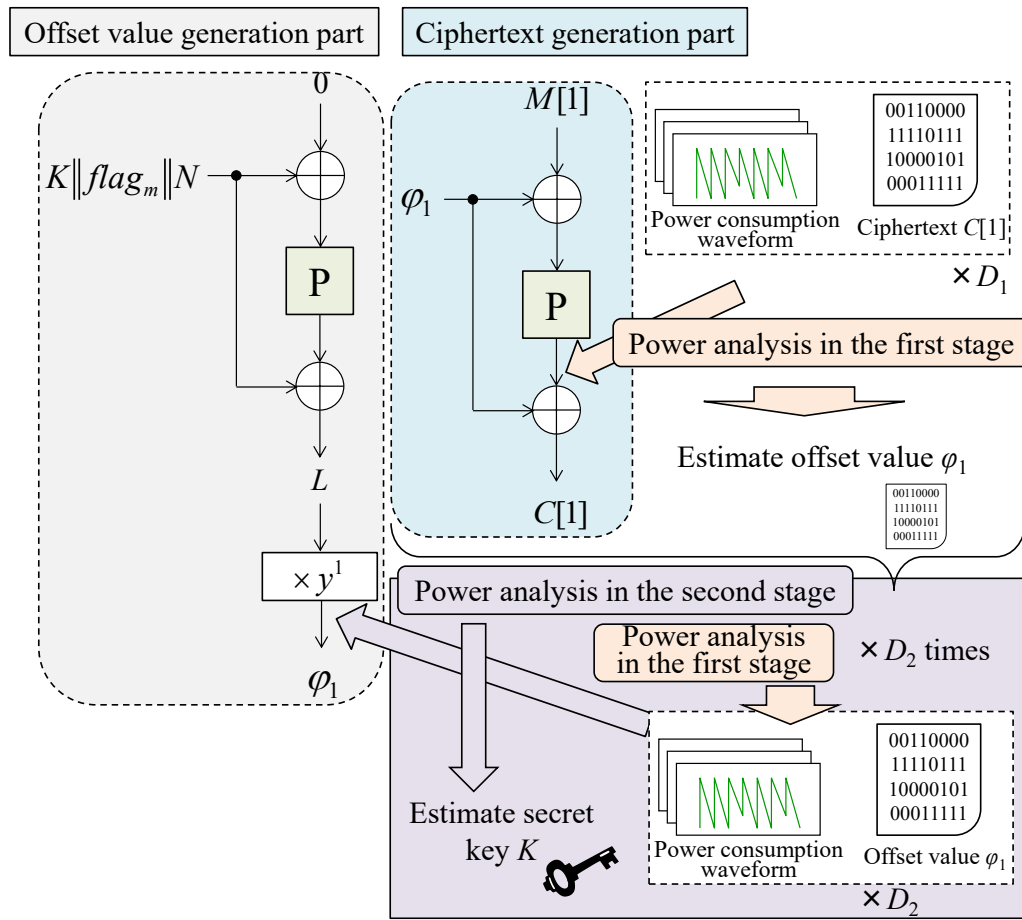


図 7.1 2 段階での電力解析手法

Fig.7.1 Power analysis method with two stages

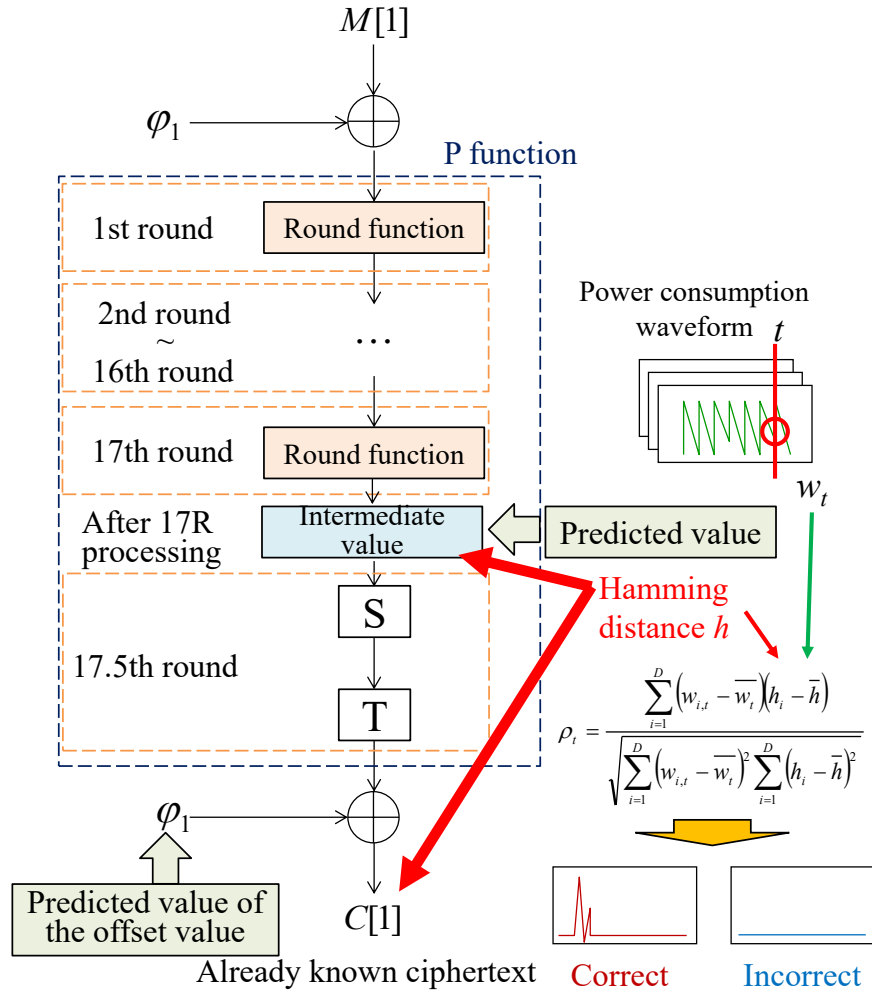


図 7.2 1 段階目の電力解析手法

Fig.7.2 Power analysis method in the first stage

7.2.3 2 段階目の電力解析手法

Minalpher のオフセット値生成部分を対象とした 2 段階目の電力解析について、図 7.3 を用いて説明する。図 7.3 に示すように、1 段階目の電力解析と同様にして、P 関数の 17.5R を対象とする。したがって、17R 計算終了後の暗号中間値 x と L とのハミング距離を計算する。このとき、 L は 1 段階目の電力解析で推定したオフセット値 ϕ_1 を利用して計算する。 L の計算方法を図 7.4 に示す。図 7.4 に示す処理から、オフセット値 ϕ_1 から秘密鍵 K に関連する 128bit 分の L の値を計算できる。また、この 1 段階目の電力解析は 2 段階目の電力解析で使用するデータ数である D_2 回分行う。そして、対象とする暗号中間値は L と秘密鍵 K の予測値を用いて、式(7.3)で計算する。また、式(7.3)の計算は 4bit 単位で行う。そし

て，暗号中間値 x と L とのハミング距離 h を算出する。

$$x = S(T(L \oplus K)) \quad (7.3)$$

秘密鍵 K の推定では，1 段階目の電力解析と同様にして，算出したハミング距離 h と消費電力値 w からピアソンの相関係数 ρ を，式(3.1)で計算する。そして，ピアソンの相関係数 ρ を最大とする秘密鍵 K の予測値を正解鍵として推定する。

以上より， D_1 個のデータによる 1 段階目の電力解析を D_2 回， D_2 個のデータによる 2 段階目の電力解析を 1 回実行することで，秘密鍵 K の全てを推定することができる。

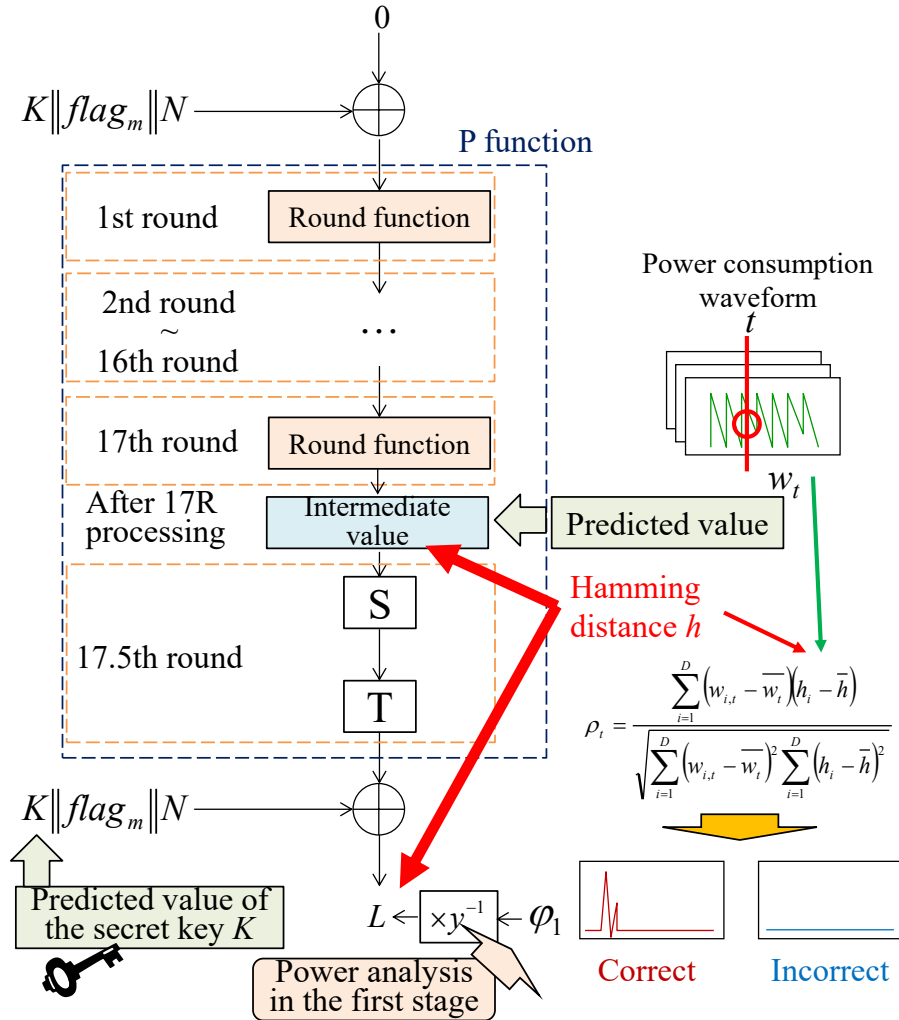


図 7.3 2 段階目の電力解析手法

Fig.7.3 Power analysis in the second stage

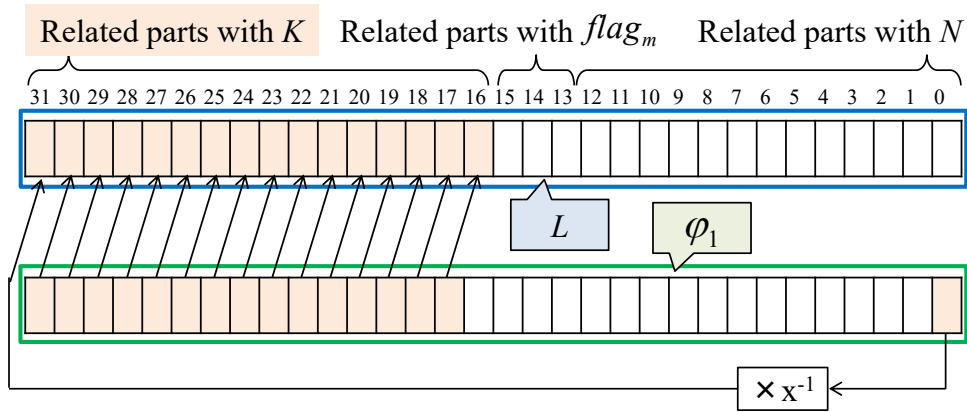


図 7.4 L の計算

Fig.7.4 Calculation of L

7.2.4 2 段階での電力解析の周波数領域での適用方法

提案する電力解析手法では，周波数領域で 2 段階での電力解析を適用することで，解析効率を向上させる。周波数領域での 2 段階の電力解析の概要を図 7.5 に示す。図 7.5 に示すように解析では，消費電力波形ではなく，消費電力波形を DFT して取得したパワースペクトル波形を使用する。このパワースペクトル波形の取得方法を図 7.6 に示す。図 7.6 に示すように，時間領域での 2 段階の電力解析で対象としていた範囲の消費電力値を DFT することで，パワースペクトル波形を算出する。

そして，導出したハミング距離 h と算出したパワースペクトル s からピアソンの相関係数 ρ を求める。ピアソンの相関係数 ρ の計算では，時間領域では消費電力値 w を用いるのに対して，周波数領域ではパワースペクトル s を利用する。この計算式を式(7.4)に示す。

$$\rho_f = \frac{\sum_{i=1}^D (s_{i,f} - \bar{s}_f)(h_i - \bar{h})}{\sqrt{\sum_{i=1}^D (s_{i,f} - \bar{s}_f)^2 \sum_{i=1}^D (h_i - \bar{h})^2}} \quad (7.4)$$

ここで， f は任意の周波数帯， $\bar{s}_f$ はパワースペクトル s の平均値である。最後に，解析ではピアソンの相関係数 ρ を最大とするオフセット値や秘密鍵の予測値を正解として推定する。

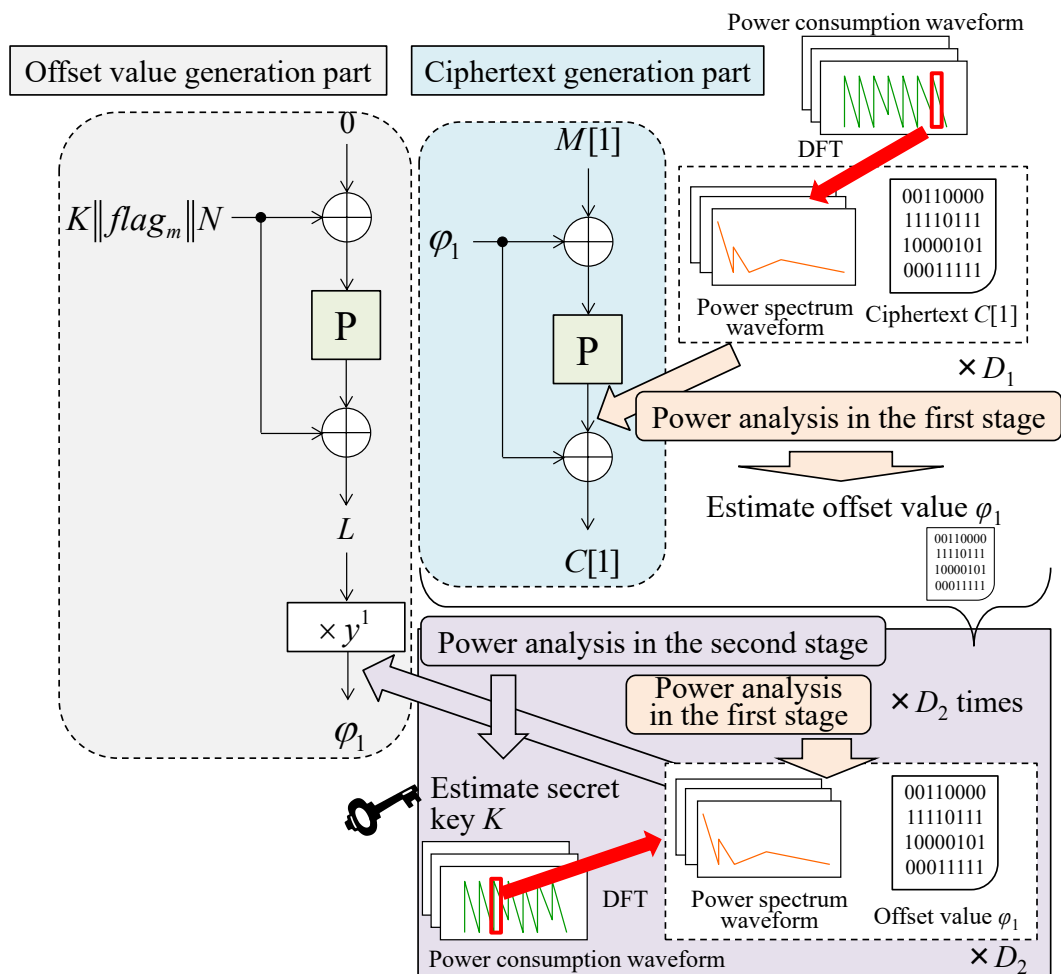


図 7.5 周波数領域での 2 段階の電力解析手法

Fig.7.5 Power analysis method with two stages in frequency domain

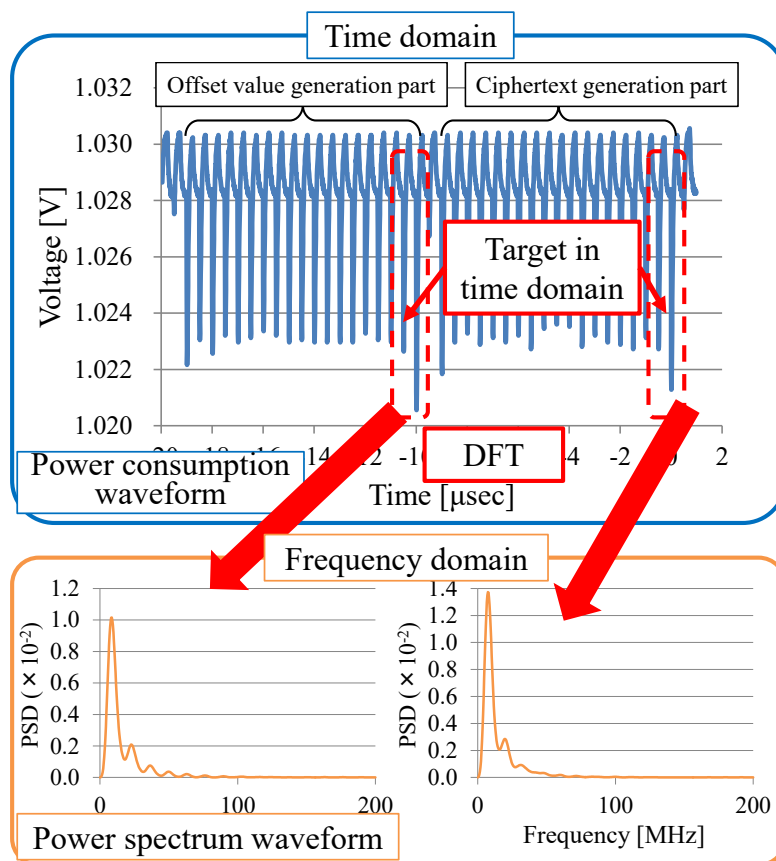


図 7.6 パワースペクトル波形の算出方法

Fig.7.6 Power spectrum waveform generation method

7.3 評価実験

7.3.1 実験環境

評価実験は、図 7.7 と表 7.1 に示す環境で実施した。評価ボードには SASEBO-GII [11] を使用し、SASEBO-GII 上の FPGA Viretux-5 に Minalpher を FPGA 実装した。ナンス (Nonce) や平文、秘密鍵は、PC 上で乱数を用いて生成したものを使用した。また測定した消費電力波形は、暗号 LSI の消費電流を $1[\Omega]$ のシャント抵抗で測定した電圧波形である。

7.3.2 実験結果

評価実験において、まず 7.3.1 項で述べた環境で予備実験を行った。予備実験の詳細については、7.3.3 項から 7.3.5 項で述べる。この予備実験では、10,000 波形のデータを使用して 1 段階目の電力解析と 2 段階目の電力解析をそれぞれ周波数領域で行った。そして、正

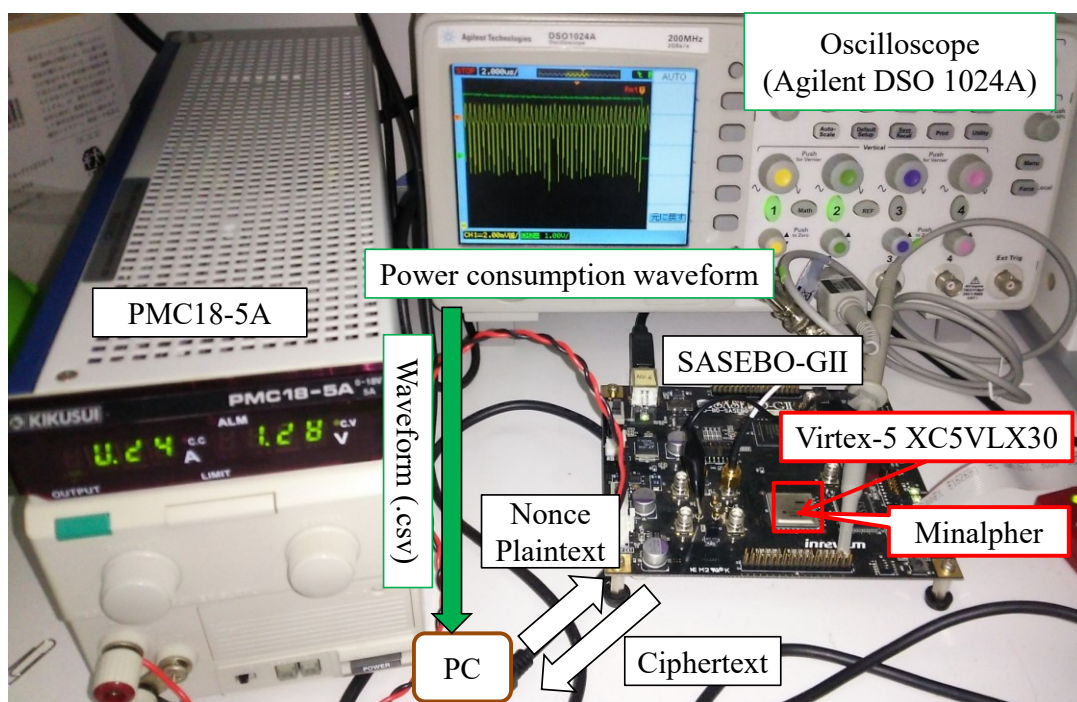


図 7.7 評価システム

Fig.7.7 Evaluation system

表 7.1 実験環境

Table 7.1 Experimental condition

Encryption algorithm	Minalpher
Evaluation board	SASEBO-GII [11]
FPGA	Virtex-5 XC5VLX30
Implementation tool	Xilinx ISE Design Suite 14.1
Oscilloscope	Agilent DSO 1024A
Sampling rate [Gsa/sec]	2
Power supply	KIKUSUI PMC18-5A
PC	HP ProBook 6570b
OS	Windows 7 Professional
Memory [GB]	8
CPU	Intel Core i7-3520M
Programming language	MATLAB 2013b

解と不正解における相関係数を算出した。その結果、1 段階目の電力解析においては 4–14MHz 帯に、2 段階目の電力解析においては 3–20MHz 帯に高い相関が観測された。

まず、1 段階目の電力解析の結果を図 7.8 に示す。この実験では、暗号文生成部分を対象に 1 段階目の電力解析を行い、オフセット値を推定した。また、実験は周波数領域で電力解析を行い、解析には予備実験で高い相関が得られた 4–14MHz 帯の値を使用した。図 7.8 の横軸は解析に使用した波形データの数を、縦軸は解析に成功したオフセット値の bit 数を表している。図 7.8 の結果から、3,000 波形を用いることで、全てのオフセット値の推定に成功していることが分かる。したがって、1 段階目の電力解析が有効であることが分かる。

次に、2 段階目の電力解析の結果を図 7.9 に示す。この実験では、オフセット値生成部分を対象に電力解析を行い、秘密鍵を解析した。このとき、実験の簡単化のため、解析に使用するオフセット値は既知の値として扱う。また、実験では周波数領域で電力解析を行い、解析には予備実験（7.3.4 項参照）で高い相関が得られた 3–20MHz 帯の値を使用した。図 7.9 の縦軸は解析に成功した秘密鍵の bit 数を表している。図 7.9 の結果から、3,000 波形を用いることで、全ての秘密鍵の解析に成功し、2 段階目の電力解析が有効であることが確認できる。

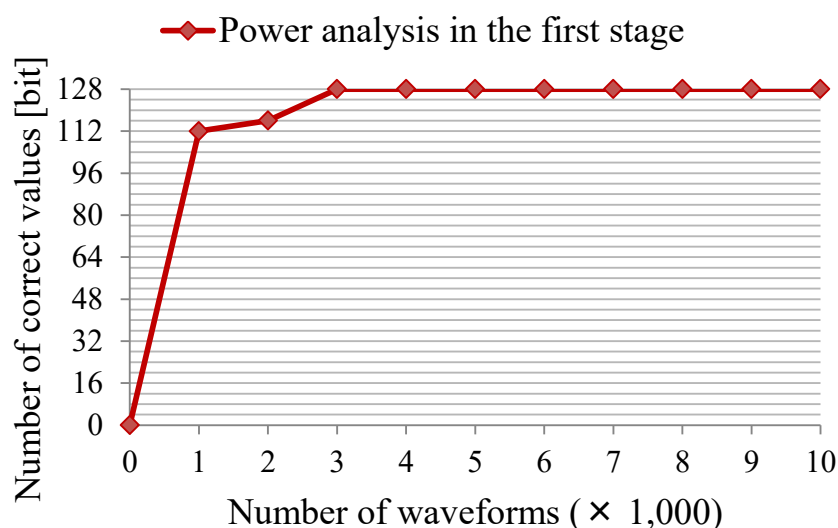


図 7.8 実験結果（1 段階目の電力解析）

Fig.7.8 Experimental result of the power analysis in the first stage

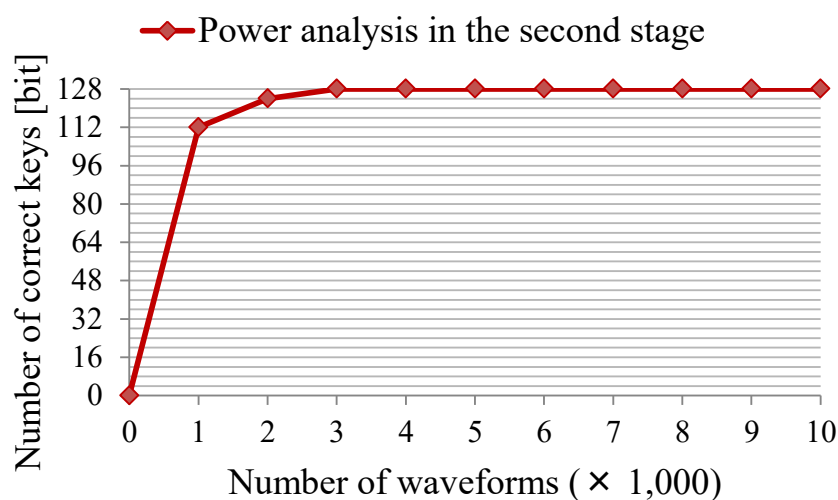


図 7.9 実験結果（2 段階目の電力解析）

Fig.7.9 Experimental result of the power analysis in the second stage

表 7.2 処理時間の比較

Table 7.2 Comparison of analytical time

	Analysis in the first stage	Analysis in the second stage
Acquisition of waveforms [sec]	671	671
Discrete Fourier transform [sec]	254	187
Analysis [sec]	305	292
Total [sec]	1,230	1,150

また、解析に必要な処理時間を表 7.2 に示す。表 7.2 は、1 段階目の電力解析と 2 段階目の電力解析の実行にかかった処理時間を示している。また、1 段階目の電力解析と 2 段階目の電力解析は、それぞれ 3,000 波形で解析に成功しているため、3,000 波形を用いて解析を行った場合の結果を示している。表 7.2 に示すように、1 段階目の電力解析は 1,230 [sec]、2 段階目の電力解析では 1,150 [sec] で解析が可能であることが分かる。

以上より、提案する電力解析手法によって Minalpher の耐タンパ性を評価することができる。

7.3.3 周波数領域での電力解析の有効性評価

周波数領域での電力解析の有効性について評価するために、周波数領域と時間領域での解析結果を比較した。比較結果を図 7.10 と図 7.11 に示す。図 7.10 と図 7.11 はそれぞれ、1 段階目の電力解析と 2 段階目の電力解析の結果を示している。

図 7.10 の結果から、1 段階目の電力解析において、周波数領域では 3,000 波形でオフセット値の解析に成功しているのに対して、時間領域では解析に 5,000 波形のデータを使用していることが分かる。また図 7.11 の結果から、2 段階目の電力解析において、周波数領域では 3,000 波形で秘密鍵の解析に成功しているのに対して、時間領域では解析に 4,000 波形のデータを使用していることが分かる。したがって、解析に必要な波形数が減少し、解析効率が向上していることから、周波数領域での電力解析は有効であると考えられる。

7.3.4 相関係数の比較

正解の値と不正解の値における相関係数について比較した。比較結果をそれぞれ図 7.12 と図 7.13 に示す。図 7.12 と図 7.13 はそれぞれ、1 段階目の電力解析と 2 段階目の電力解析における比較結果を示している。また、図 7.12 と図 7.13 の横軸は周波数帯域を、縦軸は相関係数を表している。図 7.12 と図 7.13 に示すように、低周波数帯域において、正解の値で相関係数のピークが表れていることが分かる。特に、図 7.12 では 4–14MHz の周波数帯域

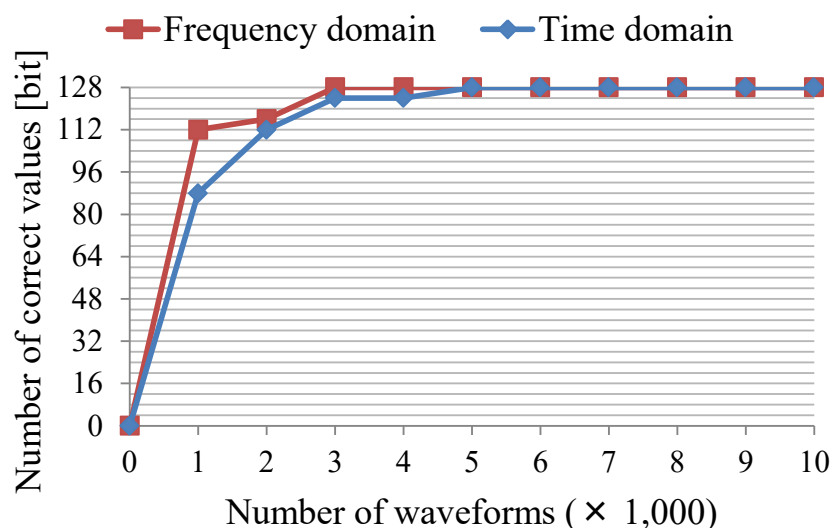


図 7.10 周波数領域と時間領域での解析結果の比較（1 段階目の電力解析）

Fig.7.10 Comparison of frequency domain and time domain in the first stage

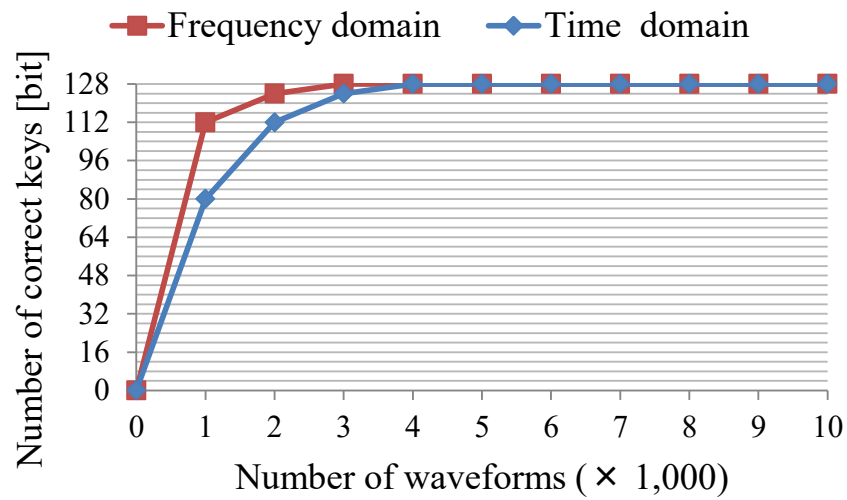


図 7.11 周波数領域と時間領域での解析結果の比較（2 段階目の電力解析）

Fig.7.11 Comparison of frequency domain and time domain in the second stage

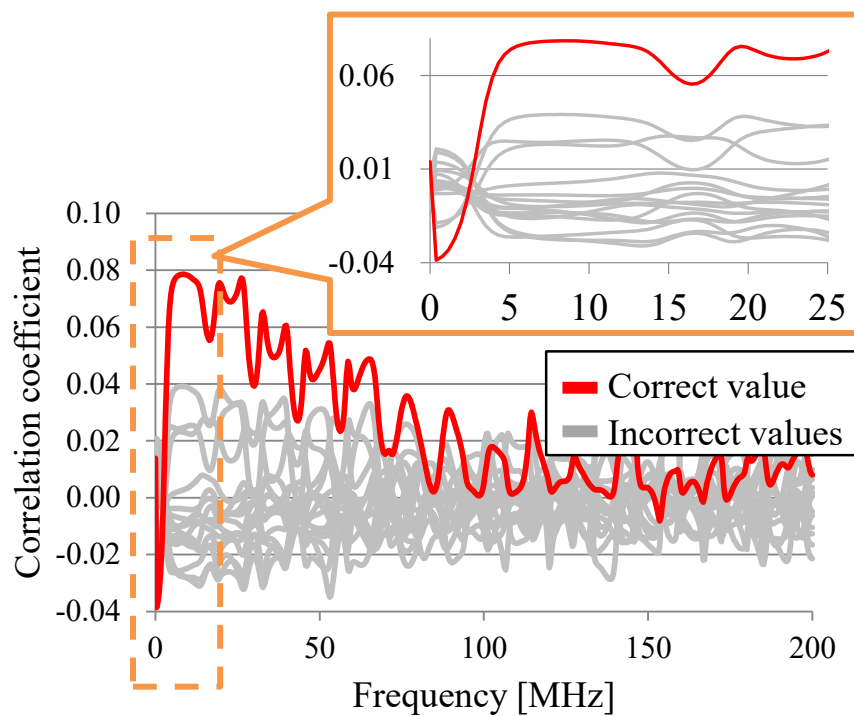


図 7.12 相関係数の比較（1 段階目の電力解析）

Fig.7.12 Comparison of the correlation coefficient of the power analysis in the first stage

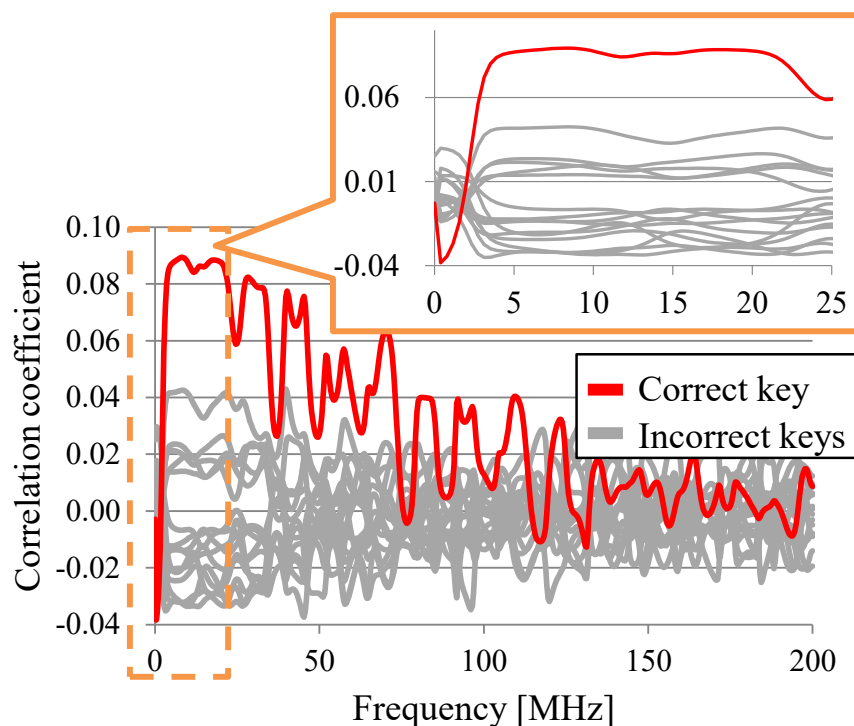


図 7.13 相関係数の比較（2 段階目の電力解析）

Fig.7.13 Comparison of the correlation coefficient of the power analysis in the second stage

で、図 7.13 では 3–20MHz の周波数帯域で相関係数が高くなっていることが確認できる。

7.3.5 使用する周波数帯域が解析に与える影響についての評価

使用する周波数帯域が解析結果へ与える影響について評価した。比較結果を図 7.14 と図 7.15 に示す。図 7.14 と図 7.15 はそれぞれ、1 段階目の電力解析と 2 段階目の電力解析における比較結果を示している。

まず図 7.14 は、1 段階目の電力解析において 4–14MHz 帯で高い相関が表れたことから、周波数帯域をそれぞれ 0–4MHz、4–14MHz、14–200MHz、0–200MHz と変化させて実験を行った。図 7.14 に示すように、4–14MHz 帯と 0–200MHz 帯を用いた解析では、3,000 波形でオフセット値の解析に成功した。一方で、4–14MHz 帯と 0–200MHz 帯以外の周波数帯域では、3,000 波形を用いた場合でもオフセット値を全て解析することはできなかった。また、波形数が 3,000 波形よりも少ない場合において、4–14MHz 帯を用いた解析は、0–200MHz 帯を用いた解析よりもオフセット値の正解数が多いことが確認できる。したがって、1 段階目の電力解析では 4–14MHz 帯の周波数帯域を使用することで最も効率的な解析が可能

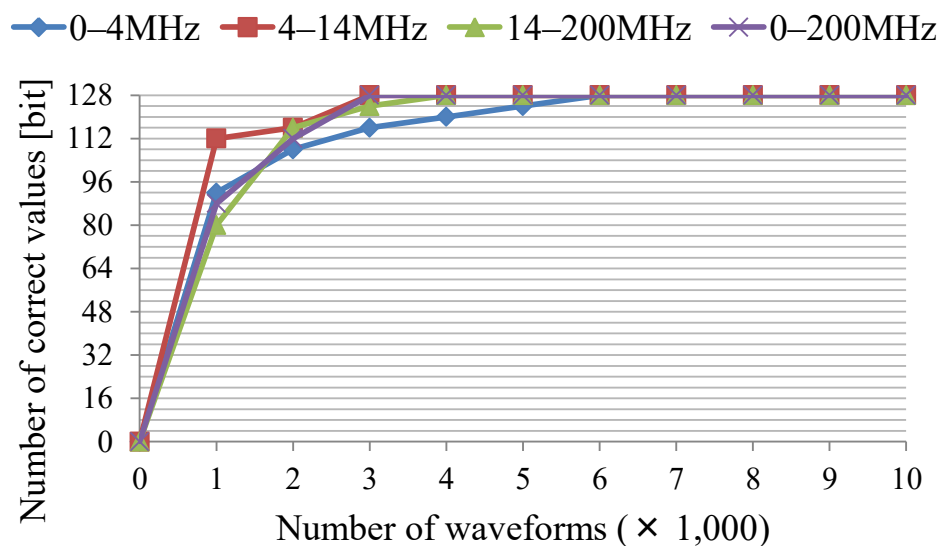


図 7.14 周波数帯域の違いによる解析結果（1段階目の電力解析）

Fig.7.14 Comparison results in the different frequency band of the power analysis in the first stage

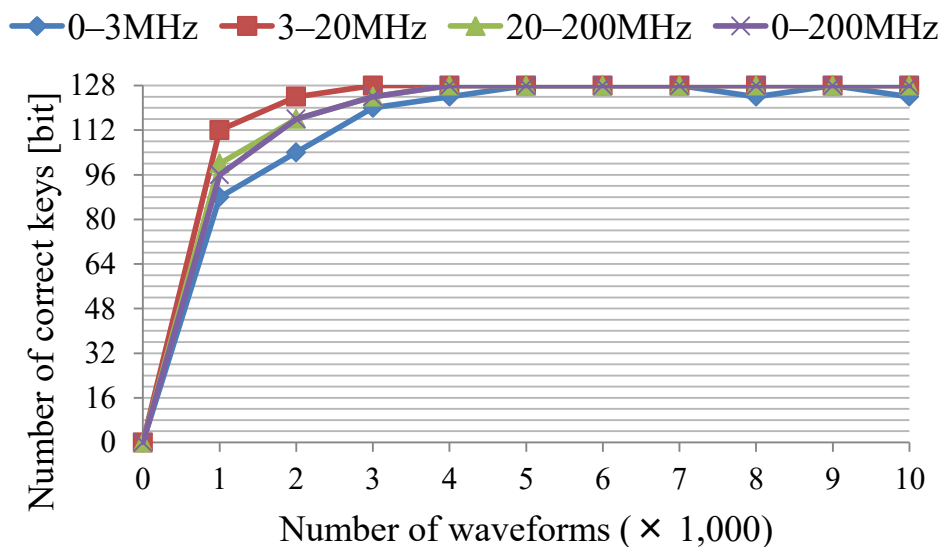


図 7.15 周波数帯域の違いによる解析結果（2段階目の電力解析）

Fig.7.15 Comparison results in the different frequency band of the power analysis in the second stage

である。これは 4–14MHz 帯のパワースペクトルに高い強度があり、強度が高い部分に解析に有意な情報が含まれているためだと考えられる。

次に図 7.15 は、2 段階目の電力解析において 3–20MHz 帯で高い相関が表れたことから、周波数帯域をそれぞれ 0–3MHz, 3–20MHz, 20–200MHz, 0–200MHz と変化させて実験を行った。図 7.15 に示すように、3–20MHz 帯を用いた解析では 3,000 波形で全ての秘密鍵の解析に成功した。一方で、3–20MHz 帯以外の周波数帯域では、3,000 波形を用いた場合でも全ての秘密鍵を解析することはできなかった。ここで、0–3MHz 帯の解析の 8,000 波形と 10,000 波形で正解鍵数が低下しているのは、0–3MHz 帯は解析に有意な情報が少なく、8,000 波形や 10,000 波形では、解析におけるノイズ成分(注目する 4bit のデータ以外の 252bit のデータの遷移に付随して生じる消費電力成分)を平準化できていないためだと考えられる。そのため、解析に使用する波形数を更に増やすか、解析に有意な情報を含む周波数帯域(3–20MHz 帯など)を用いて解析を行うことで、解析効率の向上が期待できる。

以上より、2 段階目の電力解析では 3–20MHz 帯の周波数帯域を使用することで最も効率的な解析を行うことができる。これは 3–20MHz 帯のパワースペクトルに高い強度があり、強度が高い部分に解析に有意な情報が含まれているためだと考えられる。

7.4 結言

本章では、Minalpher に対する電力解析手法を提案し、FPGA を用いた評価実験によってその有効性を示した。提案した電力解析手法では Minalpher のオフセット値生成部分と暗号文生成部分に対して、それぞれ階層的に電力解析を行う。また、提案する電力解析手法を時間領域だけでなく、周波数領域へと展開することで、解析効率を向上させた。

以上より、提案した電力解析手法を用いることで、Minalpher の耐タンパ性を評価できることを示した。

参考文献

- [1] 野崎佑典, 吉川雅弥, “改ざん検知暗号 Minalpher に対する階層的電力解析手法とその評価,” 情報処理学会論文誌コンシューマ・デバイス&システム (CDS), vol. 7, no.

- 2, pp. 125–134, May 2017.
- [2] Y. Sasaki, Y. Todo, K. Aoki, Y. Naito, T. Sugawara, Y. Murakami, M. Matsui, and S. Hirose, “Minalpher v1.1,” Aug. 2015. <http://info.isl.ntt.co.jp/crypt/minalpher/files/minalpherv11.pdf>
- [3] 佐々木悠, 藤堂洋介, 青木和麻呂, 内藤祐介, 菅原 健, 村上ユミコ, 松井 充, 廣瀬勝一, 高橋克己, “改ざん検知暗号 Minalpher,” 2015 年暗号と情報セキュリティシンポジウム講演論文集, 2E-1, pp. 1–4, Jan. 2015.
- [4] CRYPTREC, “CRYPTREC Cryptographic Technology Guideline (Lightweight Cryptography),” Mar. 2017.
<http://www.cryptrec.go.jp/report/cryptrec-gl-0001-2016-e.pdf>
- [5] M. Dworki, “Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC,” National Institute of Standards and Technology Internal Special Publication 800-38D, Nov. 2007.
- [6] S. Belaïd, P. -A. Fouque, and B. Gérard, “Side-Channel Analysis of Multiplications in $GF(2^{128})$ Application to AES-GCM,” Proc. ASIACRYPT 2014, LNCS vol. 8874, pp. 306–325, Springer, Dec. 2014.
- [7] E. Brier, C. Clavier, and F. Olivier, “Correlation Power Analysis with a Leakage Model,” Proc. 6th Int. Workshop on Cryptographic Hardware and Embedded Systems (CHES 2004), LNCS vol. 3156, pp. 16–29, Springer-Verlag, Aug. 2004.
- [8] C. H. Gebotys, S. Ho, and C. C. Tiu, “EM Analysis of Rijndael and ECC on a Wireless Java-Based PDA,” Proc. 7th Int. Workshop on Cryptographic Hardware and Embedded Systems (CHES 2005), LNCS vol. 3659, pp. 250–264, Springer-Verlag, Aug. 2005.
- [9] 菅原 健, 本間尚文, 林 優一, 水木敬明, 青木孝文, 曾根秀昭, 佐藤 証, “周波数領域での暗号モジュールの電力解析,” 第 8 回情報科学技術フォーラム講演論文集, vol. 8, no. 4, pp. 135–138, Aug. 2009.
- [10] 佐藤隆亮, 松島大祐, 汐崎 充, 藤野 毅, 吉川雅弥, “周波数領域における部分鍵推定を用いたハイブリッド電力解析攻撃とその評価,” 電気学会論文誌 C, vol. 133, no. 7, pp. 1322–1330, Jul. 2013.
- [11] “The National Institute of Advanced Industrial Science and Technology Research Institute for Secure Systems: Evaluation Environment for Side-channel Attacks,”
<http://www.risec.aist.go.jp/project/sasebo>

第 8 章

TWINE-OTR に対する電力解析手法

8.1 緒言

本章では，TWINE-OTR の耐タンパ性評価に必要な電力解析手法を提案する [1]。TWINE-OTR [2]は，NEC によって開発された認証が可能な軽量暗号であり，小回路規模で実装することが可能であり，IoT デバイスでの利用が期待されている。また，実際に IoT センサデバイスなどに搭載されており，実用化がなされている [3]。TWINE-OTR のアルゴリズムの詳細については，付録（A.6 節）で述べる。

一方で，認証付暗号を対象とした電力解析はこれまでに AES-GCM [4]を対象としたもの [5]が報告されているが，TWINE-OTR を対象とした研究は行われていない。本章で提案する電力解析手法では，CPA [6]を組み合わせた階層的な解析を導入することで鍵を推定する。また，複数の暗号文生成部分を対象とした解析を導入することで，解析効率を向上させる。

8.2 提案する電力解析手法

提案する電力解析手法では，TWINE-OTR の秘密鍵を解析するために，CPA [6]を組み合わせた 2 段階の電力解析手法を導入する。更に，新たに TWINE-OTR の構造に着目した複数の暗号文生成部分を利用した解析を行うことで，解析効率を向上させる。

まず，8.2.1 項では 2 段階での電力解析手法の概要について説明する。そして，8.2.2 項と 8.2.3 項では 2 段階での電力解析を構成する 1 段階目の電力解析と 2 段階目の電力解析について説明する。最後に，8.2.4 項では複数の暗号文生成部分を対象とした解析について説明

本章は電子情報通信学会 Copyright © 2018 IEICE で発表された内容 [1]で構成しており，本章の図は電子情報通信学会の許可を得て再利用しています。

する。

8.2.1 2 段階電力解析の概要

詳細は付録 (A.6 節) で説明するが, TWINE-OTR で出力される暗号文 C (既知の値) は, 暗号化関数 TWINE の出力をメッセージ M (未知の値) で排他的論理和演算, すなわち, 未知の値 M によってマスクした値である。そのため, 1 段階の電力解析だけでは, 対象とするラウンド鍵を解析することができない。なぜなら, 電力解析によって推定される鍵は, メッセージによってマスクされるからである。そこで, TWINE-OTR の秘密鍵を導出するために 2 段階での電力解析を導入する。2 段階での電力解析による階層的な解析を行うことで, メッセージによるマスクを外し, 対象とする鍵を導出する。具体的に 2 段階での電力解析は, 最終ラウンド (36R) を対象とした 1 段階目の電力解析と 35R を対象とした 2 段階目の電力解析で構成する。提案する電力解析手法は, 暗号文生成部分のどの部分に対しても適用することができるが, ここでは暗号文 $C[2]$ 生成部分を対象とする場合を例に説明する。

図 8.1 に示すように, まず 1 段階目の電力解析では TWINE-OTR の 36R を対象に電力解析を行い, $m_{2i} \oplus RK_i^{36} \parallel m_{2i+1}$ ($0 \leq i \leq 7$) を推定する。次に 2 段階目の電力解析では, 1 段階目の電力解析での結果を利用して, TWINE-OTR の 35R を対象に電力解析を行い, $RK_i^{35} \parallel m_{2i}$ を推定する。 $m_{2i} \oplus RK_i^{36} \parallel m_{2i+1}$ と $RK_i^{35} \parallel m_{2i}$ を推定することで, $m_{2i} \oplus RK_i^{36}$ のメッセージ m_{2i} による排他的論理和演算を外すことができる。すなわち, 最終的にラウンド鍵 RK^{35} と RK^{36} を求めることができる。

8.2.2 1 段階目の電力解析手法

1 段階目の電力解析では, 図 8.2 に示すように TWINE-OTR の 36R に対して電力解析を行い, メッセージ m_{2i} とラウンド鍵 RK_i^{36} の排他的論理和演算の値 ($m_{2i} \oplus RK_i^{36}$) と, メッセージ m_{2i+1} を推定する。具体的には, 暗号文 c_{2i+1} と 35R 終了後の暗号中間値 x_{2i+1}^{36} のハミング距離 h を解析に用いるものとする。このとき, 暗号文 c は既知の値であるが, 暗号中間値 x_{2i+1}^{36} は未知の値である。ここで, 暗号中間値 x_{2i+1}^{36} は式(A.33)を用いて式(8.1)で計算することができる。したがって, 対象とする暗号中間値 x_{2i+1}^{36} は式(8.2)で計算できる。このとき, 式(8.2)の $m_{2i} \oplus RK_i^{36}$ と m_{2i+1} は未知の値であるため, 予測値を用いて計算する。この予測値について, $m_{2i} \oplus RK_i^{36}$ は 4bit, m_{2i+1} は 4bit であるため, 合計で 8bit の値 ($m_{2i} \oplus RK_i^{36} \parallel m_{2i+1}$) を使用する。したがって, $2^8 = 256$ 通りの計算を行う。

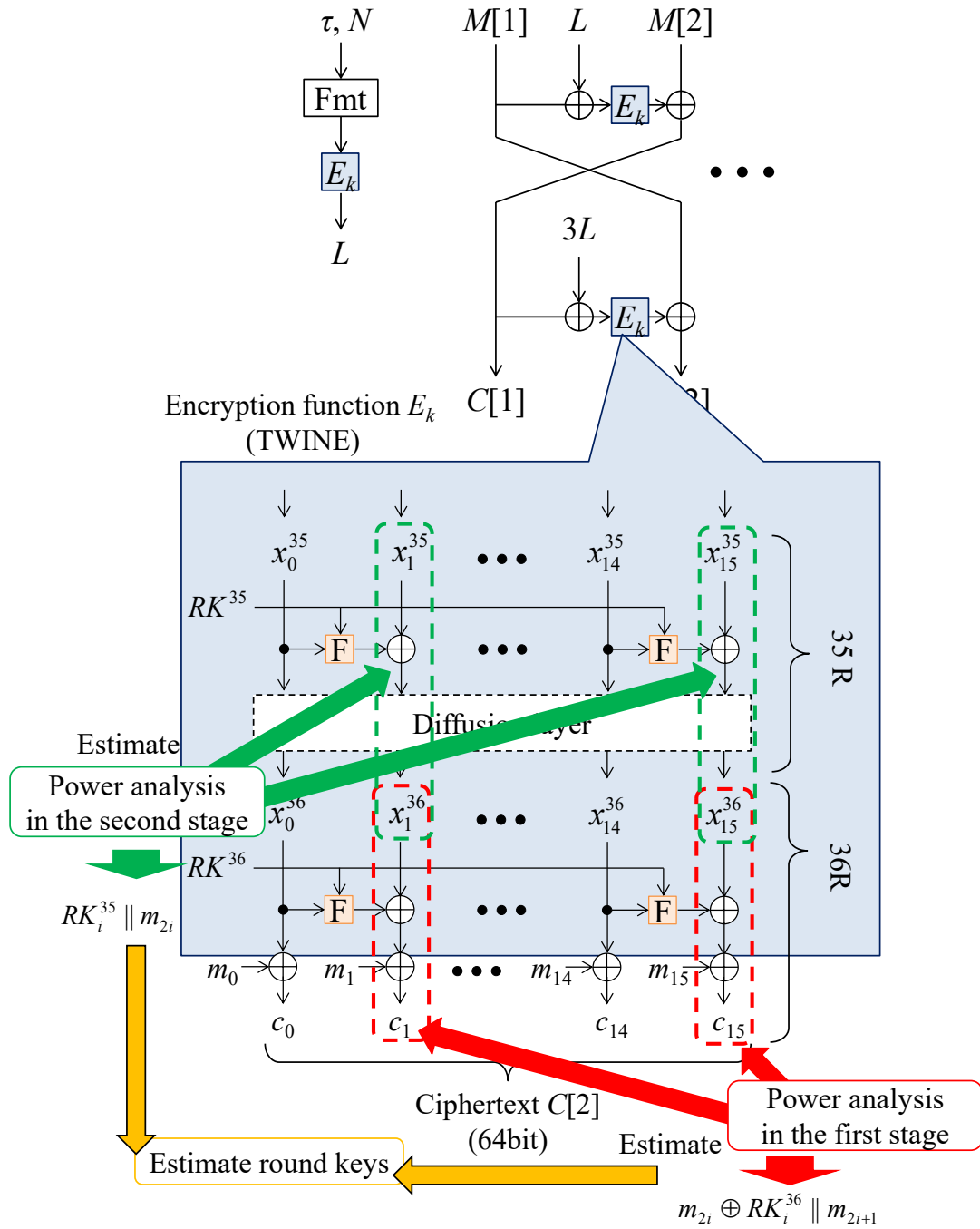


図 8.1 2 段階での電力解析手法

Fig. 8.1 Power analysis method with two stages

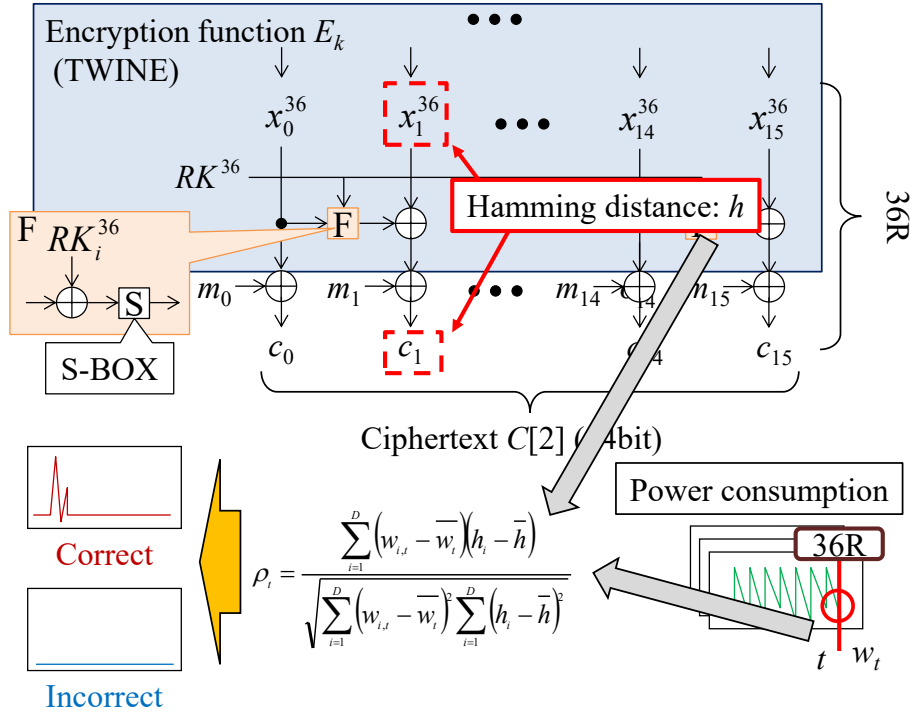


図 8.2 1 段階目の電力解析手法

Fig. 8.2 Power analysis method in the first stage

$$x_{2i}^{36} = c_{2i} \oplus m_{2i} \quad (8.1)$$

$$x_{2i+1}^{36} = S(c_{2i} \oplus m_{2i} \oplus RK_i^{36}) \oplus c_{2i+1} \oplus m_{2i+1} \quad (8.2)$$

そして、導出した暗号中間値 x_{2i+1}^{36} と既知の暗号文 c_{2i+1} とのハミング距離 h を計算する。1 段階目の電力解析では、導出したハミング距離 h と消費電力からピアソンの相関係数を式(3.1)を用いて計算し、ピアソンの相関係数を最大とする予測値を正解として推定する。

8.2.3 2 段階目の電力解析手法

2 段階目の電力解析では、図 8.3 に示すように TWINE-OTR の 35R に対して電力解析を行うことで、ラウンド鍵 RK^{35} とメッセージ m_{2i} を推定する。具体的には、図 8.3 に示すように、35R 終了後の暗号中間値 x_{2i+1}^{36} と 34R 目終了後の暗号中間値 x_{2i+1}^{35} のハミング距離 h を解析に用いるものとする。このとき、暗号中間値 x_{2i+1}^{36} は 1 段階目の電力解析から既知であるが、暗号中間値 x_{2i+1}^{35} は未知の値である。ここで、暗号中間値 x_{2i}^{35} は式(A.1), (A.2)を用いて式(8.3)で計算することができる。したがって、対象とする暗号中間値 x_{2i+1}^{35} は式(8.4)で計算できる。

$$x_{2i}^{35} = x_{\pi(2i)}^{36} \quad (8.3)$$

$$x_{2i+1}^{35} = S(x_{\pi(2i)}^{36} \oplus RK_i^{35}) \oplus x_{\pi(2i+1)}^{36} \quad (8.4)$$

このとき、式(8.4)の $x_{\pi(2i)}^{36}$ は奇数番目 $(2i+1)$ の x^{36} の値 (x_{2i+1}^{36}) を、 $x_{\pi(2i+1)}^{36}$ は偶数番目 $(2i)$ の x^{36} の値 (x_{2i}^{36}) を表している。なぜなら、拡散層の転置処理 $\pi()$ では奇数番目の値は偶数番目へ、偶数番目の値は奇数番目へ転置されるからである (A.1 節を参照)。ここで、 $x_{\pi(2i)}^{36}$ は 1 段階目の電力解析によって既知であり、 $x_{\pi(2i+1)}^{36}$ は未知の値である。また、 $x_{\pi(2i+1)}^{36}$ は式(8.2)を用いて式(8.5)で導出することができる。そして、 x_{2i+1}^{35} は式(8.4)と式(8.5)を用いて式(8.6)で計算できる。

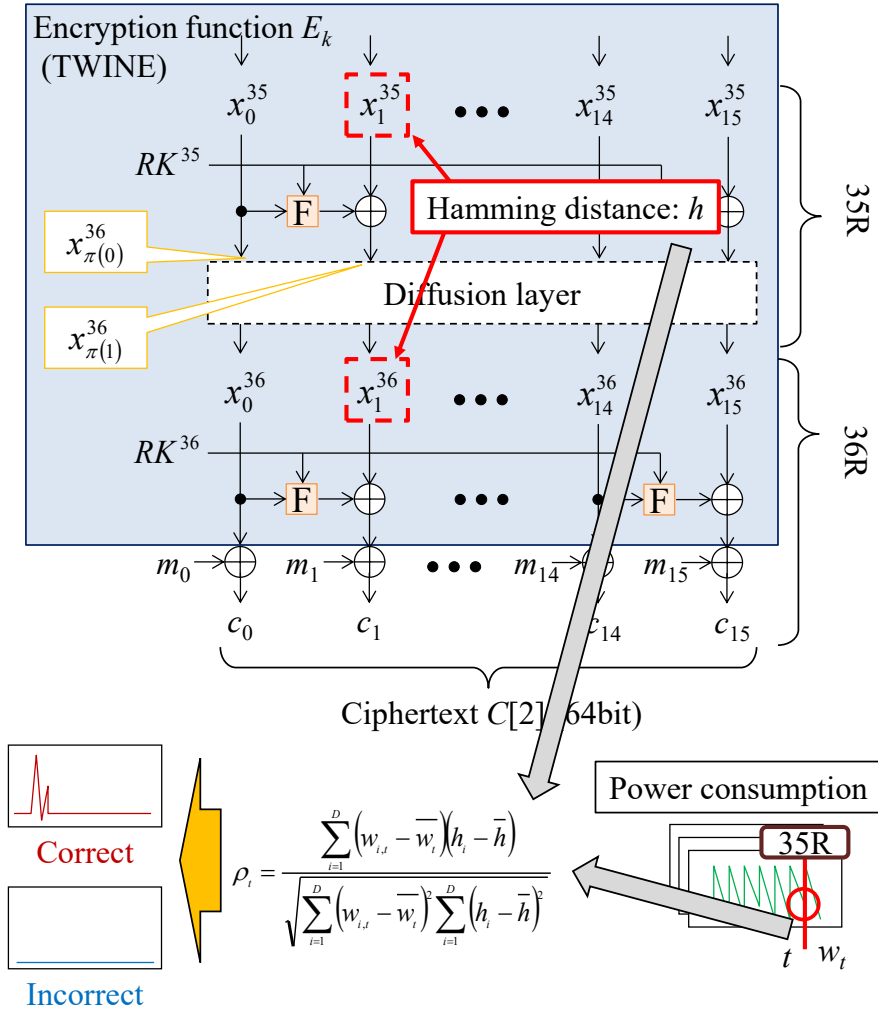


図 8.3 2 段階目の電力解析手法

Fig. 8.3 Power analysis method in the second stage

$$x_{\pi(2i+1)}^{36} = c_{\pi(2i+1)} \oplus m_{\pi(2i+1)} \quad (8.5)$$

$$x_{2i+1}^{35} = S(x_{\pi(2i)}^{36} \oplus RK_i^{35}) \oplus c_{\pi(2i+1)} \oplus m_{\pi(2i+1)} \quad (8.6)$$

このとき、式(8.6)の RK_i^{35} と m_{2i} は未知の値であるため、予測値を用いて計算する。この予測値について、 RK_i^{35} は 4bit、 m_{2i} は 4bit であるため、合計で 8bit の値 ($RK_i^{35} \parallel m_{2i}$) を使用する。したがって、 $2^8 = 256$ 通りの計算を行う。そして、導出した暗号中間値 x_{2i+1}^{35} と x_{2i+1}^{36} のハミング距離 h を計算する。2 段階目の電力解析では、式(3.1)に基づき、導出したハミング距離 h と消費電力からピアソンの相関係数を計算し、ピアソンの相関係数を最大とする予測値を正解として推定する。

したがって、2 段階目の電力解析を行うことで、 $RK_i^{35} \parallel m_{2i}$ を求めることができるため、ラウンド鍵 RK^{35} を全て求めることができる。また、2 段階目の電力解析でメッセージ m_{2i} を求めることができるため、1 段階目の電力解析の結果 ($m_{2i} \oplus RK_i^{36}$) からラウンド鍵 RK^{36} を求めることができる。以上より、ラウンド鍵 RK^{35} と RK^{36} を全て求めることができる。このように、1 段階での解析で鍵の解析が困難な場合でも、2 段階での解析を導入することで、対象とする鍵を推定することができる。

8.2.4 複数の暗号文生成部分を利用した解析

提案する電力解析手法では、TWINE-OTR の複数の暗号文生成部分 (C[1]生成部分や C[2]生成部分など) を利用した解析を導入し、より多くの情報を利用することで、解析効率を向上させる。ここでは、C[1]生成部分と C[2]生成部分を対象とした場合を例に説明する。複数の暗号文生成部分を利用した解析では、図 8.4 に示すように、2 段階での電力解析(8.2.1 項から 8.2.3 項を参照) と解析結果を用いた鍵の判定処理の 2 つの処理で構成する。

C[1]生成部分を対象とした 2 段階での電力解析 (以後、電力解析 C[1]と呼ぶ) では、図 8.4 に示すように、C[1]生成部分における 36R と 35R の処理に相当する部分の消費電力波形を解析に用いるものとする。そして、 $RK_i^{35(C[1])} \parallel m_{2i}^2$ と $m_{2i}^2 \oplus RK_i^{36(C[1])} \parallel m_{2i+1}^2$ を推定する。ここで、 $RK_i^{35(C[1])}$ と $RK_i^{36(C[1])}$ は電力解析 C[1]で推定するラウンド鍵を、 m_i^2 は 64bit のメッセージ $M[2]$ を 4bit ずつに分割した値を表している。また同様に、C[2]生成部分を対象とした 2 段階での電力解析 (以後、電力解析 C[2]と呼ぶ) では、C[2]生成部分を対象に電力解析を行い、 $RK_i^{35(C[2])} \parallel m_{2i}^1$ と $m_{2i}^1 \oplus RK_i^{36(C[2])} \parallel m_{2i+1}^1$ を推定する。ここで、 $RK_i^{35(C[2])}$ と $RK_i^{36(C[2])}$ は電力解析 C[2]で推定するラウンド鍵を、 m_i^1 は 64bit のメッセージ $M[1]$ を 4bit

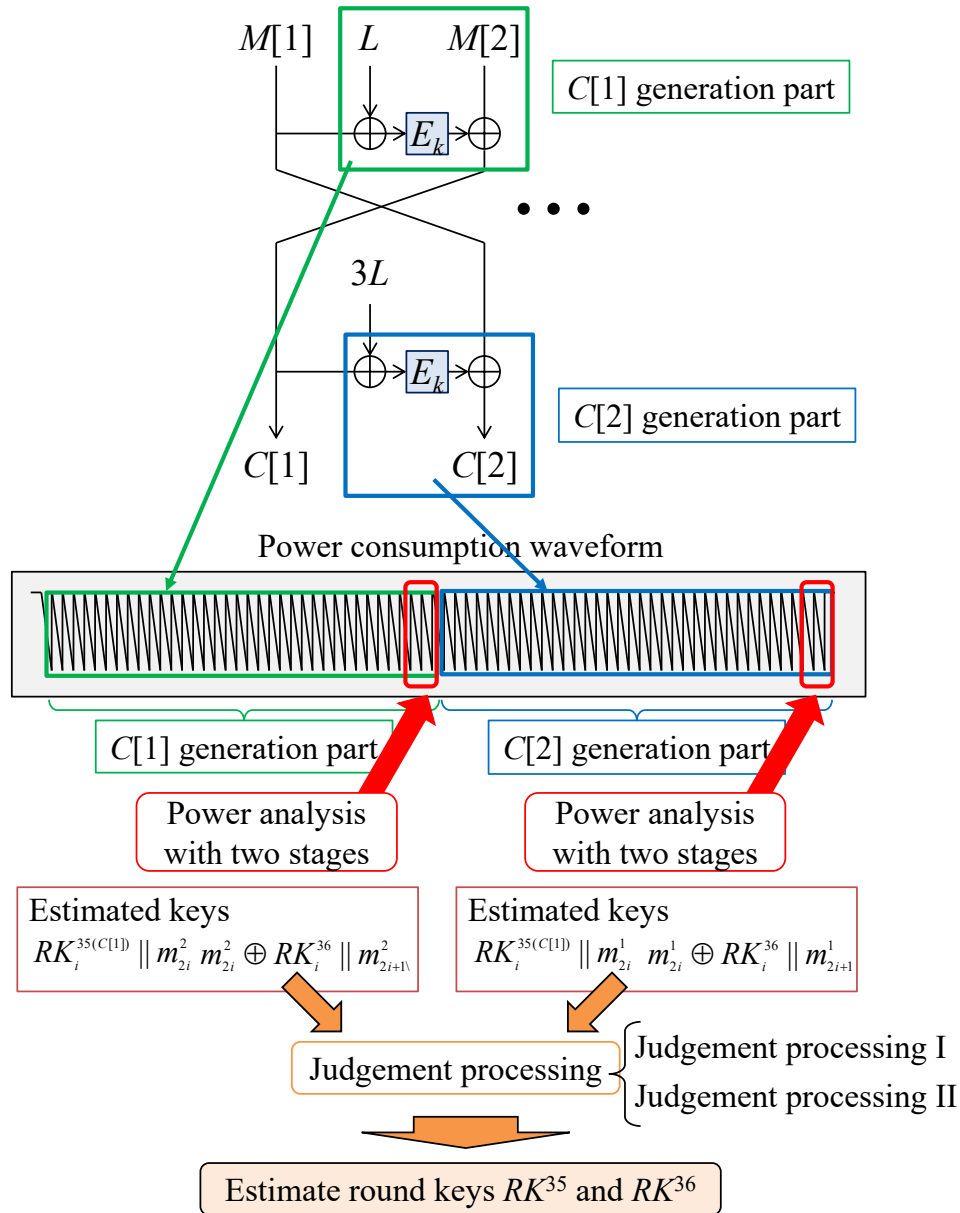


図 8.4 複数の暗号文生成部分を利用した解析

Fig. 8.4 Analysis using multiple ciphertext generation parts

ずつに分割した値を表している。

次に、鍵の判定処理について説明する。鍵の判定処理はラウンド鍵 RK^{35} の判定を行う判定処理 I と、ラウンド鍵 RK^{36} の判定を行う判定処理 II の 2 つの処理で構成する。まず、判定処理 I について図 8.5 を用いて説明する。図 8.5 に示すように、判定処理 I ではそれぞれ、電力解析 C[1] で推定した $RK^{35(C[1])}$ と、電力解析 C[2] で推定した $RK^{35(C[2])}$ の値が一致するかどうかを比較する。そして、一致する場合、 $RK^{35(C[1])}$ (または $RK^{35(C[2])}$) を正解鍵と判定す

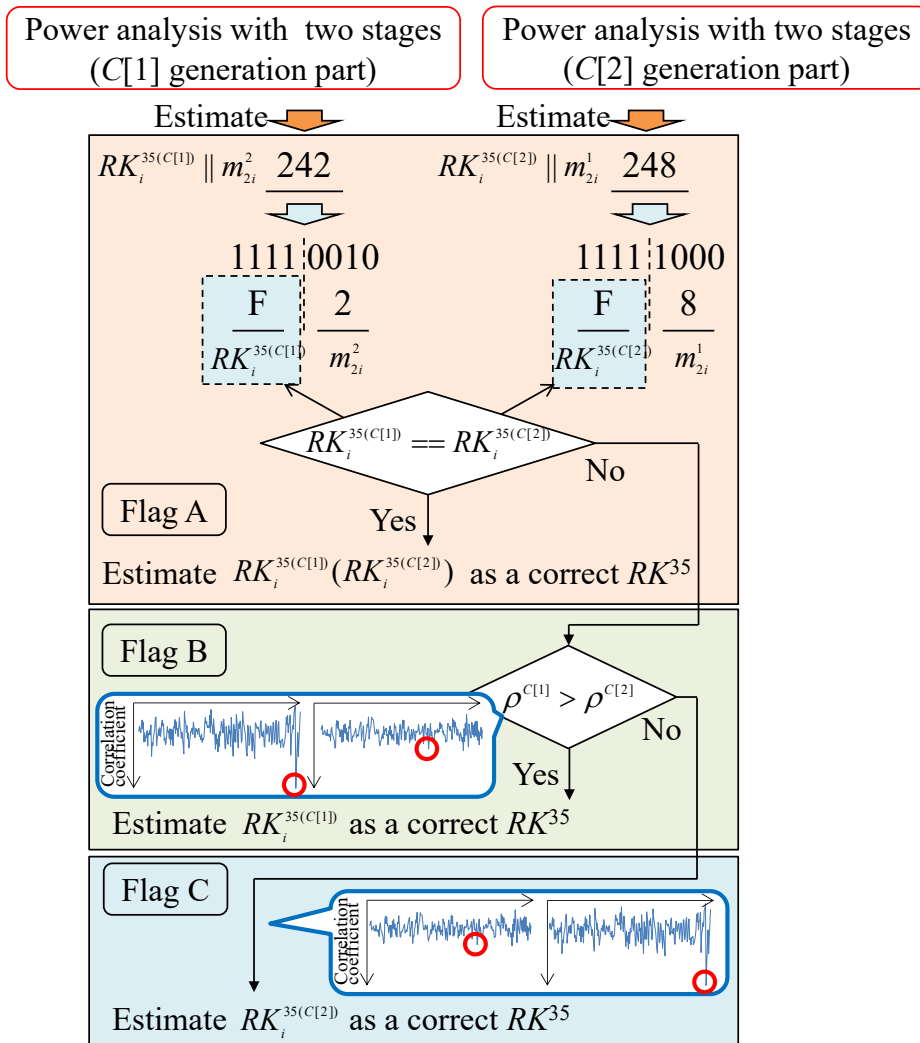


図 8.5 判定処理 I

Fig. 8.5 Judgement processing I

る（これを Flag A とする）。一方で、一致しない場合、図 8.5 に示すように各電力解析（電力解析 C[1]と電力解析 C[2]）で正解値として算出したピアソンの相関係数の値を比較する。そして、相関係数が大きい方を正解（ $RK_i^{35(C[1])} \parallel m_{2i}^2$ か $RK_i^{35(C[2])} \parallel m_{2i}^1$ ）とし、正解鍵の判定を行う（電力解析 C[1]の相関係数 $\rho^{C[1]}$ が大きい場合を Flag B、電力解析 C[2]の相関係数 $\rho^{C[2]}$ が大きい場合を Flag C とする）。このように、提案する電力解析手法では複数の解析結果の多くの情報を利用することで、解析効率を向上させる。

判定処理 II について図 8.6 と図 8.7 を用いて説明する。ここでは、 RK_0^{36} を判定する場合を例に説明する。このとき、1 段階目の電力解析の結果（ $m_0 \oplus RK_0^{36} \parallel m_1$ ）からラウンド鍵

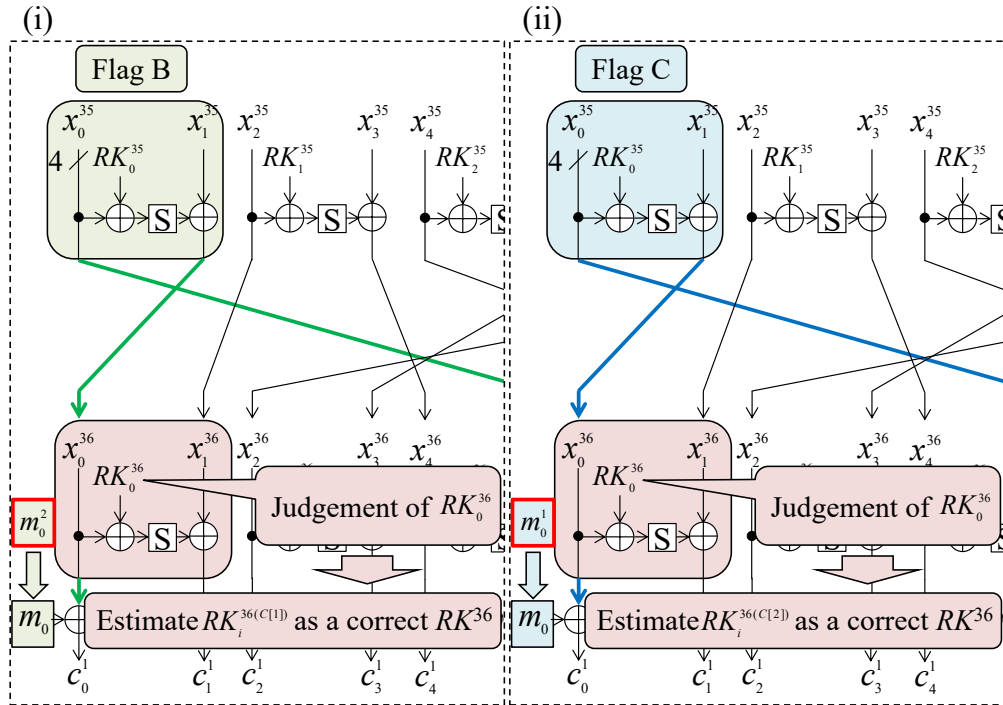


図 8.6 判定処理 II (Flag B または Flag C の場合)

Fig. 8.6 Judgement processing II in case of Flag B or C

RK_0^{36} を導出するためには、正しいメッセージ m_0 を使用する必要がある。そこで、判定処理 II では、判定処理 I で正しい鍵 RK_0^{35} を判定した電力解析（電力解析 C[1] または電力解析 C[2]）の結果を利用する。具体的には、図 8.6 の (i) に示すように、判定処理 I で Flag B の場合、電力解析 C[1] で推定したラウンド鍵 $RK^{36(C[1])}$ を正解鍵と判定する。また、図 8.6 の (ii) に示すように、Flag C の場合、電力解析 C[2] で推定したラウンド鍵 $RK^{36(C[2])}$ を正解鍵と判定する。このように、提案する電力解析手法では複数の電力解析（電力解析 C[1] と電力解析 C[2]）の結果から正解値を選択する。すなわち、複数の情報を利用することで、解析効率を向上させる。

一方で、図 8.7 の (i) に示すように、Flag A の場合、各電力解析で推定したラウンド鍵（ $RK^{36(C[1])}$ と $RK^{36(C[2])}$ ）の値が一致するかを比較する。一致する場合、 $RK^{36(C[1])}$ （または $RK^{36(C[2])}$ ）を正解鍵と判定する。また一致しない場合、暗号中間値 x_1^{36} に関連する部分（図 8.7 (ii)(iii) の例では、 RK_1^{35} を推定する電力解析）における判定処理 I の結果を利用する。これは、電力解析 C[1]（電力解析 C[2]）の 1 段階目の電力解析の結果が正解の場合、 x_1^{36} に関連部分の判定処理 I の結果は Flag B（Flag C）だと考えられるためである。なぜなら、1 段

階目の電力解析の結果が正解の場合、1 段階目の電力解析によって導出される x_1^{36} は正しい値だからである。したがって、図 8.7 の(ii)と(iii)に示すように、Flag B (Flag C) の場合、電力解析 C[1] (電力解析 C[2]) で推定したラウンド鍵 $RK^{36(C[1])}$ ($RK^{36(C[2])}$) を正解鍵と判定する。以上のように、提案する電力解析手法では解析効率を向上させるために、複数の電力解析の結果から正解値を選択する。

8.3 評価実験

8.3.1 実験環境

実験環境を図 8.8 と表 8.1 に示す。実験では、評価ボードに SASEBO-GII [7] を使用した。そして、SASEBO-GII 上に搭載されている FPGA Virtex-5 に TWINE-OTR を実装した。また、ナンス (Nonce) とメッセージを乱数で生成し、20,000 回の暗号処理を行い、20,000 個の消費電力波形をオシロスコープで取得した。実際に取得した消費電力波形の例を図 8.9 に示す。評価実験では、図 8.9 の C[1]生成部分と C[2]生成部分の 35R, 36R 目の処理に相当する部分の消費電力を解析に使用する。

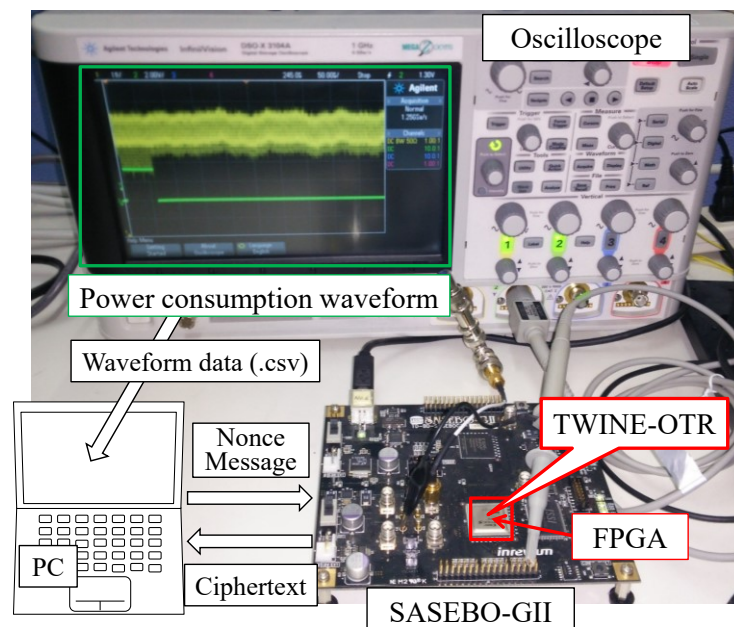


図 8.8 評価システム

Fig. 8.8 Evaluation system

表 8.1 実験環境

Table 8.1 Experimental condition

Encryption algorithm	TWINE-OTR
Key size [bit]	80
Message size [bit]	128
Nonce size [bit]	48
Evaluation board	SASEBO-GII [7]
FPGA	Virtex-5 XC5VLX30
Implementation tool	Xilinx ISE Design Suite 14.1
Oscilloscope	Agilent DSO 3104A
Sampling rate [Gsa/sec]	5
PC	HP ProBook 6570b
OS	Windows 7 Professional
Memory [GB]	8
CPU	Intel Core i7-3520M
Programming language	MATLAB 2013b

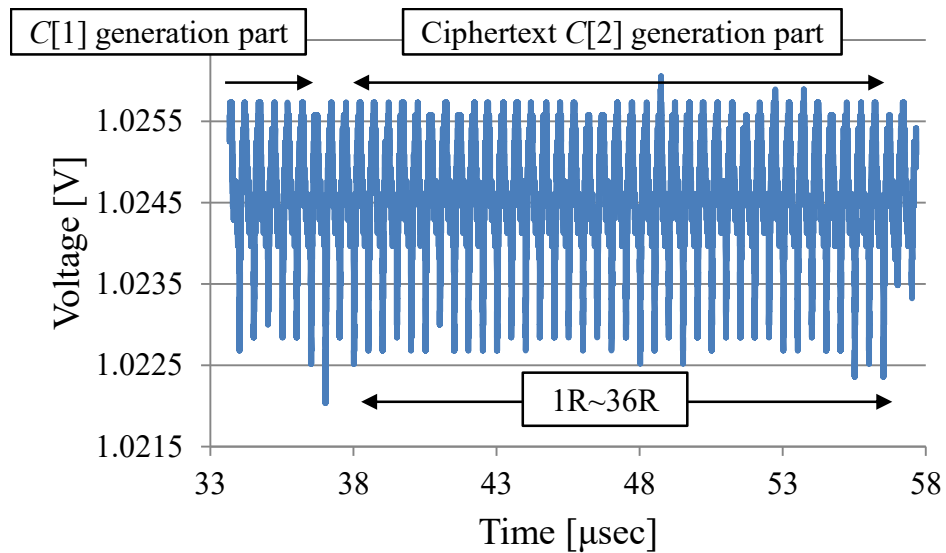


図 8.9 取得消費電力波形の例

Fig. 8.9 Example of the measured power consumption waveform

8.3.2 実験結果

実験結果を図 8.10 に示す。実験では、TWINE-OTR に対して本提案の電力解析手法を実施した。図 8.10 の縦軸は解析に成功したラウンド鍵 (RK^{35} と RK^{36}) の bit 数を、横軸は解析に使用した消費電力波形の数をそれぞれ示している。図 8.10 に示すように、本提案の電力解析によって、17,000 個の波形データでラウンド鍵 RK^{35} と RK^{36} を全て解析することに成功した。したがって、本提案の電力解析手法を用いることで TWINE-OTR の耐タンパ性を評価することができる。

8.3.3 複数の暗号文生成部分を利用した解析の有効性評価

複数の暗号文生成部分を利用した解析の有効性について検証した。この検証では、電力解析 C[1] (C[1]生成部分のみを対象) と電力解析 C[2] (C[2]生成部分のみを対象) を行い、それらの結果を比較した。比較結果を図 8.11 に示す。図 8.11 に示すように、電力解析 C[2] は、20,000 個の波形データを用いても全てのラウンド鍵を解析することができなかった。また、本提案の電力解析手法 (C[1] + C[2]) と電力解析 C[1] は、どちらも 17,000 個の波形データでラウンド鍵の解析に成功している。このとき、本提案の電力解析手法では、2,000 個少ない波形データで 60bit のラウンド鍵の解析に成功していることが確認できる。このように、本提案の電力解析手法では、波形データが少ない場合であっても、1 つの暗号文

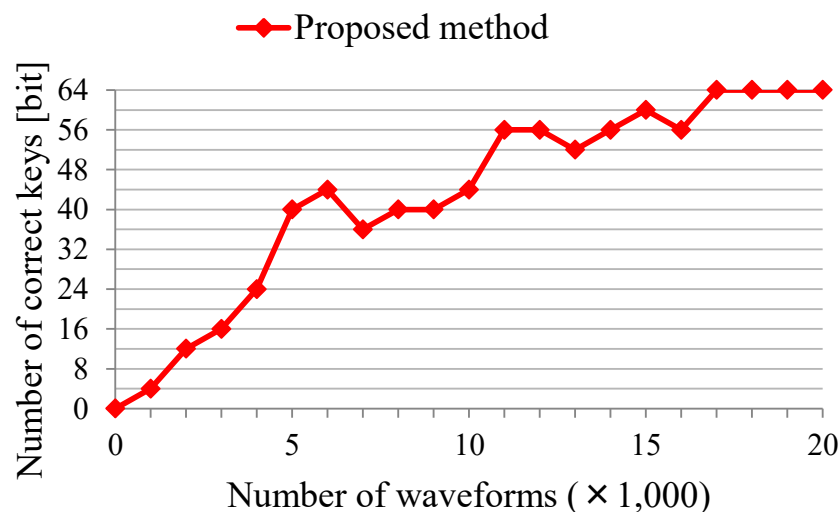


図 8.10 実験結果

Fig. 8.10 Experimental result

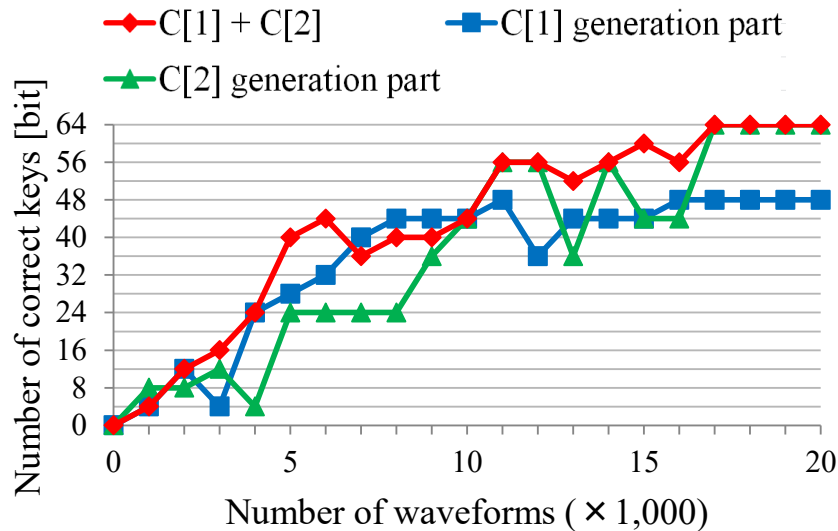


図 8.11 1つの暗号文生成部分のみを利用する解析との比較

Fig. 8.11 Comparison with the power analysis for only ciphertext generation part

生成部分のみを対象とした電力解析よりも多くのラウンド鍵を解析することができる。すなわち、より効率的に耐タンパ性を検証することができると考えられる。更に、本提案の電力解析手法では、電力解析 C[1]が解析に成功して、電力解析 C[2]が解析に失敗するなど、片方の電力解析（電力解析 C[1]か電力解析 C[2]）が失敗している場合でも、安定して解析を行うことができる。

8.3.4 相関係数の比較

各電力解析（電力解析 C[1]と電力解析 C[2]）での正解値における相関係数を比較した。実験結果を図 8.12 と図 8.13 に示す。図 8.12 は電力解析 C[1]の 1 段階目の電力解析における結果を、図 8.13 は電力解析 C[2]の 1 段階目の電力解析における結果を示している。また、図 8.12 と図 8.13 の縦軸は相関係数を、横軸は時間を示している。図 8.12 の結果においては、C[1]生成部分の 36R 目の処理に相当する箇所相関係数のピークが表れていることが確認できる。また、図 8.13 の結果においては、C[2]生成部分の 36R 目の処理に相当する箇所相関係数のピークが表れていることがわかる。

8.4 結言

本章では、TWINE-OTR に対する電力解析手法を提案し、FPGA を用いた評価実験によ

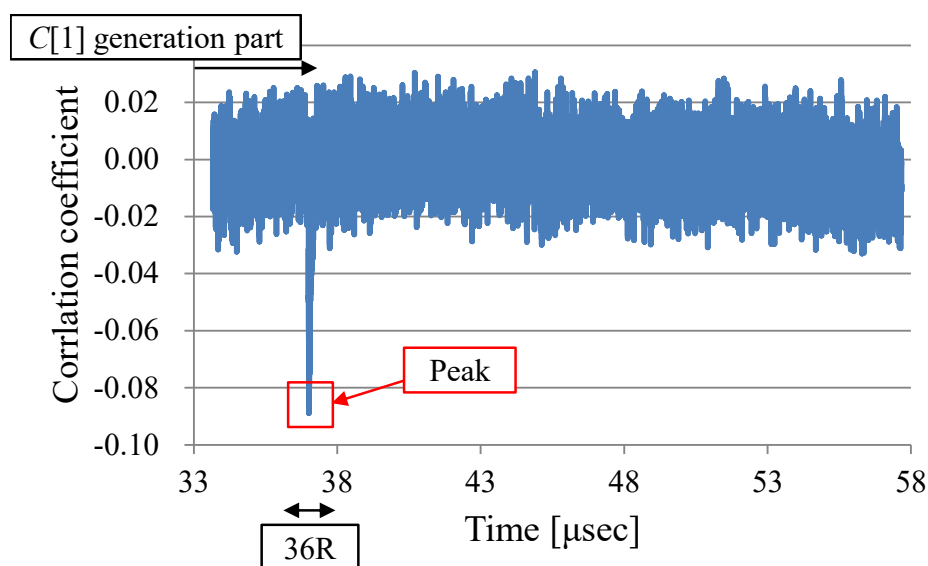


図 8.12 暗号文 C[1]生成部分を対象とした解析での相関係数

Fig. 8.12 Correlation coefficient for the analysis in the ciphertext C[1] generation part

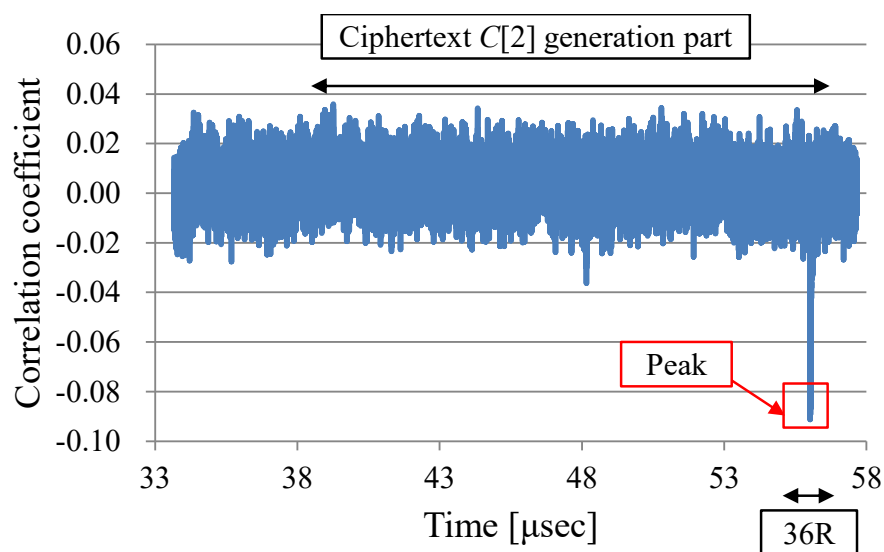


図 8.13 暗号文 C[2]生成部分を対象とした解析での相関係数

Fig. 8.13 Correlation coefficient for the analysis in the ciphertext C[2] generation part

ってその有効性を実証した。提案した電力解析手法では、TWINE-OTR の秘密鍵を解析するために、暗号文生成部分に対して2段階の階層的な解析を行う。また、提案した電力解析手法では、TWINE-OTR の構造に着目した複数の暗号文生成部分に対する電力解析を導入し、解析効率を向上させることに成功した。

以上より，提案した電力解析手法を利用することで，TWINE-OTR の耐タンパ性を評価できることを示した。

参考文献

- [1] 野崎佑典，吉川雅弥，“軽量認証暗号 TWINE-OTR に対する 2 段階電力解析とその評価，” 電子情報通信学会論文誌 D，vol. J101-D，no. 3，pp. 512–521，Mar. 2018.
- [2] K. Minematsu，“AES-OTR v3.1,” Sep. 2016.
<http://competitions.cr.yp.to/round3/aesotrv31.pdf>
- [3] “NEC プラットフォームズ セキュアセンサ BOX,”
<http://www.necplatforms.co.jp/product/secsensor/spec.html>
- [4] M. Dworki，“Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC,” National Institute of Standards and Technology Internal Special Publication 800-38D, Nov. 2007.
- [5] S. Belaïd, P. -A. Fouque, and B. Gérard，“Side-Channel Analysis of Multiplications in $GF(2^{128})$ Application to AES-GCM,” Proc. ASIACRYPT 2014, LNCS vol. 8874, pp. 306–325, Springer, Dec. 2014.
- [6] E. Brier, C. Clavier, and F. Olivier，“Correlation Power Analysis with a Leakage Model,” Proc. 6th Int. Workshop on Cryptographic Hardware and Embedded Systems (CHES 2004), LNCS vol. 3156, pp. 16–29, Springer-Verlag, Aug. 2004.
- [7] “The National Institute of Advanced Industrial Science and Technology Research Institute for Secure Systems: Evaluation Environment for Side-channel Attacks,”
<http://www.risec.aist.go.jp/project/sasebo>

第 9 章

TWINE に対するフォールト解析手法

9.1 緒言

本章では、TWINE の耐タンパ性評価に必要なフォールト解析手法を提案する [1][2]。関連する研究として、これまでにソフトウェア実装した TWINE [3]を対象としたラウンド加算 DFA が提案されている [4]。ラウンド加算 DFA は、汎用マイコンのソフトウェア処理でのラウンドのインクリメント命令をバイパスさせることで、ラウンド処理を繰り返し実行させる。そして、ラウンド加算によって出力される暗号文をフォールト入り暗号文として利用する。解析では、取得したフォールト入り暗号文と通常の暗号文を比較することで、ラウンド鍵の解析を行っている。一方で、これまでに FPGA や ASIC などのハードウェア実装した TWINE を対象としたフォールト解析の研究は見当たらない。

本章で提案するフォールト解析は、ハードウェア実装した TWINE を対象とし、フォールト混入手法にはクロックグリッチを想定する。そして、クロックグリッチによって混入するフォールトでも解析可能な統計的な解析手法を導入することで鍵の推定を行う。

9.2 提案するフォールト解析手法

提案するフォールト解析手法では、最も容易に実現可能なクロックグリッチ [5][6]によるフォールトを利用する。この解析では、TWINE の信号伝搬時間の偏りを利用する。信号伝搬時間の偏りによって、クロックグリッチによって混入するフォールトの数に偏りが生じる。提案するフォールト解析では、このフォールト混入数の偏りに着目し、統計的な処

本章は電子情報通信学会 Copyright © 2015 IEICE で発表された内容 [1][2]で構成しており、本章の図は電子情報通信学会の許可を得て再利用しています。

理を行うことで鍵を推定する。

まず 9.2.1 項では、提案するフォールト解析で利用する TWINE の信号伝搬時間の偏りについて説明する。次に 9.2.2 項では、提案するフォールト解析で使用するフォールトモデルについて説明する。最後に、9.2.3 項で具体的な解析手法について説明する。

9.2.1 TWINE の信号伝搬時間の偏り

まず、TWINE の信号伝搬時間の偏りについて説明する。TWINE は図 9.1 に示すように演算を行う部分と、演算を行わない部分が存在する。具体的には、S-BOX による処理と排他的論理和演算を行う部分と、拡散層によるビット移動のみを行う部分に分かれる。 r ラウンド目の処理では、奇数番目の暗号中間値 x_{2j+1}^r は演算を行うため、信号伝搬時間が大きくなる。一方で、偶数番目の暗号中間値 x_{2j}^r は演算を行わないため、信号伝搬時間は小さくなる。すなわち、演算量の違いから信号伝搬時間に偏りが生じる。

また、 r ラウンド目計算終了後は拡散層によって、信号伝搬時間の大きい値は偶数番目に、信号伝搬時間の小さい値は奇数番目に移動する。したがって、 $r+1$ ラウンド目の奇数

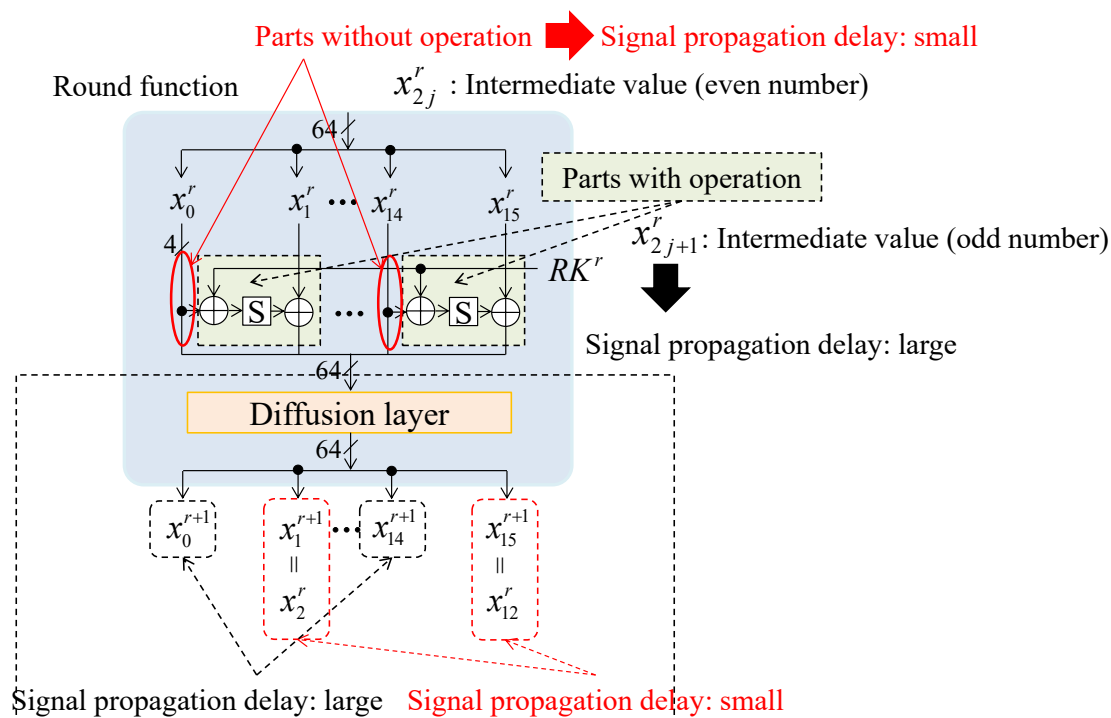


図 9.1 信号伝搬時間の偏り

Fig. 9.1 Deviation of signal propagation delay

番目の暗号中間値には、信号伝搬時間の小さい値が転置される。

9.2.2 フォールトモデル

提案するフォールト解析で用いるフォールトモデルについて、図9.2を用いて説明する。

図 9.2 に示すように、35 ラウンド目 (35R) に対してクロックグリッチを挿入する。3.4.2 項でも述べた通り、クロックグリッチによってフリップフロップでセットアップタイム制約違反を引き起こすことで、複数のフォールトを混入させることができる。一方で、クロックグリッチは暗号回路全体に供給されるため、フォールトが混入する位置や、個数などを制御することはできない。

しかし、TWINE は 9.2.1 項で説明した信号伝搬時間の偏りから、フォールト混入数に偏りが生じる。具体的には、信号伝搬時間が大きい箇所では、多くのフォールトが混入し、信号伝搬時間の小さい箇所では、フォールトが混入しにくい。すなわち、図 9.2 に示すように、暗号中間値 x_{2j+1}^{36*} はフォールト混入数が少なくなる。これは、信号伝搬時間が大きい場合は、フリップフロップの値が確定するタイミングが遅く、信号伝搬時間が小さい場合

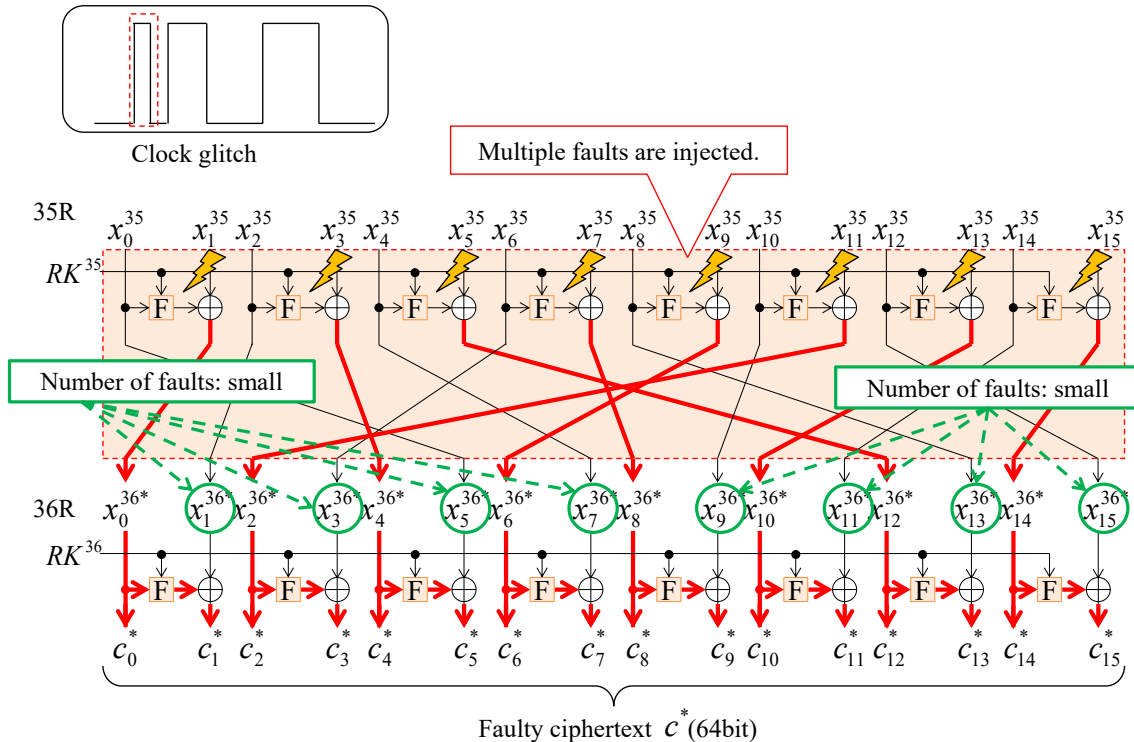


図 9.2 フォールトモデル

Fig. 9.2 Fault model

は、フリップフロップの値が確定するタイミングが早くなるためである。

また、混入したフォールトは伝搬し、最終的にフォールト入り暗号文 c^* を取得することができる。

9.2.3 統計的な解析手法

まず、提案するフォールト解析手法の概要について図 9.3 を用いて説明する。提案するフォールト解析では、通常の暗号文とフォールト入り暗号文のペアを複数組用意する。そして、暗号文ペアから、35R 計算終了後の暗号中間値 x^{36} を算出する。この暗号中間値は、TWINE の信号伝搬時間の偏りによって、フォールト混入数が少ない部分 x_{2j}^{36} を対象として 4bit ごとに算出する。

この暗号中間値は図 9.3 の(i)に示すように、既知の暗号文と部分鍵の予測値を用いた逆算処理によって計算できる。このとき、通常の暗号文 c から求めた暗号中間値を通常の暗号中間値 x^{36} 、フォールト入り暗号文 c^* から求めた暗号中間値をフォールト入り暗号中間値

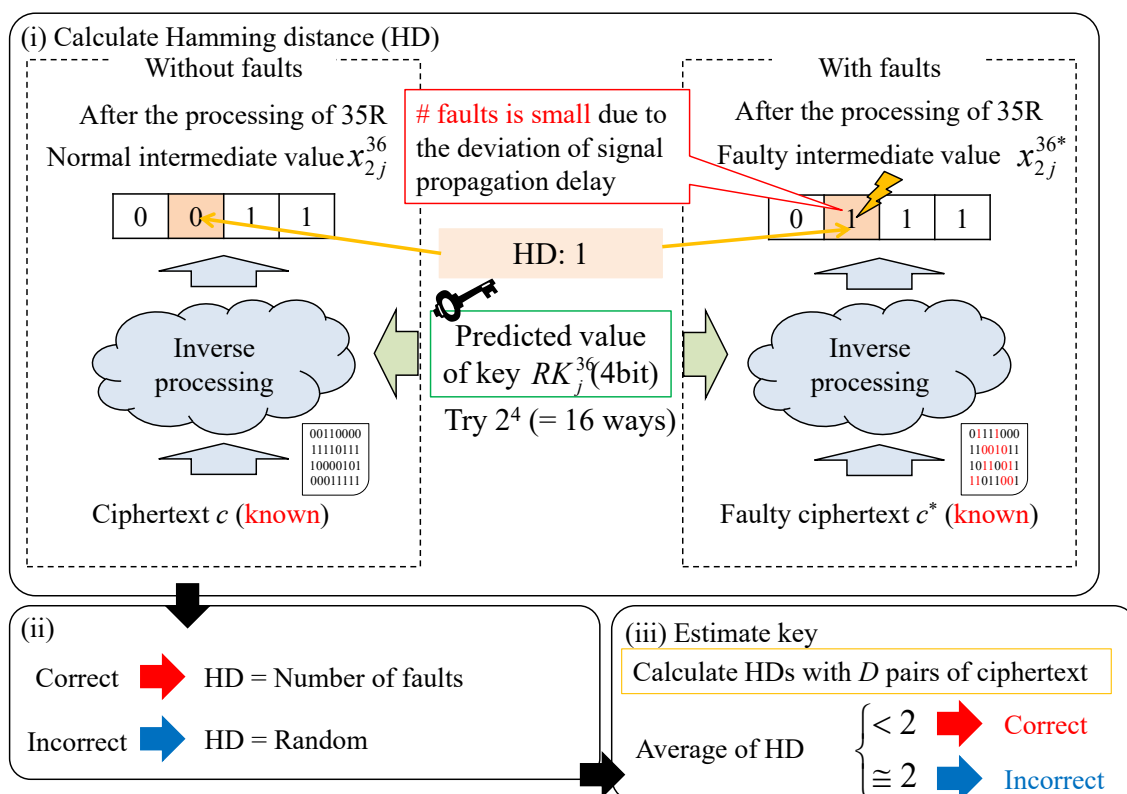


図 9.3 提案するフォールト解析手法の概要

Fig. 9.3 Outline of the proposed fault analysis method

x^{36*} と呼ぶ。そして、各暗号中間値（通常の暗号中間値とフォールト入り暗号中間値）でハミング距離を計算する。このとき、図 9.3 の(ii)に示すように、部分鍵の予測値が正しい場合、計算したハミング距離はフォールト混入数と等しくなる。一方で、部分鍵の予測値が間違っている場合、計算したハミング距離はランダムな値となる。

提案するフォールト解析では、このハミング距離を十分に大きい D 組の暗号文ペアを用いてそれぞれ計算する。そして、部分鍵の予測値ごとに平均ハミング距離をそれぞれ算出する。このとき、部分鍵の予測値が正しい場合、平均ハミング距離は小さくなる。なぜなら、算出したフォールト入り暗号中間値は、TWINE の信号伝搬時間の偏りから、混入するフォールトの数が少なくなるためである。一方で、部分鍵の予測値が間違っている場合、平均ハミング距離は 4bit のデータのハミング距離の期待値である 2 に近い値となる。したがって、図 9.3 の(iii)に示すように、平均ハミング距離を最小とする部分鍵の予測値を正解鍵として推定する。

提案するフォールト解析手法の詳細について説明する。まず、通常の暗号文とフォールト入り暗号文では、式(A.1), (A.2)に基づき、既知の暗号文と部分鍵を利用して、式(9.1), (9.2)で表すことができる。

$$\begin{aligned} c_{2j} &= x_{2j+1}^{36} \\ c_{2j+1} &= S(x_{2j}^{36} \oplus RK_j^{36}) \oplus x_{2j+1}^{36} \end{aligned} \quad (9.1)$$

$$\begin{aligned} c_{2j}^* &= x_{2j+1}^{36*} \\ c_{2j+1}^* &= S(x_{2j}^{36*} \oplus RK_j^{36}) \oplus x_{2j+1}^{36*} \end{aligned} \quad (9.2)$$

ここで、式(9.1), (9.2)から $c_{2j} = x_{2j+1}^{36}$, $c_{2j}^* = x_{2j+1}^{36*}$ である。したがって、各暗号中間値 x_{2j+1}^{36} , x_{2j+1}^{36*} は式(9.3), (9.4)で計算できる。

$$x_{2j+1}^{36} = S(c_{2j} \oplus RK_j^{36}) \oplus c_{2j+1} \quad (9.3)$$

$$x_{2j+1}^{36*} = S(c_{2j}^* \oplus RK_j^{36}) \oplus c_{2j+1}^* \quad (9.4)$$

このとき、式(9.3), (9.4)の部分鍵 RK_j^{36} は未知の値である。そこで、部分鍵の予測値として pK ($0 \leq pK \leq 15$) を与える。この計算結果を pX , pX^* とすると、式(9.5), (9.6)で計算す

ることができる。

$$pX = S(c_{2j} \oplus pK) \oplus c_{2j+1} \quad (9.5)$$

$$pX^* = S(c_{2j}^* \oplus pK) \oplus c_{2j+1}^* \quad (9.6)$$

また、式(9.3)から式(9.6)では x_{2j+1}^{36} と pX , x_{2j+1}^{36*} と pX^* に式(9.7)に示す関係が成り立つ。

$$pX \begin{cases} = x_{2j+1}^{36} & (pK = RK_j^{36}) \\ \neq x_{2j+1}^{36} & (pK \neq RK_j^{36}) \end{cases}, \quad pX^* \begin{cases} = x_{2j+1}^{36*} & (pK = RK_j^{36}) \\ \neq x_{2j+1}^{36*} & (pK \neq RK_j^{36}) \end{cases} \quad (9.7)$$

ここで、2つの値 a, b のハミング距離を求める関数を $HD(a, b)$ とすると、 x_{2j+1}^{36} と x_{2j+1}^{36*} のハミング距離 h と、 pX と pX^* のハミング距離 hpX は式(9.8)と式(9.9)で計算できる。

$$h = HD(x_{2j+1}^{36}, x_{2j+1}^{36*}) \quad (9.8)$$

$$hpX = HD(pX, pX^*) \quad (9.9)$$

加えて、それぞれ計算したハミング距離において式(9.10)が成り立つ。

$$h \begin{cases} = hpX & (pK = RK_j^{36}) \\ \neq hpX & (pK \neq RK_j^{36}) \end{cases} \quad (9.10)$$

式(9.10)において、 $pK \neq RK_j^{36}$ のとき、 pX と pX^* には相関はなく、ハミング距離の期待値は2 (4bit の中央値) である。一方で、 $pK = RK_j^{36}$ のとき、 pX と pX^* には、TWINE の信号伝搬時間の偏りからフォールトが混入しにくいいため、ハミング距離の期待値は2 よりも小さくなる。

したがって、十分に大きい N 組の暗号文ペアを用いて、式(9.8), (9.9)から、ハミング距離 $hpX_1, \dots, hpX_N$ を求めると、平均ハミング距離 $\overline{HD}$ は式(9.11)で計算できる。

$$\overline{HD} = \frac{1}{N} \sum_{n=1}^N hpX_n \begin{cases} < 2 & (pK = RK_j^{36}) \\ \cong 2 & (pK \neq RK_j^{36}) \end{cases} \quad (9.11)$$

以上より、平均ハミング距離 $\overline{HD}$ を最小とする部分鍵の予測値 pK を正解鍵として推定する。

9.3 評価実験

9.3.1 実験環境

実験環境を図 9.4 と表 9.1 に示す。評価ボードには SASEBO-GII [7] を使用し, SASEBO-GII 上の FPGA に鍵長が 80bit の TWINE を実装した。TWINE の実装に関して, S-BOX の実装には合成体方式 [3][8][9] を使用し, 部分鍵の生成は事前計算方式 [10] とした。

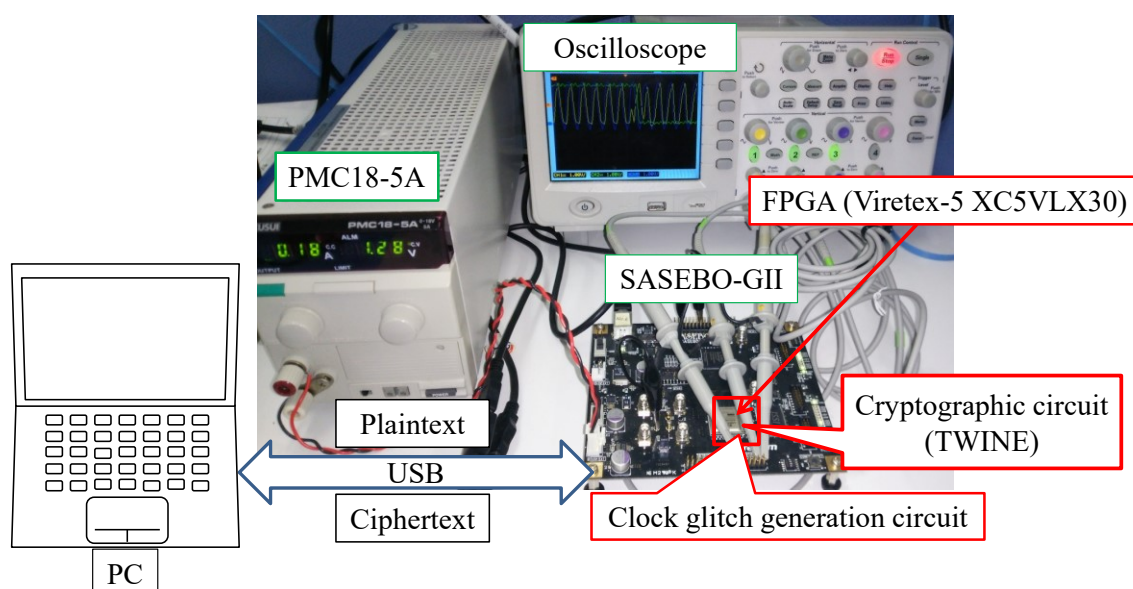


図 9.4 評価システム

Fig. 9.4 Evaluation system

表 9.1 実験環境

Table. 9.1 Experimental condition

Encryption algorithm	TWINE
Key size [bit]	80
S-BOX	Composite field
Evaluation board	SASEBO-GII [7]
FPGA	Viretex-5 XC5VLX30
Implementation tool	ISE Design Suite 14.1
Power supply	KIKUSUI PMC18-5A

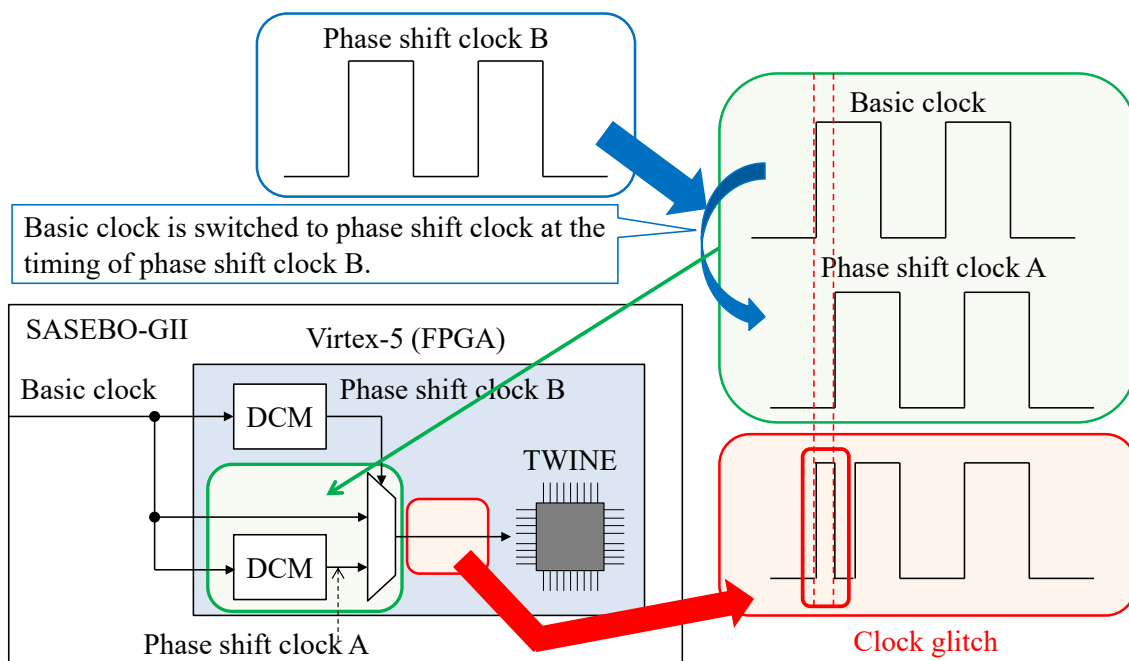


図 9.5 クロックグリッチの生成方法

Fig. 9.5 Clock glitch generation method

また、フォールトを混入させるためのクロックグリッチの生成方法を図 9.5 に示す。図 9.5 に示すように、クロックグリッチは、2 種類の位相の異なるクロックパルスを任意のタイミングで切り替えることで生成した。具体的には、まず、基本となるクロックパルス（基本クロック）と位相の異なる 2 種類のクロックパルス（位相シフトクロック A、位相シフトクロック B）を用意する。そして、基本クロックを任意のタイミングで位相シフトクロック A に切り替える。この任意のタイミングは位相シフトクロック B によって決定する。以上のように、クロックパルスを切り替えることでクロックグリッチを生成することが可能である。

ここで、位相シフトクロックは、FPGA Virtex-5 で用意されている DCM（Digital Clock Manager）を用いて生成した。DCM では、任意の位相パラメータを設定することで位相を調整することができる。

9.3.2 実験結果

評価実験では、まず、36R の部分鍵を対象として提案するフォールト解析手法を適用した。この実験では、35R に対して、クロックグリッチを挿入し、100 組の暗号文のペアを

取得した。実験結果を図 9.6 に示す。図の横軸は解析に使用した暗号文ペアの数を、縦軸は解析に成功した部分鍵の数を示している。図 9.6 に示すように、100 組の暗号文ペアを用いることで、全ての部分鍵を解析することに成功していることが分かる。したがって、提案するフォールト解析手法は有効であると考えられる。

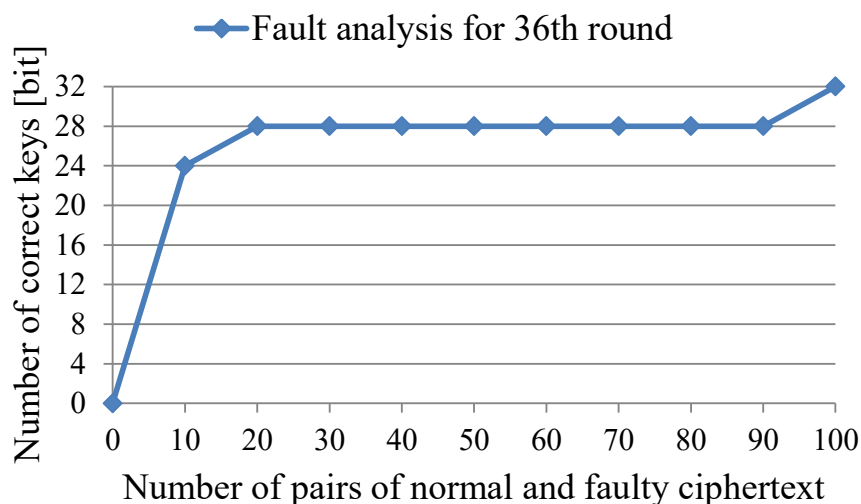


図 9.6 実験結果 (36 ラウンド目)

Fig. 9.6 Experimental result for 36th round

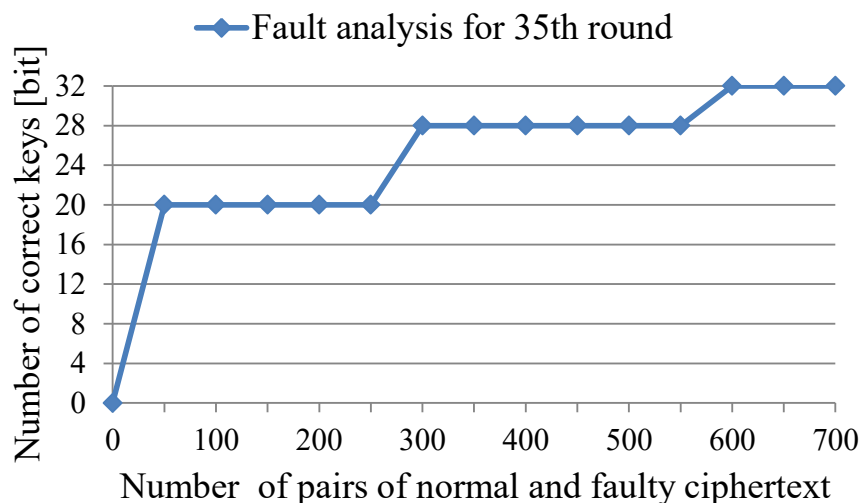


図 9.7 実験結果 (35 ラウンド目)

Fig. 9.7 Experimental result for 35th round

ここで、本章で対象としている TWINE は、第 4 章でも述べた通り鍵長が 80bit であるため、36R の部分鍵 32bit だけでは、全ての秘密鍵を解析することはできない。そこで、35R の部分鍵を解析するための実験を行った。この実験では、解析に成功した 36R の部分鍵を用いて、35R に対して提案するフォールト解析手法を適用した。このとき、フォールト入り暗号文は、34R に対してクロックグリッチを挿入することで取得した。実験結果を図 9.7 に示す。図 9.7 の結果においては、600 組の暗号文のペアを用いることで全ての部分鍵の解析に成功しており、提案するフォールト解析手法が有効であることが分かる。

以上より、提案するフォールト解析手法を用いて 35R と 36R の部分鍵をそれぞれ解析し、合計で 64bit の部分鍵を解析することに成功した。残りの 16bit の部分鍵については、34R に対して、同様に提案するフォールト解析手法を適用するか、総当たりの計算によって求めることができる。総当たりの計算では、 $2^{16} = 65536$ 通りの鍵候補を試すだけであり、現実的な時間で解析することが可能であると考えられる。最後に、鍵スケジュール部の逆スケジュール処理を行うことで、全ての秘密鍵を解析することができる。したがって、提案するフォールト解析手法によって、TWINE の耐タンパ性を評価することができる。

9.3.3 ハミング距離の比較

各鍵候補における平均ハミング距離を比較した。比較結果を図 9.8 に示す。この実験は、36R を対象に暗号文ペアを 100 組用いて行った。図の横軸は各鍵候補を、縦軸は平均ハミング距離を示している。また、図の赤色は正解鍵を、青色は不正解鍵の結果を示している。図 9.8 の結果においては、正解鍵で平均ハミング距離が最小となっていることが確認できる。これは、TWINE の信号伝搬時間の偏りによって、混入するフォールトの数が少ないためであると考えられる。一方で、不正解鍵では、平均ハミング距離は、4bit のハミング距離の期待値である 2 に近付いていることが分かる。

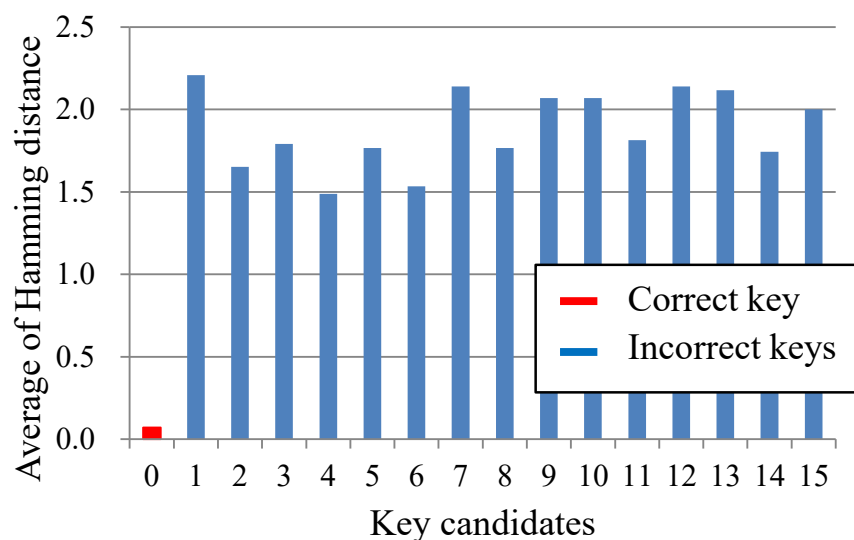


図 9.8 平均ハミング距離の比較

Fig. 9.8 Comparison of the average of Hamming distance

9.4 結言

本章では、TWINE に対するフォールト解析手法を提案し、FPGA を用いた評価実験によってその有効性を実証した。提案したフォールト解析手法では、最も容易に実現可能なフォールト混入手法の 1 つである、クロックグリッチを用いた解析を行った。解析では、TWINE の信号伝搬時間の偏りによるフォールト混入数の偏りに着目し、統計的な処理を行うことで鍵の推定を成功させ、提案したフォールト解析手法を用いることで、TWINE の耐タンパ性を評価できることを示した。

参考文献

- [1] 野崎佑典, 吉川雅弥, “軽量暗号 TWINE に対するフォールト解析とその実装評価,” 電子情報通信学会技術研究報告, IEICE-ITS, vol. 115, no. 187, pp. 57–62, Aug. 2015.
- [2] 野崎佑典, 吉川雅弥, “TWINE に対するエラー値を考慮したフォールト攻撃,” 2015 年暗号と情報セキュリティシンポジウム講演論文集, 3A1-5, pp. 1–6, Jan. 2015.

- [3] T. Suzaki, K. Minematsu, S. Morioka, and E. Kobayashi, “TWINE: A Lightweight, Versatile Blockcipher,” Proc. ECRYPT Workshop on Lightweight Cryptography (LC11), pp. 146–149, Nov. 2011.
- [4] H. Yoshikawa, M. Kaminaga, A. Shikoda, and T. Suzuki, “Round addition DFA on 80-bit Piccolo and TWINE,” IEICE Trans. Information and Systems, vol. E96-D, no. 9, pp. 2031–2035, Sep. 2013.
- [5] 福永利徳, 高橋順子, “暗号 LSI に対するフォールト解析攻撃実験,” 2009 年暗号と情報セキュリティシンポジウム講演論文集, 2A3-5, pp. 1–6, Jan. 2009.
- [6] S. Endo, T. Sugawara, N. Homma, T. Aoki and A. Satoh, “A Configurable On-Chip Glitchy-Clock Generator for Fault Injection Experiments,” IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences, vol. E95-A, no. 1, pp. 263–266, Jan. 2012.
- [7] “The National Institute of Advanced Industrial Science and Technology Research Institute for Secure Systems: Evaluation Environment for Side-channel Attacks,”
<http://www.risec.aist.go.jp/project/sasebo>
- [8] A. Satoh, S. Morioka, K. Takano, and S. Munetoh, “A Compact Rijndael Hardware Architecture with S-Box Optimization,” Proc. ASIACRYPT 2001, LNCS vol. 2248, pp. 239–254, Springer-Verlag, Dec. 2001.
- [9] 森岡澄夫, 佐藤 証, 高野光司, “ $GF(((2^2)^2)^2)$ 上の演算を用いた AES の S-Box 構成法,” 情報処理学会第 63 回全国大会講演論文集, 3G-04, Sep. 2001.
- [10] (独) 産業技術総合研究所 情報セキュリティ研究センター, “標準暗号 LSI 仕様書～サイドチャネル攻撃対策版～,” Aug. 2009.
http://www.rcis.aist.go.jp/files/special/SASEBO/CryptoLSI-en/CryptoLSI2_Spec_Ver1.0_Japanese.pdf

第 10 章

SIMECK に対するフォールト解析手法

10.1 緒言

本章では，SIMECK の耐タンパ性評価に必要なフォールト解析手法を提案する [1]。関連する研究として，これまでに SIMECK [2]に対する DFA [3]が提案されている。文献 [3]では，SIMECK に対して bit ランダムフォールトモデルと 1Byte フォールトモデルを導入して DFA を適用している。一方で，文献 [3]ではランダムな位置に複数個のフォールトが混入する場合は解析を行うことは難しい。

他にも近年では，ソフトウェア実装された SIMECK に対するラウンド加算 DFA が提案されている [4]。文献 [4]では，汎用マイコンにおけるソフトウェア処理のラウンドのインクリメント命令をバイパスさせることで，ラウンド処理を繰り返し実行させる。そして，ラウンド加算によって出力される暗号文をフォールト入り暗号文として利用した解析によって，ラウンド鍵を推定している。

本章で提案するフォールト解析手法は，ハードウェア実装した SIMECK を対象とし，ランダムな位置に複数個のフォールトが混入した際にも解析が可能な統計的な解析手法を導入する。

10.2 提案するフォールト解析手法

10.2.1 フォールトモデル

提案するフォールト解析手法で用いるフォールトモデルについて図 10.1 を用いて説明す

本章は電子情報通信学会 Copyright © 2016 IEICE で発表された内容 [1]で構成しており，本章の図は電子情報通信学会の許可を得て再利用しています。

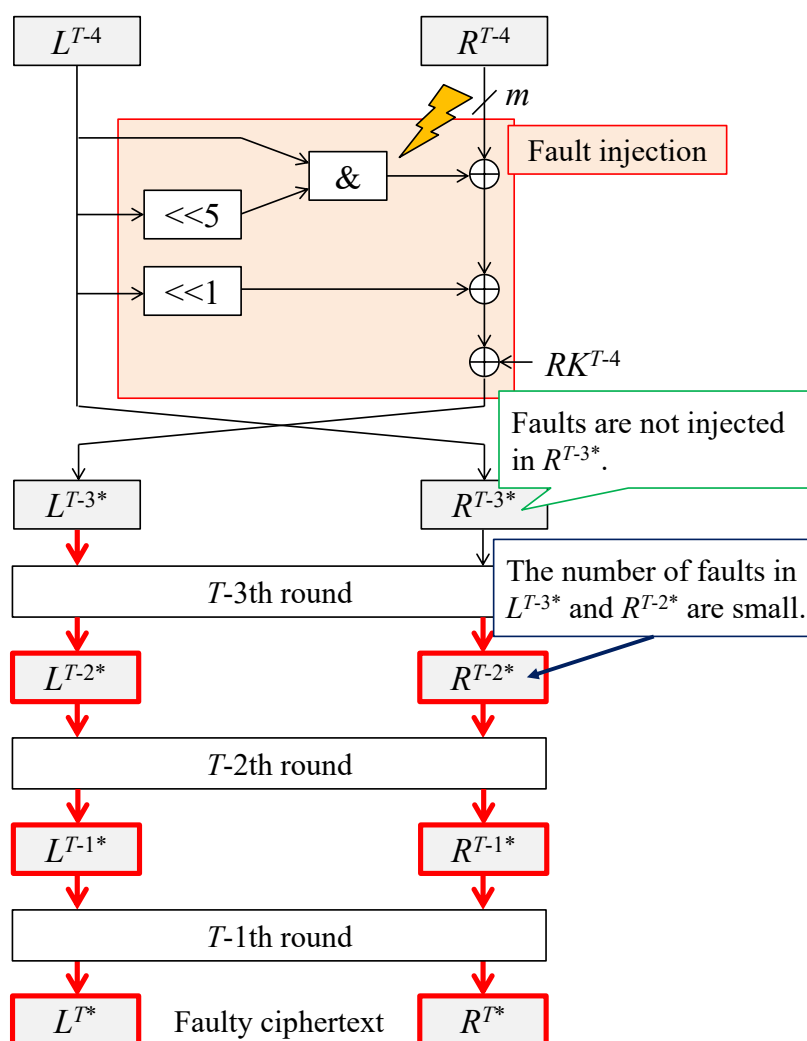


図 10.1 フォールトモデル

Fig. 10.1 Fault model

る。提案するフォールト解析手法では、図 10.1 に示すように SIMECK の $T-4$ ラウンド目に対して、クロックグリッチ [5][6] を挿入する。3.4.2 項でも述べた通り、クロックグリッチによってフリップフロップのセットアップタイム制約違反を引き起こすことでフォールトを混入させることができる。このとき、クロックグリッチは回路全体に供給され、ランダムな位置で複数のフォールトが混入する。

具体的には、図 10.1 に示すように L^{T-3*} ($=R^{T-2*}$) で複数のフォールトが混入する。また、 R^{T-3*} に供給される値は式(A.11)から分かるように、ほとんど演算を行っていないため、セットアップタイム制約違反によるフォールトは混入しない。このとき、全てのフリップフロップでセットアップタイム制約違反が発生した場合、フリップフロップの値の変化するパ

ターンは $0 \rightarrow 0$, $0 \rightarrow 1$, $1 \rightarrow 0$, $1 \rightarrow 1$ の 4 種類であり、値が反転（フォールトが混入）する確率は $1/2$ である。しかし、実際には全てのフリップフロップでセットアップタイム制約違反は発生しないため、各ビットでフォールトが発生する確率は $1/2$ よりも小さくなる。したがって、 $R^{T-2*} (=L^{T-3*})$ のフォールト混入数は少なくなる。

そして、図 10.1 に示すようにフォールトは伝搬し、最終的にフォールト入り暗号文 L^{T*} , R^{T*} を取得する。

10.2.2 解析手法の概要

提案するフォールト解析手法の概要を図 2.2 に示す。提案するフォールト解析手法は、10.2.3 項で説明する 1 段階目のフォールト解析と、10.2.3 項で説明する 2 段階目のフォールト解析で構成する。1 段階目のフォールト解析では、10.2.1 項のフォールトモデルに基づき、 R^{T-2*} はフォールト混入数が少ないという性質を利用した解析を行い、ラウンド鍵 RK^{T-1} を推定する。2 段階目のフォールト解析では、 R^{T-3} ではフォールトは混入しないという性質を利用した解析を行い、ラウンド鍵 RK^{T-2} を推定する。

10.2.3 1 段階目のフォールト解析手法

1 段階目のフォールト解析の概要を図 10.3 に示す。1 段階目のフォールト解析は、 R^{T-2*} はフォールト混入数が少ないという性質を利用してラウンド鍵 RK^{T-1} を推定する。まず、通常の暗号文 L^T , R^T とフォールト入り暗号文 L^{T*} , R^{T*} の組を複数個用意する。そして、図

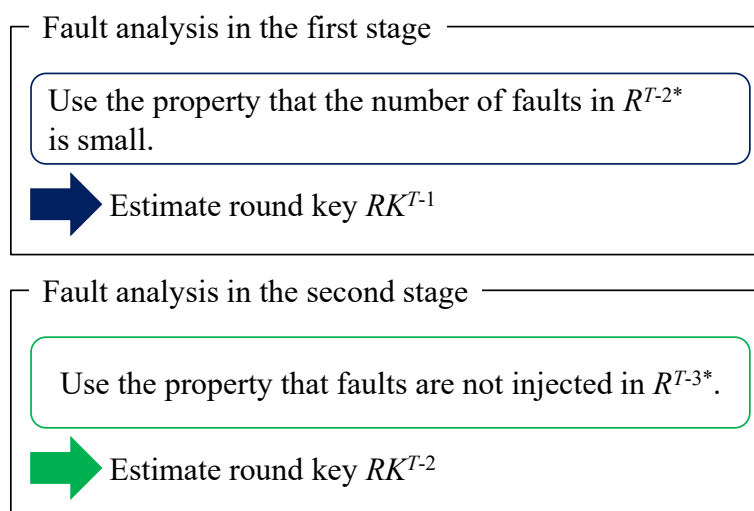


図 10.2 提案するフォールト解析手法の概要

Fig. 10.2 Outline of the proposed fault analysis method

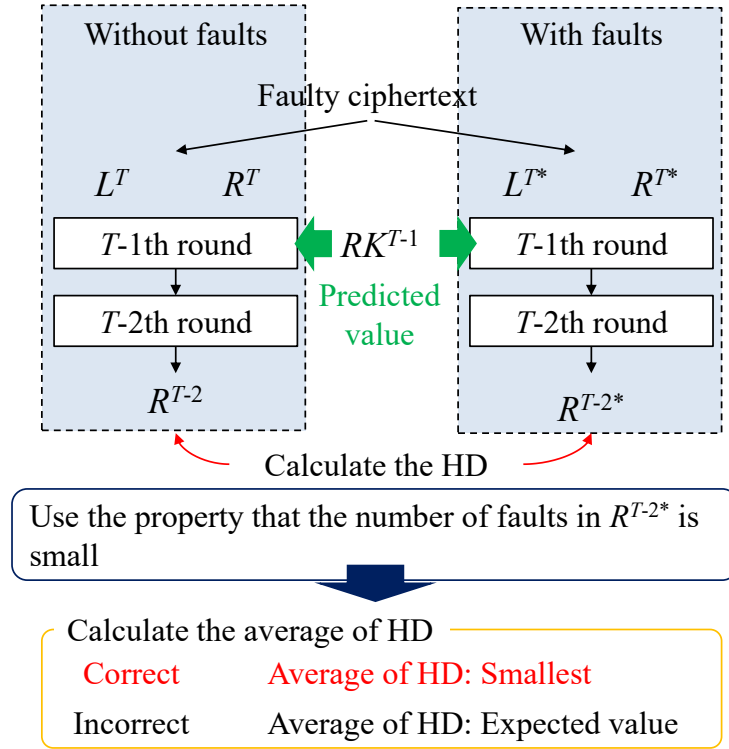


図 10.3 1段階目のフォールト解析

Fig. 10.3 Fault analysis in the first stage

10.3 に示すように各暗号文の組から T -3 ラウンド目計算終了後の暗号中間値 R^{T-2} と R^{T-2*} を計算する。

ここで、 R^{T-2} は通常の暗号文から計算した暗号中間値（通常の暗号中間値）、 R^{T-2*} はフォールト入り暗号文から計算した暗号中間値（フォールト入り暗号中間値）である。この各暗号中間値は、既知の暗号文とラウンド鍵 RK^{T-1} の予測値を用いた計算によって求めることができる。そして、各暗号中間値 R^{T-2} と R^{T-2*} の組でそれぞれハミング距離を計算し、ハミング距離の平均値を算出する。

このとき、ラウンド鍵 RK^{T-1} の予測値が正しい場合、求めたハミング距離とフォールト混入数は等しくなる。また、 R^{T-2*} はフォールト混入数が少ないため、ハミング距離の平均値は小さくなる。一方で、ラウンド鍵 RK^{T-1} の予測値が間違っている場合、ハミング距離の平均値は予測値が正しい場合と比較して大きくなる。したがって、ハミング距離の平均値が最小となるラウンド鍵 RK^{T-1} の予測値を正解鍵として推定する。

具体的な計算方法について説明する。各暗号中間値 R^{T-2} と R^{T-2*} は式(A.10), (A.11)から、式(10.1), (10.2)によって計算できる。

$$R^{T-2} = R^T \oplus f(R^{T-1}) \oplus RK^{T-2} \quad (10.1)$$

$$R^{T-2*} = R^{T*} \oplus f(R^{T-1*}) \oplus RK^{T-2} \quad (10.2)$$

ここで、式(10.1), (10.2)における R^{T-1} と R^{T-1*} は、式(10.3), (10.4)を用いて計算できる。

$$R^{T-1} = L^T \oplus f(R^T) \oplus RK^{T-1} \quad (10.3)$$

$$R^{T-1*} = L^{T*} \oplus f(R^{T*}) \oplus RK^{T-1} \quad (10.4)$$

したがって、 R^{T-1} と R^{T-1*} は式(10.5), (10.6)で計算できる。

$$R^{T-2} = R^T \oplus f(L^T \oplus f(R^T) \oplus RK^{T-1}) \oplus RK^{T-2} \quad (10.5)$$

$$R^{T-2*} = R^{T*} \oplus f(L^{T*} \oplus f(R^{T*}) \oplus RK^{T-1}) \oplus RK^{T-2} \quad (10.6)$$

ここで、式(10.5), (10.6)のラウンド鍵 RK^{T-1} は未知の値である。そこで、ラウンド鍵 RK^{T-1} の予測値として pRK^{T-1} を与える。この計算結果をそれぞれ pR^{T-2} , pR^{T-2*} とすると、式(10.7), (10.8)で計算できる。

$$pR^{T-2} = R^T \oplus f(L^T \oplus f(R^T) \oplus pRK^{T-1}) \oplus RK^{T-2} \quad (10.7)$$

$$pR^{T-2*} = R^{T*} \oplus f(L^{T*} \oplus f(R^{T*}) \oplus pRK^{T-1}) \oplus RK^{T-2} \quad (10.8)$$

このとき、ラウンド鍵はブロック長によってそのビット長が異なり、ブロック長が 32bit, 48bit, 64bit の場合に合わせて、それぞれ 16bit, 24bit, 32bit と変化する。そのため、ラウンド鍵の予測値には各ブロック長 (32bit, 48bit, 64bit) に合わせて、それぞれ 2^{16} , 2^{24} , 2^{32} 通りの計算が必要となり、計算量が膨大になる。

そこで提案するフォールト解析手法では、SIMECK の演算がビット単位で行っていることを利用して計算量を削減する。具体的には、図 10.4 に示すように各暗号中間値をビット単位で導出する。図 10.4 に示す例は、ブロック長が 32bit, ラウンド鍵が 16bit の SIMECK を対象としたものである。この計算では、ラウンド鍵の予測値は 2bit ($2^2=4$ 通り) ずつの値を使用するため、計算量を大幅に削減することが可能である。

次に、式(10.5)から式(10.8)において、 R^{T-2} と pR^{T-2} , R^{T-2*} と pR^{T-2*} には式(10.9)に示す関係が成り立つ。

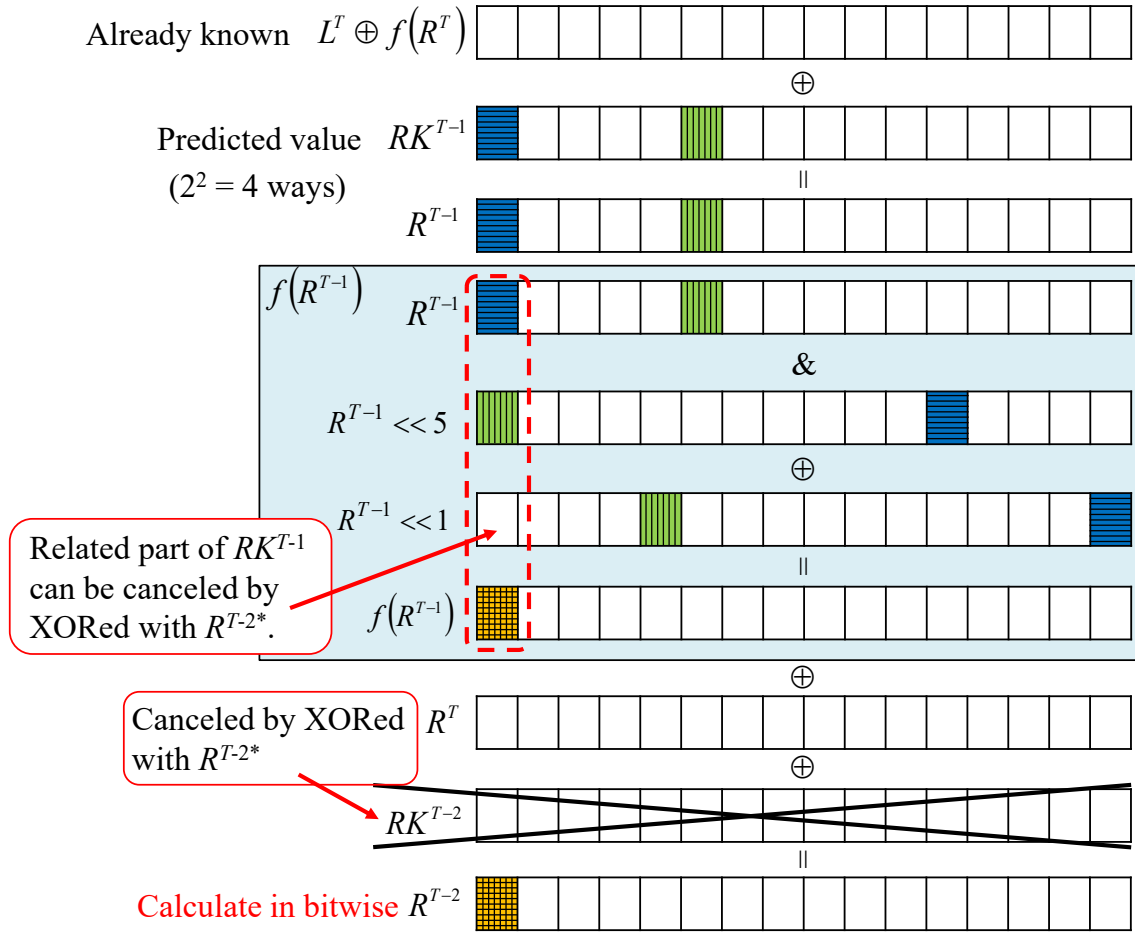


図 10.4 ビット単位での計算方法

Fig. 10.4 Calculation method in bitwise

$$R^{T-2} \begin{cases} = pR^{T-2} & (pRK^{T-1} = RK^{T-1}) \\ \neq pR^{T-2} & (pRK^{T-1} \neq RK^{T-1}), \end{cases} \quad R^{T-2*} \begin{cases} = pR^{T-2*} & (pRK^{T-1} = RK^{T-1}) \\ \neq pR^{T-2*} & (pRK^{T-1} \neq RK^{T-1}) \end{cases} \quad (10.9)$$

ここで、2つの値 a , b のハミング距離を求める関数を $HD(a, b)$ とすると、 R^{T-2} と R^{T-2*} のハミング距離 hR^{T-2} と、 pR^{T-2} と pR^{T-2*} のハミング距離 hpR^{T-2} は式(10.10)と式(10.11)で計算できる。そして、ハミング距離 hR^{T-2} , hpR^{T-2} には、式(10.12)に示す関係が成り立つ。

$$hR^{T-2} = HD(R^{T-2}, R^{T-2*}) \quad (10.10)$$

$$hpR^{T-2} = HD(pR^{T-2}, pR^{T-2*}) \quad (10.11)$$

$$hR^{T-2} \begin{cases} = hpR^{T-2} & (pRK^{T-1} = RK^{T-1}) \\ \neq hpR^{T-2} & (pRK^{T-1} \neq RK^{T-1}) \end{cases} \quad (10.12)$$

式(10.12)において、 $RK^{T-1}=pRK^{T-1}$ のとき、 pR^{T-2} と pR^{T-2*} のハミング距離 hpR^{T-2} はフォールト混入数と等しくなる。また、 R^{T-1*} はフォールト混入数が少ないため、ハミング距離は小さい値となる。

ここで、計算途中の暗号中間値の値によってフォールト混入数は異なる。これは、暗号中間値によってフリップフロップが確定するタイミングが異なるためである。そこで提案するフォールト解析手法では、十分に多い組である N 組の暗号文を使用して、このハミング距離の平均値 $\overline{HD}$ を計算することで、バラツキを平準化する。ハミング距離の平均値 $\overline{HD}$ は、それぞれ正常な暗号中間値とフォールト入り暗号文中間値の組からハミング距離 $hpR_1^{T-2}, \dots, hpR_N^{T-2}$ を求めて、式(10.13)によって計算する。

$$\overline{HD} = \frac{1}{N} \sum_{n=1}^N hpR_n^{T-2} \quad (10.13)$$

最後に、このハミング距離の平均値 $\overline{HD}$ を最小とするラウンド鍵の予測値 pRK^{T-1} を正解鍵として推定する。

10.2.4 2 段階目のフォールト解析手法

2 段階目のフォールト解析の概要を図 10.5 に示す。2 段階目のフォールト解析では、 R^{T-3} にはフォールトが混入しないという性質を利用して、ラウンド鍵 RK^{T-2} を推定する。具体的には、1 段階目のフォールト解析で推定したラウンド鍵 RK^{T-1} を利用して、 R^{T-1} と R^{T-1*} を求める。次に、この R^{T-1} と R^{T-1*} を利用して、 R^{T-3} と R^{T-3*} を式(10.14)、(10.15)によって計算する。

$$R^{T-3} = R^{T-1} \oplus f(L^{T-1} \oplus f(R^{T-1}) \oplus RK^{T-2}) \oplus RK^{T-2} \quad (10.14)$$

$$R^{T-3*} = R^{T-1*} \oplus f(L^{T-1*} \oplus f(R^{T-1*}) \oplus RK^{T-2}) \oplus RK^{T-2} \quad (10.15)$$

そして、この R^{T-3} と R^{T-3*} を利用して 1 段階目のフォールト解析と同様にして、ハミング距離を求める。このとき、 R^{T-3} にはフォールトが混入しないため、ラウンド鍵の予測値が正しい場合、 $R^{T-3}=R^{T-3*}$ が成り立つ。すなわち、 $HD(R^{T-3}, R^{T-3*})=0$ が成り立つ。

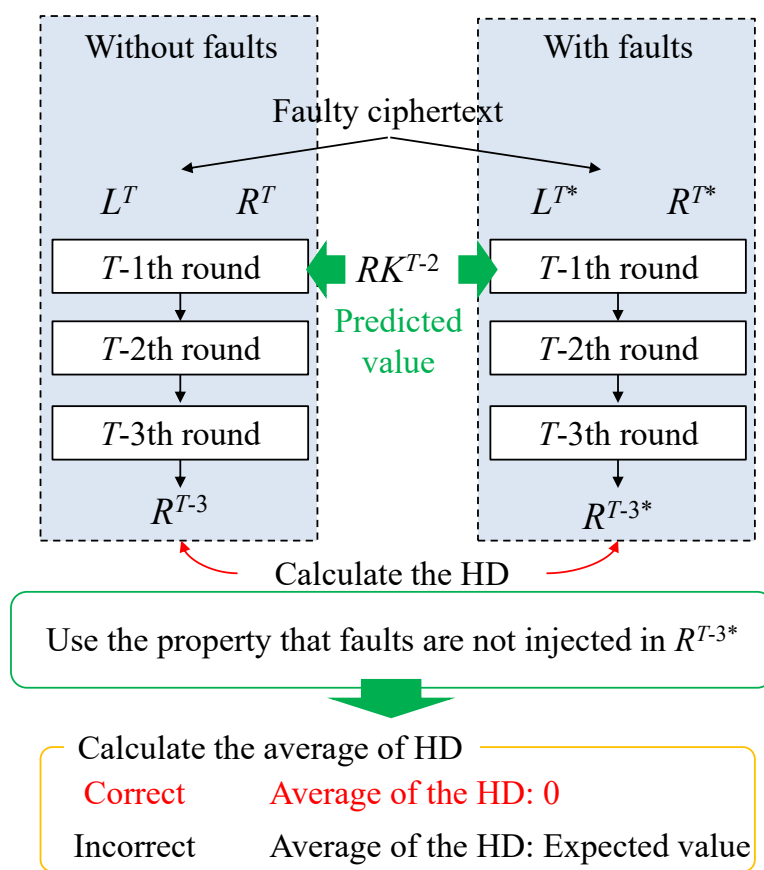


図 10.5 2 段階目のフォールト解析

Fig. 10.5 Fault analysis in the second stage

最後に、複数の暗号文の組を利用してハミング距離の平均値を導出して、このハミング距離の平均値を最小とするラウンド鍵 RK^{T-2} の予測値を正解鍵として推定する。

10.3 評価実験

10.3.1 実験環境

評価実験は、ブロック長が 32bit、鍵長が 64bit の SIMECK（ラウンド数：32（ $T=33$ ））を対象として行った。実験環境を表 1 に、実験方法を図 10.6 に示す。フォールトは L^{30} の各ビットでそれぞれ 10%, 20%, 30%と確率を変化させて、値を反転させることで生成した。実験ではラウンド鍵 RK^{31} と RK^{32} を対象として解析を行った。また、実験は暗号文の組を 100 組使用した解析を 100 回行い、その平均値を算出した。

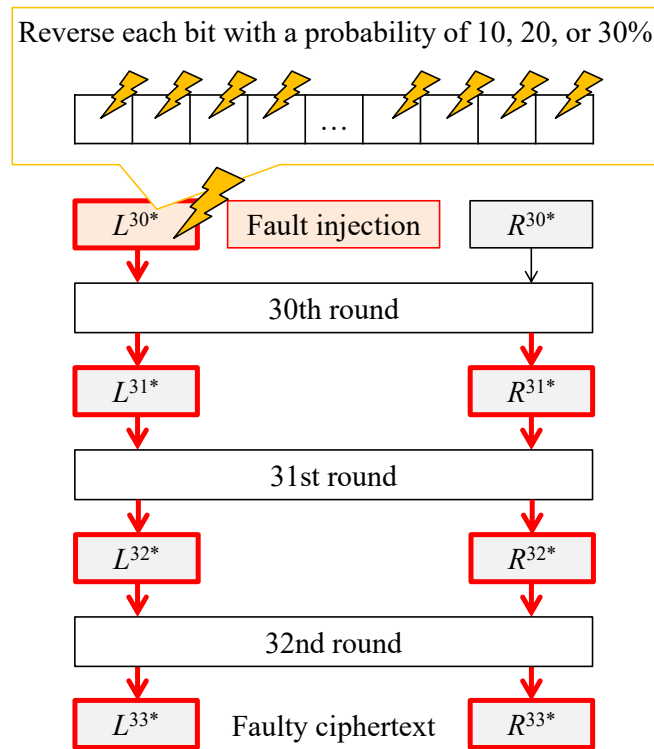


図 10.6 シミュレーションによるフォールト生成

Fig. 10.6 Fault generation for simulation

表 10.1 実験環境

Table 10.1 Experimental condition

Encryption algorithm	SIMECK
Block size [bit]	32
Key size [bit]	64
Secret key	Random
Plaintext	Random
Fault injection positions	$L^{30*} (= R^{31*})$
Fault injection probability	10, 20, or 30%
# of simulation	100
PC	HP ProBook 6570b
Memory [GB]	8
CPU	Core i7-3570M
Programming language	MATLAB 2013b

10.3.2 実験結果

まず，ラウンド鍵 RK^{32} を対象とした結果を図 10.7 に示す。図 10.7 の結果においては，100 組の暗号文の組を使用した解析では，フォールト発生確率が 10, 20, 30% のとき，それぞれ，15.56, 15.77, 15.35bit のラウンド鍵 RK^{32} の解析に成功した。ここで，解析に成功した鍵数が 16bit でないのは，使用する暗号文によって，フォールト混入数がそれぞれ異なるためだと考えられる。また，図 10.7 の実験結果では，フォールト発生確率が 20% のときが最も解析に成功した鍵数が高くなっている。したがって，1 段階目のフォールト解析は各ビットでのフォールト混入確率が 20% 程度の場合，最も解析効率が高くなると考えられる。

次に，ラウンド鍵 RK^{31} を対象とした結果を図 10.8 に示す。図 10.8 の結果においては，65 組を使用した解析によってフォールト発生確率の違いに関わらず，ラウンド鍵 RK^{31} の解析に成功していることが分かる。ここで，図 10.8 の結果においては，フォールト発生確率が 30% のとき，解析に必要な暗号文の組数が最も少ないことが分かる。これは 2 段階目のフォールト解析では， R^{T-3*} にフォールトが発生しないという性質を利用しており， L^{T-3*} でより多くのフォールトが混入することで，解析に有意な情報が増加するためだと考えら

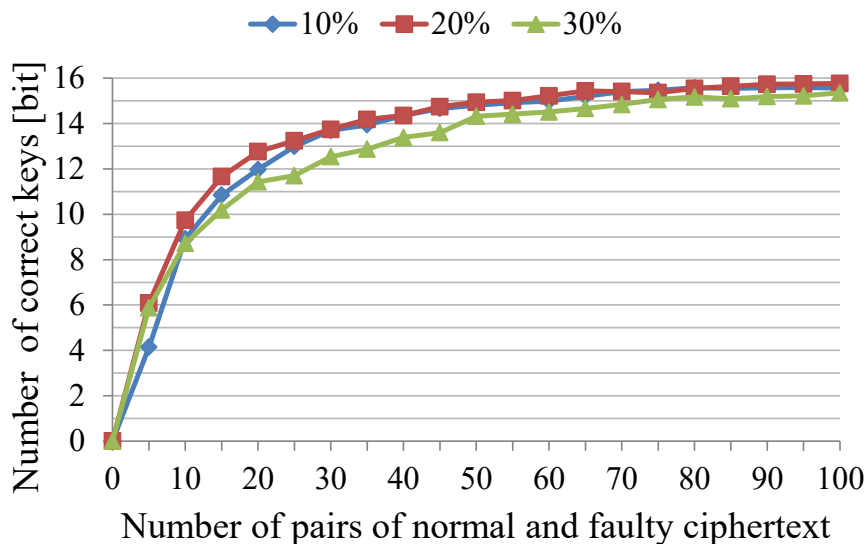


図 10.7 1 段階目のフォールト解析の実験結果

Fig. 10.7 Experimental results of the fault analysis in the first stage

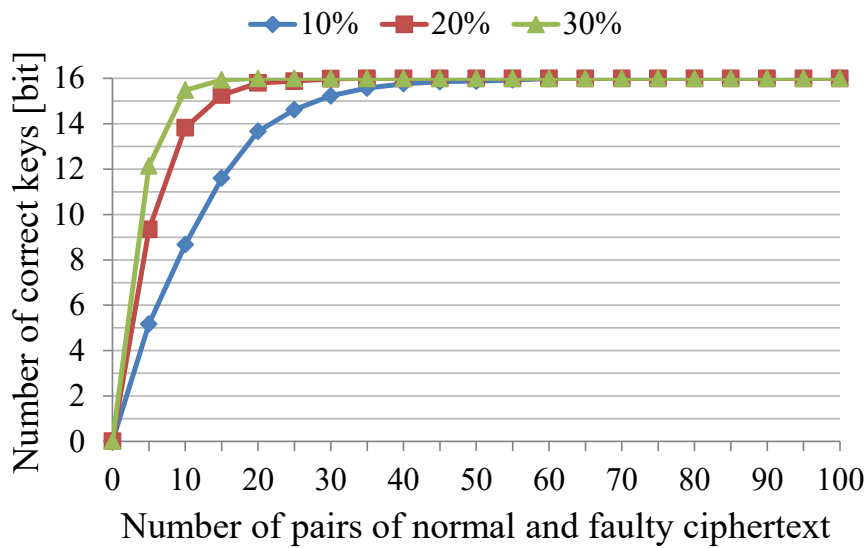


図 10.8 2 段階目のフォールト解析の実験結果

Fig. 10.8 Experimental results of the fault analysis in the second stage

れる。したがって、2 段階目のフォールト解析では各ビットでより多くのフォールトが混入することで、解析効率が向上すると推察される。以上より、本提案のフォールト解析手法によって、SIMECK の耐タンパ性を評価することができる。

10.3.3 平均ハミング距離の比較

平均ハミング距離について比較した。まず 1 段階目のフォールト解析における比較結果を図 10.9 に示す。図の横軸は鍵候補を、縦軸は平均ハミング距離を、赤色は正解鍵を、青色は不正解鍵をそれぞれ示している。図 10.9 の結果においては、正解鍵では平均ハミング距離が最も小さくなっていることが確認できる。一方で不正解鍵では、平均ハミング距離は期待値である 0.5 に近付いていることがわかる。

また、2 段階目のフォールト解析における比較結果を図 10.10 に示す。図 10.10 の結果においては、正解鍵では、平均ハミング距離が 0 であることが確認できる。これは、提案するフォールト解析では、 R^{T-3*} にはフォールトは混入しないというフォールトモデルを導入しているためである。

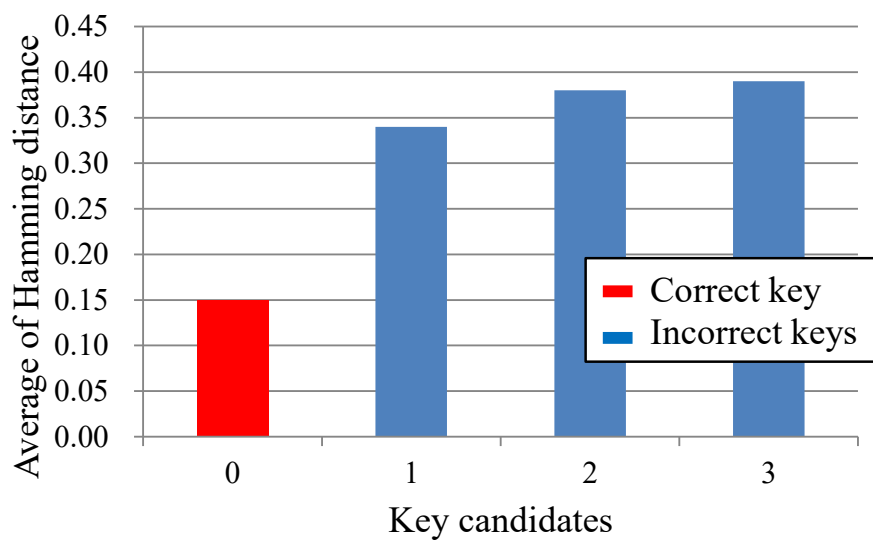


図 10.9 平均ハミング距離の比較（1 段階目のフォールト解析）

Fig. 10.9 Comparison of the average of Hamming distance for the fault analysis in the first stage

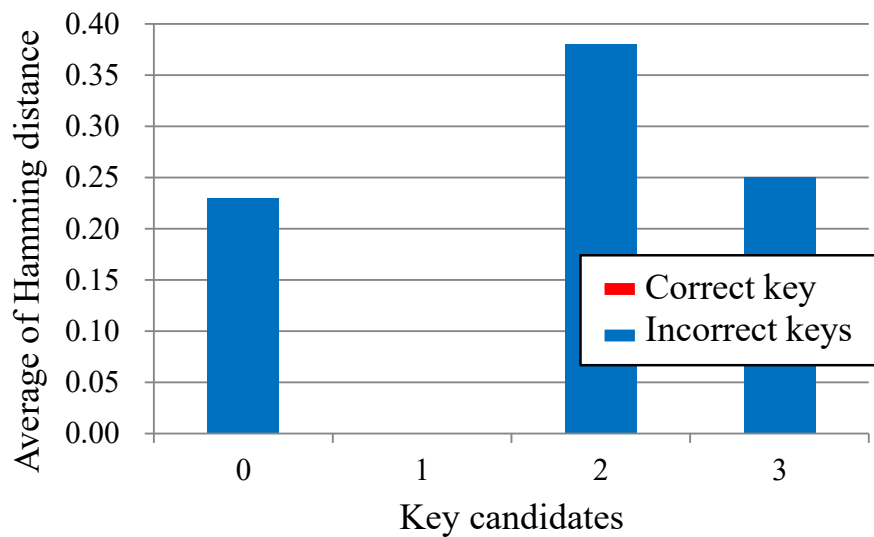


図 10.10 平均ハミング距離の比較（2 段階目のフォールト解析）

Fig. 10.10 Comparison of the average of Hamming distance for the fault analysis in the second stage

10.4 結言

本章では，SIMECK に対するフォールト解析手法を提案し，シミュレーション実験によってその有効性を示した。提案したフォールト解析手法では，クロックグリッチによるフォールト混入を想定したフォールトモデルを導入した。そして，統計的な解析によってラウンド鍵を推定し，提案したフォールト解析手法によって SIMECK の耐タンパ性を評価できることを示した。

参考文献

- [1] 野崎佑典, 吉川雅弥, “軽量暗号 Simeck に対するフォールト攻撃,” 電子情報通信学会技術研究報告, IEICE-CAS, vol. 115, no. 422, pp. 33–38, Jan. 2016.
- [2] G. Yang, B. Zhu, V. Suder, M. D. Aagaard, and G. Gong, “The Simeck Family of Lightweight Block Ciphers,” Proc. 17th Int. Workshop on Cryptographic Hardware and Embedded Systems (CHES 2015), LNCS vol. 9293, pp. 307–329, Springer, Sep. 2015.
- [3] V. Nalla, R. A. Sahu and V. Saraswat, “Differential Fault Attack on SIMECK,” Proc. 3rd Workshop on Cryptography and Security in Computing Systems (CS2’16), pp. 45–48, Jan. 2016.
- [4] 吉川英機, “ラウンド加算 DFA による Feistel 型ブロック暗号における鍵導出法, 電子情報通信学会論文誌 D,” vol. J101-D, no. 7, pp. 1027–1036, Jul. 2018.
- [5] 福永利徳, 高橋順子, “暗号 LSI に対するフォールト解析攻撃実験,” 2009 年暗号と情報セキュリティシンポジウム講演論文集, 2A3-5, pp. 1–6, Jan. 2009.
- [6] S. Endo, T. Sugawara, N. Homma, T. Aoki and A. Satoh, “A Configurable On-Chip Glitchy-Clock Generator for Fault Injection Experiments,” IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences, vol. E95-A, no. 1, pp. 263–266, Jan. 2012.

第 11 章

Midori に対するフォールト解析手法

11.1 緒言

本章では，Midori の耐タンパ性評価に必要なフォールト解析手法を提案する [1]。関連する研究として，これまでに Midori [2]に対する DFA [3]が報告されている。文献 [3]では，Midori の Cell 構造を指向したフォールトモデルを導入し，特定のラウンドの特定の位置に対してフォールトが混入することを仮定した解析を行う。ブロック長 64bit の Midori を対象としたシミュレーション実験では，48bit の鍵が復元できており，秘密鍵の鍵空間を 2^{128} 通りから 2^{80} 通りまで削減することに成功している。一方で，文献 [3]ではランダムな位置に複数のフォールトが混入する場合は，解析を行うことは難しい。更に，文献 [3]では全ての秘密鍵の解析は行われていない。

本章で提案するフォールト解析手法では，ランダムな位置に複数のフォールトが混入した際にも解析が可能な統計的な解析手法を導入する。また，階層的な解析を行うことで，全ての秘密鍵の推定を行う。

11.2 提案するフォールト解析手法

11.2.1 解析手法の概要

提案するフォールト解析手法の概要を図 11.1 に示す。提案手法では，Midori の 128bit の秘密鍵を全て解析するために，2 段階でのフォールト解析手法を導入する。すなわち，提案するフォールト解析手法は，1 段階目のフォールト解析と 2 段階目のフォールト解析

本章は[1]で発表された内容から構成される。

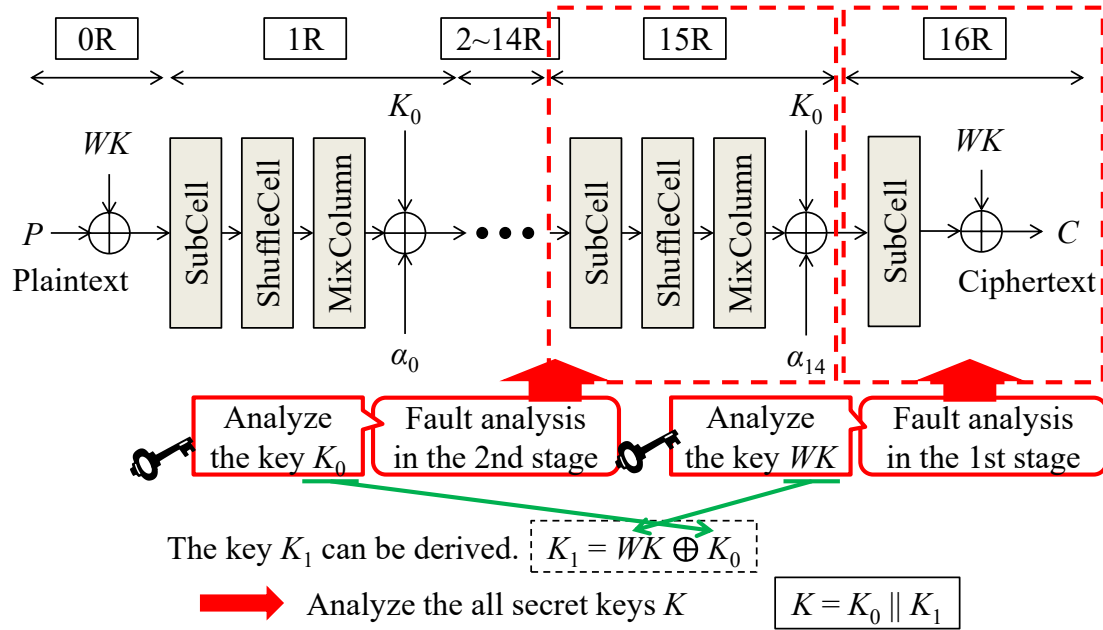


図 11.1 提案するフォールト解析の概要

Fig. 11.1 Outline of the proposed fault analysis

で構成する。まず1段階目のフォールト解析では、Midoriの16Rを対象として、鍵 WK を解析する。次に2段階目のフォールト解析では、15Rを対象として鍵 K_0 を解析する。このとき、2段階目のフォールト解析では、1段階目のフォールト解析で解析済の鍵 WK を用いる。そして、求めた鍵 WK と鍵 K_0 から鍵 K_1 を導出する。この計算式を式(11.1)に示す。したがって、2段階のフォールト解析を行うことで、最終的に鍵 K_0 と鍵 K_1 が求まるため、秘密鍵 K を全て求めることができる。

$$K_1 = WK \oplus K_0 \quad (11.1)$$

11.2.2 フォールトモデル

提案するフォールト解析手法では、2段階のフォールト解析を行うため、2種類のフォールトモデル（フォールトモデルⅠとフォールトモデルⅡ）を導入する。各フォールトモデルを図11.2と図11.3に示す。また、提案するフォールト解析手法では、フォールト混入方法として、容易に実現可能なクロックグリッチ [4][5]の挿入を想定する。3.4.3項でも述べた通りクロックグリッチを挿入することで、フリップフロップにセットアップタイム制約違反を引き起こし、複数のフォールトを混入させることができる。ここで、クロックグリ

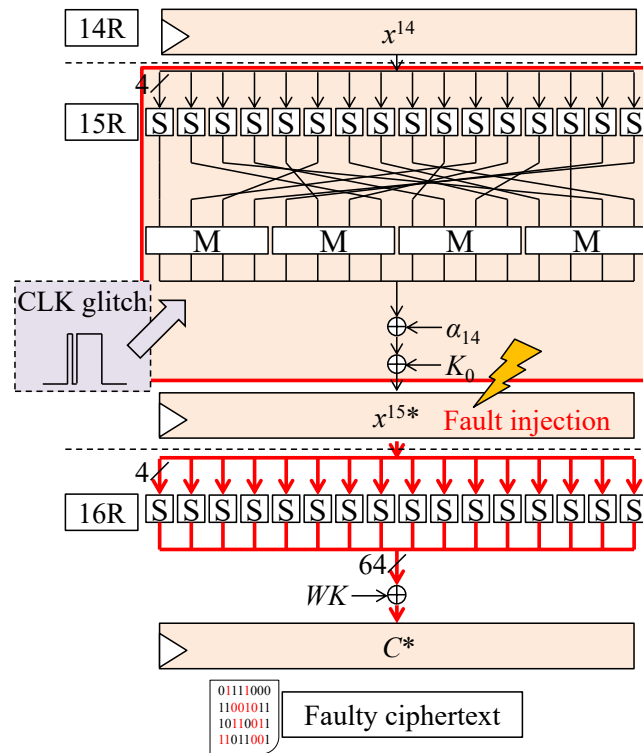


図 11.2 フォールトモデル I

Fig. 11.2 Fault model I

ッチは暗号回路全体に供給されるため、フォールト混入位置やフォールト混入数を制御することは難しい。

このとき、全フリップフロップでセットアップタイム制約違反が発生した場合、フリップフロップのビット値の変化パターンは、 $0 \rightarrow 0$, $0 \rightarrow 1$, $1 \rightarrow 0$, $1 \rightarrow 1$ の 4 通りである。ここで、ビット値が反転する確率（フォールト混入確率）は $1/2$ である。なぜなら、ビット値が反転するパターンは、 $0 \rightarrow 1$ と $1 \rightarrow 0$ の 2 通りだからである。このことについて、Midori のレジスタ間のビットの遷移数に関する予備実験を行い検証した。検証結果を図 11.4 に示す。図 11.4 は Midori の 14R と 15R のレジスタ間のビット遷移について、1,000 種類のランダム平文と 10 種類のランダム秘密鍵の 10,000 種類の暗号化についての結果を示したものである。図 11.4 に示すように、64bit のデータにおいて、ビット遷移の平均は約 32bit となっており、約 $1/2$ の確率でビットが遷移していることが分かる。そのため、全てのフリップフロップでセットアップタイム制約違反が発生した場合、約 $1/2$ の確率でビット値が反転する。しかし、実際の暗号回路では、組み合わせ回路に入力される暗号中間値によって信号遅延は異なるため、フリップフロップのビット値の確定するタイミングは異なる。す

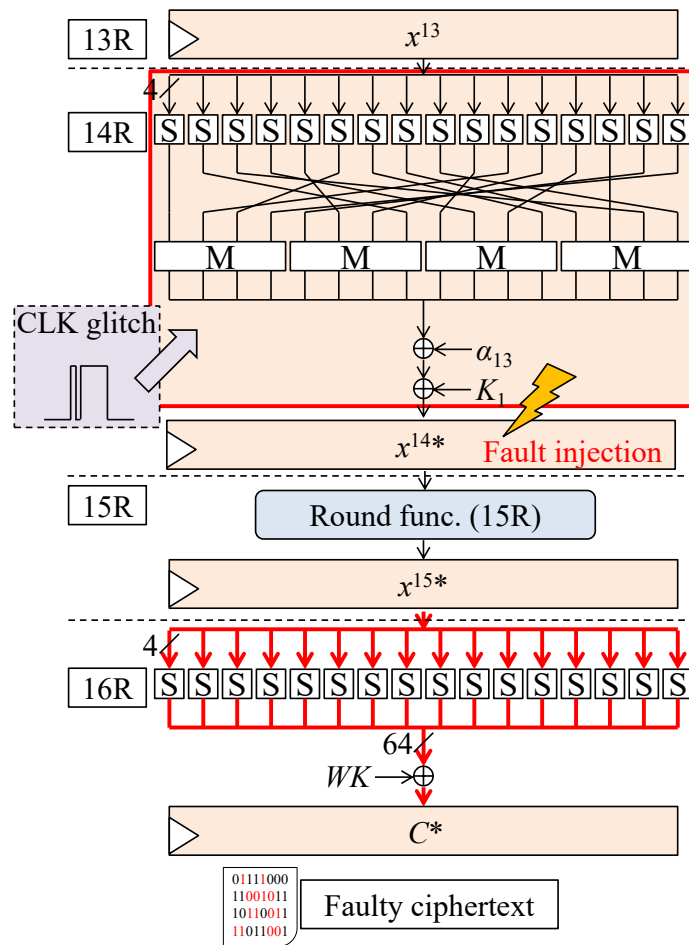


図 11.3 フォールトモデル II

Fig. 11.3 Fault model II

なわち、全フリップフロップでセットアップタイム制約違反は発生しない。この組み合わせ回路の信号遅延は、回路のゲート遅延や配線遅延により発生するもので、通常的设计で生じるものである。したがって、クロックグリッチによるフォールト混入確率は 1/2 よりも小さくなる。

図 11.2 に示すように、1 段階目のフォールト解析では 15R を対象にクロックグリッチを挿入し、15R 計算終了後のデータレジスタに複数のフォールトを混入させるフォールトモデル（フォールトモデル I）を使用する。また、図 11.3 に示すように、2 段階目のフォールト解析では 14R を対象にクロックグリッチを挿入し、14R 計算終了後のデータレジスタに複数のフォールトを混入させるフォールトモデル（フォールトモデル II）を使用する。

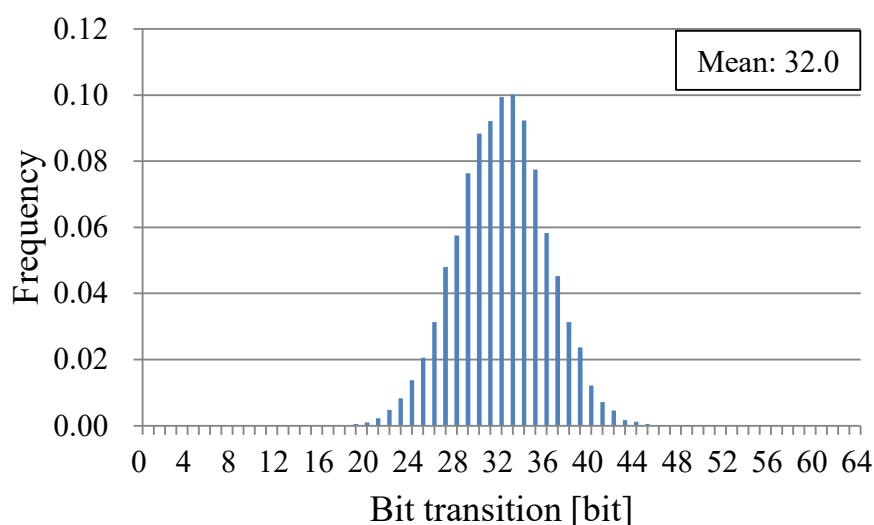


図 11.4 データレジスタ間のビット遷移の頻度

Fig. 11.4 Frequency of bit transition between data registers

11.2.3 1 段階目のフォールト解析手法

1 段階目のフォールト解析では、フォールトモデル I を用いた解析を行い、鍵 WK を推定する。解析に際して、まず通常の暗号文 C とフォールト入り暗号文 C^* の組 (暗号文ペア) を複数個用意する。そして、図 11.5 (i) に示すように暗号文ペアから 15R 計算終了後の暗号中間値 (通常の暗号中間値 x^{15} とフォールト入り暗号中間値 x^{15*}) をそれぞれ計算する。各暗号中間値は、既知の暗号文 C , C^* と鍵 WK の予測値を用いた計算で求めることができる。次に、求めた通常の暗号中間値 x^{15} とフォールト入り暗号中間値 x^{15*} とのハミング距離を計算する。このとき、鍵 WK の予測値が正解値の場合、導出したハミング距離は、フォールト混入数と等しくなる。また、11.2.2 項で述べた通り、クロックグリッチによるフォールト混入確率は $1/2$ よりも小さく、4bit 単位の値で考えたとき、フォールト混入数は 4bit の期待値である 2 よりも少なくなる。したがって、正解鍵の場合、ハミング距離は 2 よりも小さくなる。一方で、鍵 WK の予測値が不正解値の場合、導出したハミング距離は 4bit の期待値である 2 に近づく。提案するフォールト解析手法では、複数個の暗号文ペアを使用して平均ハミング距離を導出し、この平均ハミング距離を最小とする鍵 WK の予測値を正解鍵として推定する。

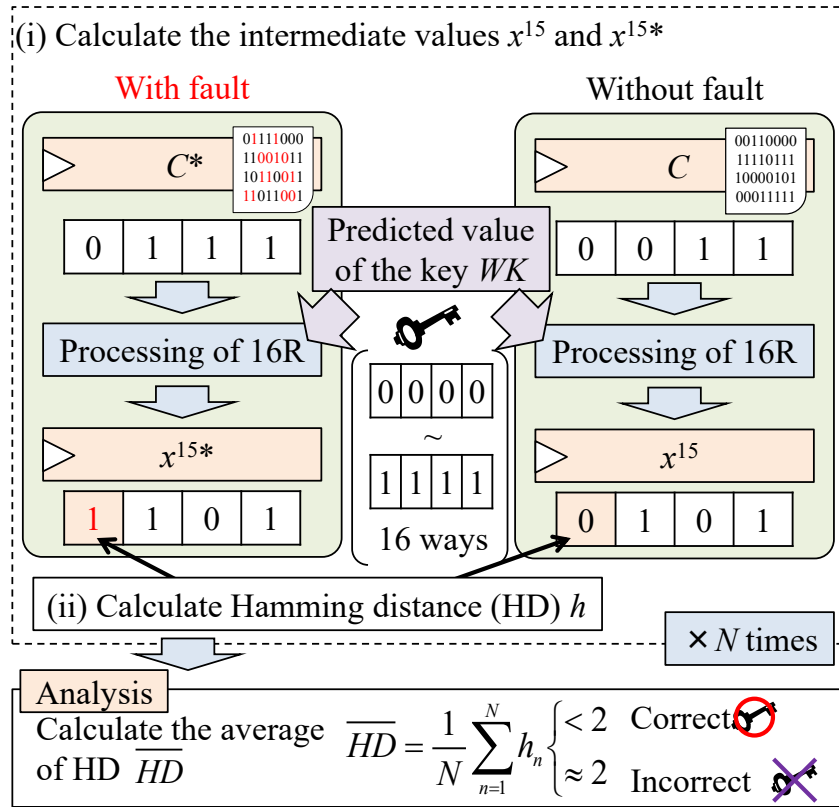


図 11.5 1 段階目のフォールト解析

Fig. 11.5 Fault analysis in the first stage

1 段階目のフォールト解析の処理の詳細について説明する。1 段階目のフォールト解析の計算処理を図 11.6 に示す。図 11.6 に示すように、各暗号中間値 x^{15} と x^{15*} は式(11.2)で計算することができる。このとき、式(11.2)の $S()$ は S-BOX の置換処理 (A.3 節、表 A.5 を参照) を表している。

$$\begin{aligned} x^{15} &= S(C \oplus WK) \\ x^{15*} &= S(C^* \oplus WK) \end{aligned} \quad (11.2)$$

また、鍵 WK は未知の値であるため、予測値として pWK を与える。この予測値は、式(11.2)の計算が 4bit 単位で行われていることから、 $2^4 = 16$ 通りの値 ($0 \leq pWK \leq 15$) を全て試す。この計算結果を X , X^* とすると、 X , X^* は式(11.3)で計算することができる。

$$\begin{aligned} X &= S(C \oplus pWK) \\ X^* &= S(C^* \oplus pWK) \end{aligned} \quad (11.3)$$

ここで、鍵 WK の予測値 pWK が正解値の場合 ($pWK = WK$)、計算結果 X , X^* はそれぞれ各暗号中間値 x^{15} , x^{15*} と等しくなり、式(11.4)に示す関係が成り立つ。

$$x^{15} \begin{cases} = X & (pWK = WK) \\ \neq X & (pWK \neq WK), \end{cases} \quad x^{15*} \begin{cases} = X^* & (pWK = WK) \\ \neq X^* & (pWK \neq WK) \end{cases} \quad (11.4)$$

次に、各暗号中間値 x^{15} と x^{15*} のハミング距離 h を計算する。この計算式を式(11.5)に示す。

$$h = HD(x^{15}, x^{15*}) \quad (11.5)$$

また、鍵 WK の予測値 pWK を用いた計算結果 X と X^* のハミング距離 hX は式(11.6)で計算できる。

$$hX = HD(X, X^*) \quad (11.6)$$

そして、式(11.5)と式(11.6)で求めたハミング距離 h と hX には、式(11.7)に示す関係が成り立つ。

$$h \begin{cases} = hX & (pWK = WK) \\ \neq hX & (pWK \neq WK) \end{cases} \quad (11.7)$$

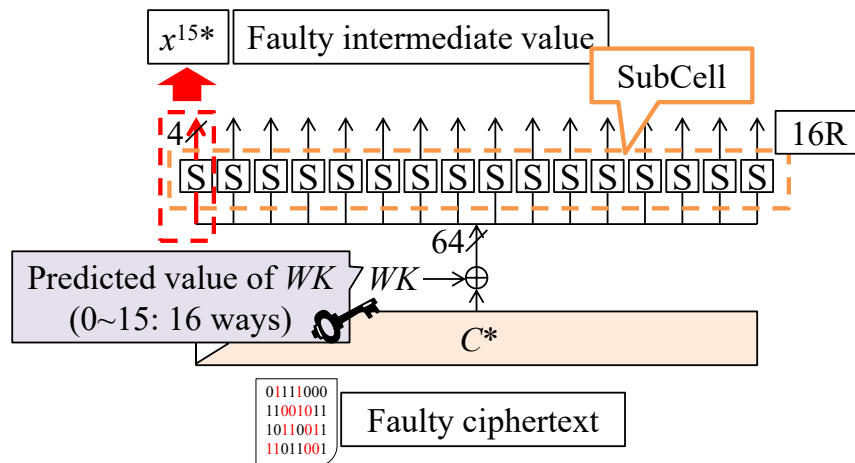


図 11.6 1 段階目のフォールト解析での計算

Fig. 11.6 Calculation for the fault analysis in the first stage

ここで、鍵 WK の予測値 pWK が正解値の場合 ($pWK = WK$)、導出したハミング距離 hX はフォールト混入数と等しくなる。一方で、鍵 WK の予測値 pWK が不正解値の場合 ($pWK \neq WK$)、導出したハミング距離 hX は 4bit の期待値である 2 に近づく。

提案するフォールト解析手法では、 N 個の暗号文ペアを使用して、 N 個のハミング距離 $hX_1, \dots, hX_N$ を計算し、平均ハミング距離 $\overline{HD}$ を導出する。この計算式を式(11.8)に示す。そして、この平均ハミング距離 $\overline{HD}$ を最小とする鍵 WK の予測値 pWK を正解鍵として推定する。

$$\overline{HD} = \frac{1}{N} \sum_{n=1}^N hX_n \begin{cases} < 2 & (pWK = WK) \\ \approx 2 & (pWK \neq WK) \end{cases} \quad (11.8)$$

11.2.4 2 段階目のフォールト解析手法

2 段階目のフォールト解析では、フォールトモデル II を用いて、鍵 K_0 を解析する。2 段階目のフォールト解析の概要を図 11.7 に示す。2 段階目のフォールト解析では、1 段階目のフォールト解析で解析済の鍵 WK と既知の暗号文 C 、 C^* を利用して、15R 計算終了後の暗号中間値 x^{15} 、 x^{15*} を、式(11.2)を用いて計算する。

そして、図 11.7 に示すように、暗号中間値 x^{15} 、 x^{15*} から 14R 計算終了後の暗号中間値 x^{14} 、 x^{14*} を計算する。この計算では、まず AddKey と MixColumn の処理を適用し、計算結果 y 、 y^* を得る。このとき、MixColumn は 16bit 単位で処理を行うため、 y 、 y^* は 16bit の値である。そして、 y 、 y^* を 4bit ずつに分割した値 y_1, y_2, y_3, y_4 ($y = y_1 \parallel y_2 \parallel y_3 \parallel y_4$) と $y_1^*, y_2^*, y_3^*, y_4^*$ ($y^* = y_1^* \parallel y_2^* \parallel y_3^* \parallel y_4^*$) に対して SubCell を適用し、暗号中間値 x^{14} 、 x^{14*} を計算する。この計算式を式(11.9)と式(11.10)に示す。ここで、 $M()$ は MixColumn の処理 (A.3 節, 式(A.13) を参照) を表している。また、鍵 K_0 には予測値として $2^{16} = 65536$ 通りの値を全て試行する。

$$\begin{aligned} y &= M(x^{15} \oplus \alpha_{14} \oplus K_0) \\ y^* &= M(x^{15*} \oplus \alpha_{14} \oplus K_0) \end{aligned} \quad (11.9)$$

$$\begin{aligned} x^{14} &= S(y_1) \parallel S(y_2) \parallel S(y_3) \parallel S(y_4) \\ x^{14*} &= S(y_1^*) \parallel S(y_2^*) \parallel S(y_3^*) \parallel S(y_4^*) \end{aligned} \quad (11.10)$$

そして、1段階目のフォールト解析と同様にして、通常の暗号中間値 x^{14} とフォールト入り暗号中間値 x^{14*} のハミング距離を計算する。この計算を N 個の暗号文ペアに対して行い、平均ハミング距離を導出する。最後に、この平均ハミング距離を最小とする鍵 K_0 の予測値を正解鍵として推定する。以上より、鍵 WK と鍵 K_0 が求まることから鍵 K_1 が求まり、秘密鍵 K を構成する全ての鍵を求めることができる。すなわち、128bit の秘密鍵 K を全て導出することができる。

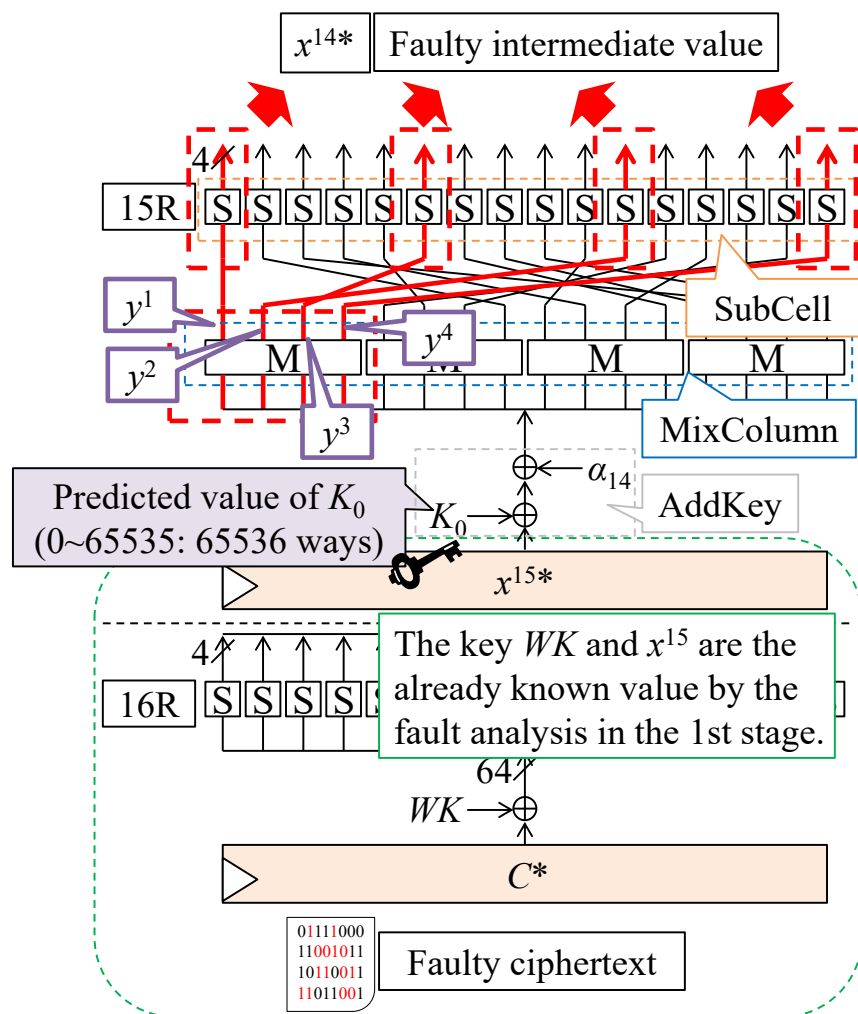


図 11.7 2段階目のフォールト解析での計算

Fig. 11.7 Calculation for the fault analysis in the second stage

11.4 評価実験

11.4.1 実験環境

評価実験は、ブロック長が 64bit の Midori を対象に、シミュレーションによって行った。実験環境を表 11.1 に示す。また、フォールト混入方法を図 11.8 に示す。図 11.8 に示すように、1 段階目のフォールト解析では 15R 計算終了後の暗号中間値 x^{15} を、2 段階目のフォールト解析では 14R 計算終了後の暗号中間値 x^{14} を対象として、ビット値を反転させることで、フォールトを混入させた。このフォールト混入では、各ビット値が反転する確率（フォールト混入確率）をそれぞれ 10%, 20%, 30% と変化させた。したがって、ランダムな位置で複数のフォールトが混入する。

表 11.1 実験環境

Table 11.1 Experimental condition

Encryption algorithm	Midori
Block size [bit]	64
Key size [bit]	128
Plaintext	Random
Secret key	Random
Probability of fault injection	10, 20, and 30%
PC	HP ProBook 6570b
Memory [GB]	8
CPU	Intel core i7 3570M
Programming language	MATLAB 2013b

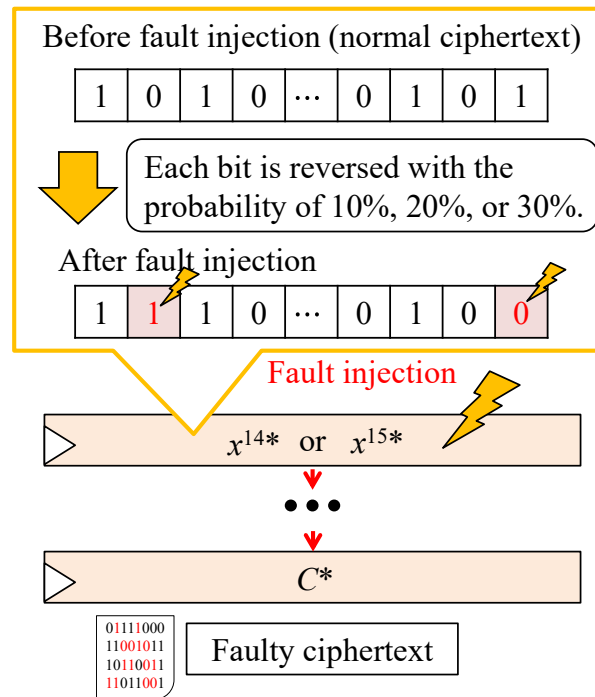


図 11.8 シミュレーションによるフォールト混入方法

Fig. 11.8 Fault injection method for simulation

11.4.2 実験結果

1 段階目のフォールト解析の結果を図 11.9 に、2 段階目のフォールト解析の結果を図 11.10 にそれぞれ示す。それぞれ図の縦軸は解析に成功した鍵のビット数を、横軸は解析に使用した暗号文ペアの数を示している。まず、図 11.9 に示すように、1 段階目のフォールト解析では、80 組の暗号文ペアを使用することで、どのフォールト混入確率においても、鍵 WK を全て解析することに成功した。したがって、1 段階目のフォールト解析が有効であることがわかる。次に、2 段階目のフォールト解析では、図 11.10 に示すように、60 組の暗号文ペアを使用することで、どのフォールト混入確率においても、鍵 K_0 を全て解析することに成功した。以上より、鍵 WK と鍵 K_0 の解析に成功したことから、式(11.1)によって 128bit の秘密鍵 K を全て求めることに成功した。したがって、提案するフォールト解析手法は有効であり、提案するフォールト解析手法を用いることで、Midori の耐タンパ性を評価することができる。

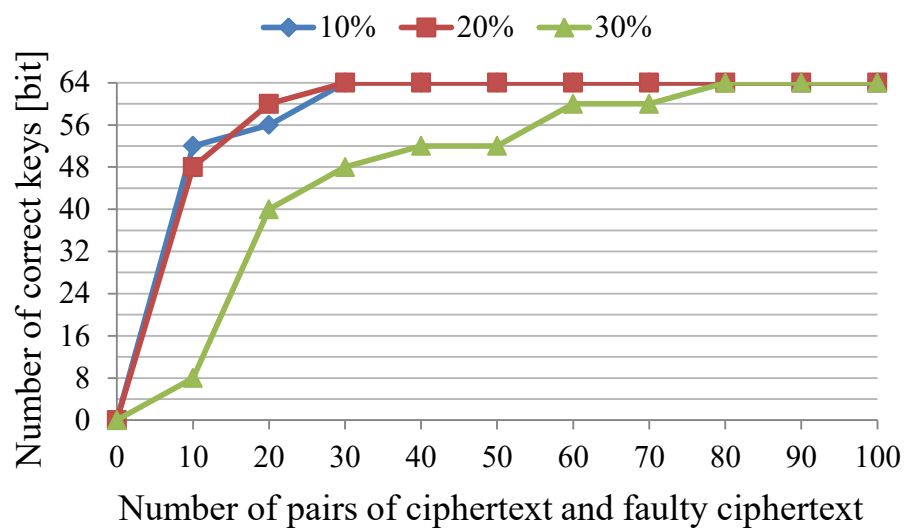


図 11.9 1 段階目のフォールト解析の結果

Fig. 11.9 Experimental results of the fault analysis in the first stage

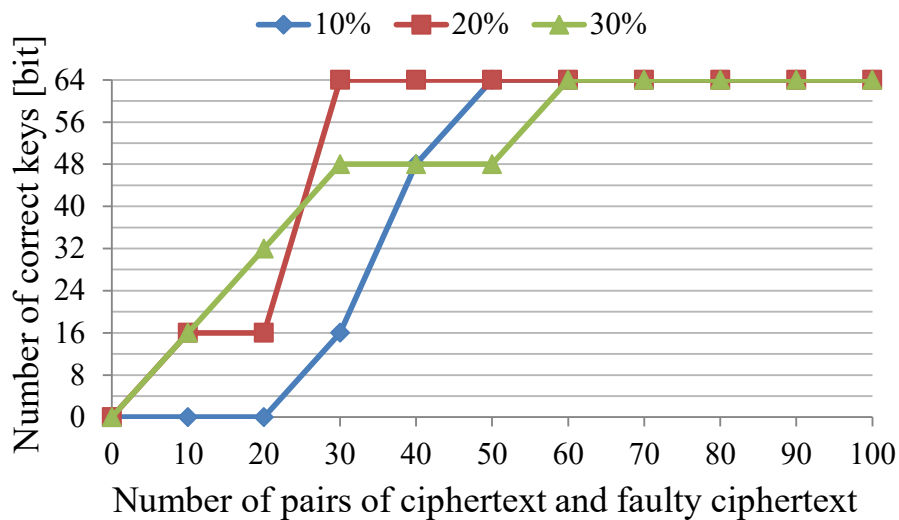


図 11.10 2 段階目のフォールト解析の結果

Fig. 11.10 Experimental results of the fault analysis in the second stage

11.4.3 平均ハミング距離の比較

各鍵の予測値における平均ハミング距離について比較した。1 段階目のフォールト解析における平均ハミング距離の比較結果を図 11.11 に示す。図 11.11 の横軸は鍵 WK の予測値（0 から 15 の 16 通り）を、縦軸は平均ハミング距離を示している。また、図 11.11 の赤色は鍵 WK の正解値を、青色は不正解値を示している。図 11.11 に示すように、鍵 WK が正解値の場合、平均ハミング距離が最小となっていることが確認できる。一方で、鍵 WK が不正解の場合、平均ハミング距離は 2 に近い値となっていることが分かる。

次に、2 段階目のフォールト解析における平均ハミング距離の比較結果を図 11.12 に示す。この比較では、平均ハミングが小さいものを顕著にするため、ハミング距離の特徴量を計算した。この特徴量は、平均ハミング距離とハミング距離の中央値（16bit の半分である 8）との差の二乗を計算することで算出する。特徴量は、平均ハミング距離が小さくなるほど大きくなり、中央値（8）に近づくほど小さくなる。ここで、図 11.12 の横軸は鍵 K_0 の予測値（0 から 65535 の 65536 通り）を、縦軸はハミング距離の特徴量を示している。図 11.12 に示すように、鍵 K_0 が正解値の場合、ハミング距離の特徴量は最大となっていることが確認できる。これは平均ハミング距離が最小となっていることを示している。

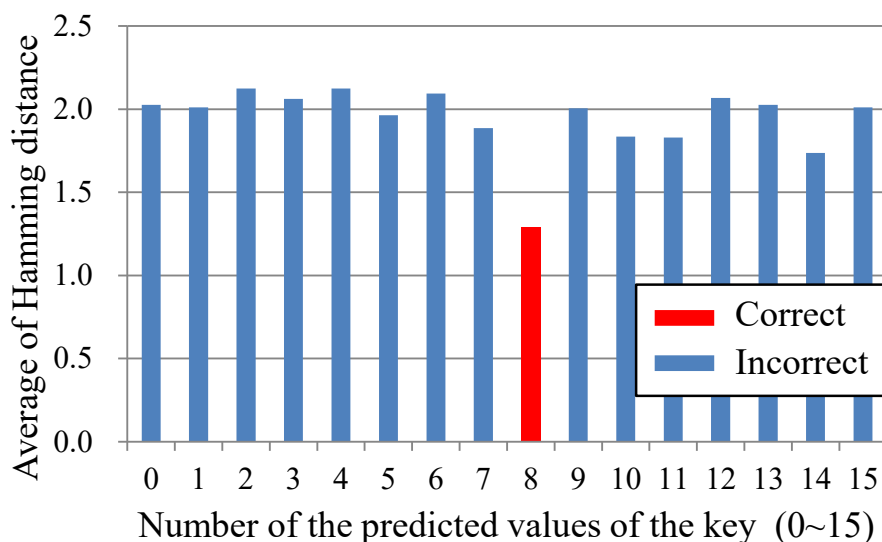


図 11.11 1 段階目のフォールト解析における平均ハミング距離の比較結果

Fig. 11.11 Comparison results of the average of Hamming distance of the fault analysis in the first stage

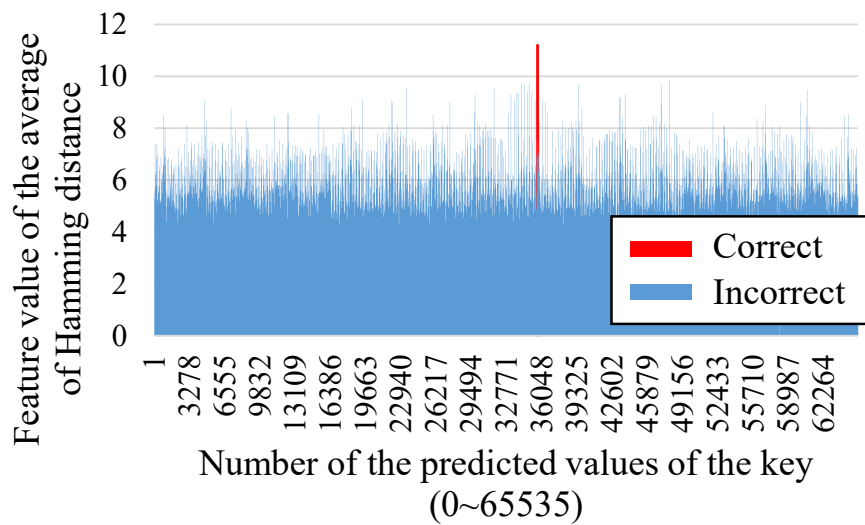


図 11.12 2 段階目のフォールト解析におけるハミング距離の特徴量の比較結果

Fig. 11.12 Comparison results of the feature value of the fault analysis in the second stage

11.4.4 フォールト混入の有効性検証

提案するフォールト解析手法では、クロックグリッチによるフォールト混入を想定とした解析を行っている。そこで、クロックグリッチによるフォールト混入の実用性について検証した。この検証では、Midori をハードウェア記述言語 Verilog HDL で設計し、Xilinx 社の Isim の遅延付きシミュレーションを行った。検証結果を図 11.13 に示す。図 11.13 は Midori の 15R の処理を示したものであり、Dnext は暗号中間値を表している。図 11.13 の結果から、暗号中間値の値が確定するのに必要な時間（クリティカルパス）は 3.231 [nsec] である。ここで、文献 [6]では、回路のクリティカルパスに対して、それよりも短いクロック周期で処理を打ち切った際に故障を混入させることができる。また、文献 [7]では 2.3 [nsec]のクロックグリッチの挿入に成功している。したがって、文献 [7]で用いたような 3.231 [nsec]よりも短い周期のクロックグリッチを挿入することで、解析に有効なフォールトを混入させることが可能である。

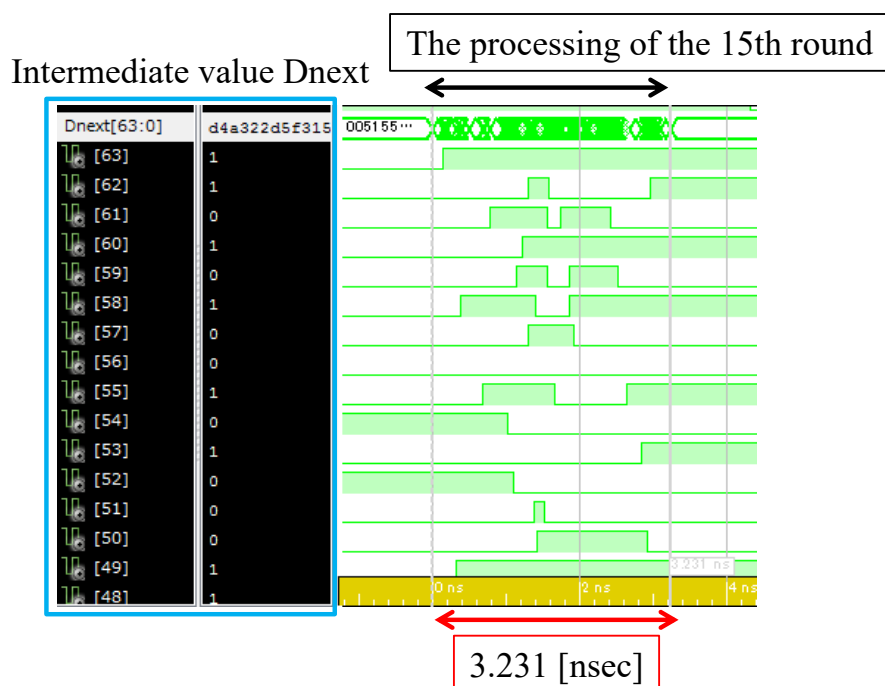


図 11.13 遅延付きシミュレーションの結果

Fig. 11.13 Result of delay simulation

11.5 結言

本章では，Midori に対するフォールト解析手法を提案し，シミュレーションによる評価実験によってその有効性を明らかにした。提案した Midori に対するフォールト解析手法では，クロックグリッチによるフォールト混入を想定し，得られた通常の暗号文とフォールト入り暗号文を複数個利用した統計的な解析を行う。解析では，16R と 15R を対象とした 2 段階の階層的な解析を導入することで 128bit の秘密鍵全てを推定する。また，Isim による遅延付きシミュレーションを用いた実験によって，解析に有効なフォールトの混入が可能であることを示した。

以上より，提案したフォールト解析手法を用いることで，Midori の耐タンパ性を評価できることを示した。

参考文献

- [1] 野崎佑典, 池崎良哉, 吉川雅弥, “Midori に対する 2 段階の統計的故障利用解析手法とその評価,” 電気学会論文誌 C, vol. 137, no. 12, pp. 1554–1561, Dec. 2017.
- [2] S. Banik, A. Bogdanov, T. Isobe, K. Shibutani, H. Hiwatari, T. Akishita, and F. Regazzoni, “Midori: A Block Cipher for Low Energy,” Proc. ASIACRYPT 2015, LNCS vol. 9453, pp. 411–436, Springer, Dec. 2015.
- [3] W. Cheng, Y. Zhou, and L. Sauvage, “Differential Fault Analysis on Midori,” Proc. 18th Int. Conf. Information and Communications Security (ICICS 2016), LNCS vol. 9977, pp. 307–317, Springer, Dec. 2016.
- [4] 福永利徳, 高橋順子, “暗号 LSI に対するフォールト解析攻撃実験,” 2009 年暗号と情報セキュリティシンポジウム講演論文集, 2A3-5, pp. 1–6, Jan. 2009.
- [5] S. Endo, T. Sugawara, N. Homma, T. Aoki and A. Satoh, “A Configurable On-Chip Glitchy-Clock Generator for Fault Injection Experiments,” IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences, vol. E95-A, no. 1, pp. 263–266, Jan. 2012.
- [6] 菅原 健, 鈴木大輔, “Fault Sensitivity Analysis のための回路シミュレーション,” 2012 年暗号と情報セキュリティシンポジウム講演論文集, 3C3-1, pp. 1–6, Jan. 2012.
- [7] 遠藤 翔, “故障注入を用いた暗号モジュールの動作解析に関する研究,” 博士学位論文, 東北大学大学院情報基礎科学専攻, 2015.

第 12 章

PRINCE に対するフォールト解析手法

12.1 緒言

本章では，PRINCE の耐タンパ性評価に必要なフォールト解析手法を提案する [1]。PRINCE [2]は，軽量暗号の 5 大要素の 1 つで，低遅延で動作可能な軽量暗号であり，CRYPTREC 軽量暗号 WG の暗号技術ガイドライン（軽量暗号） [3]の評価対象になっている。PRINCE は低遅延で動作するため，メモリ暗号化や車載システムのリアルタイム処理での利用が期待されている。PRINCE のアルゴリズムの詳細については，付録（A.4 節）で述べる。

関連する研究として，これまでに PRINCE に対する DFA が報告されている [4][5]。文献 [4][5]では，PRINCE の *S* 処理，定数加算処理，*SR* 処理のいずれか 1 つの処理だけに対し，複数のフォールトを混入させることで秘密鍵を解析している。文献 [4][5]の解析では，特定の処理，すなわち特定の位置に対してフォールトを混入させる必要があるため，レーザー照射 [6]のような時間的・空間的に精密な制御が可能なフォールト混入手法が必要である。このとき，レーザー照射は解析対象のレイアウト情報を参照することで，フォールトを混入させるタイミングや位置などを制御する。一方で，実際には解析者がレイアウト情報を持っていない場合も想定される。このような状況においては，レーザー照射を用いた場合でも解析に必要なフォールトを混入させることは難しい。

本章で提案するフォールト解析では，レイアウト情報を持っておらず，精密にフォールトの混入位置を制御できない場合においても適用可能なフォールト解析手法を導入する。

本章は[1]で発表された内容から構成される。

12.2 フォールトモデル

提案するフォールト解析手法では、2つのフォールトモデル（フォールトモデルⅠとフォールトモデルⅡ）を導入する。各フォールトモデルをそれぞれ図12.1と図12.2に示す。フォールトモデルⅠは、図12.1に示すように9ラウンド目（9R）計算終了後のレジスタに対して、フォールトを混入させるフォールトモデルである。フォールトモデルⅡは、図12.2に示すように8R計算終了後のレジスタに対して、フォールトを混入させるフォールトモデルである。

各フォールトモデルでは、図12.1と図12.2に示すように、暗号中間値64bitを16bitずつの4つのゾーン（ゾーン1からゾーン4）に分割する。図12.1の例では、 $x_{(1-16)}^{10}$ が、図12.2の例では、 $x_{(1-16)}^{11}$ が対応する部分が、ゾーン1となる。ここで、 $x_{(j-k)}^r$ は r ラウンド目の中間値における j bit目から k bit目のデータを表している。そして、フォールトの混入には、レーザー照射による1ビットのビットフリップを想定する。また、提案するフォールト解析手法では、解析者が解析対象のレイアウト情報を持っておらず、レーザー照射による解析対象の大まかなレイアウト情報の解析 [7]が可能であることを前提とする。この大まかなレイアウト情報では、解析対象のどの部分がレジスタ部分とロジック部分（ラウンド関数の各処理）であるのかだけを把握することが可能であり、ロジック部分のどの部分でどのような処理を行っているかまでは把握することはできない。そして、レジスタ部分に対してランダムにフォールトを混入させることで、4つのゾーンに対応するレジスタ部分のどこかに、ランダムに1ビットのフォールトが混入する。このとき、少なくとも1つはフォールトの混入しないゾーンが存在する。すなわち、どの位置にレーザーを照射しても、フォールトが混入しているゾーンは1つで、残りの3つのゾーンにはフォールトは混入していない。提案するフォールト解析手法では、このフォールトが混入していないゾーンに着目して解析を行う。

文献 [4][5]では、フォールト解析を行うために、特定の位置（ S 処理、定数加算処理、 SR 処理のいずれか1つの処理にだけ）にフォールトを混入させる必要がある。一方で、解析に必要なレイアウト情報については、ロジック部分がどの部分にあるかしか把握できない。したがって、ロジック部分のどの部分で S 処理、定数加算処理、 SR 処理を行っているのかを把握することができないため、文献 [4][5]の解析で必要となるフォールトを混入させることは難しいと考えられる。しかし、提案するフォールト解析手法にはそのような制

約は無く、ランダムなレジスタに対して、レーザーによるフォールトを混入させることで解析を行うことが可能である。

図 12.1 に示すフォールトモデル I では、9R 計算終了後のレジスタのゾーン 4 に対応する部分にフォールトが混入しているため、10R の入力値のゾーン 4 の値は、フォールト入り中間値となる。したがって、10R の計算によって、フォールトは伝搬し、最終的にフォールト入り暗号文を取得することができる。

一方、図 12.2 に示すフォールトモデル II では、8R 計算終了後のレジスタのゾーン 4 に対応する部分にフォールトが混入しているため、9R の入力値のゾーン 4 の値はフォールト入り中間値となる。したがって、9R と 10R の計算によってフォールトは伝搬し、最終的にフォールト入り暗号文を取得することができる。

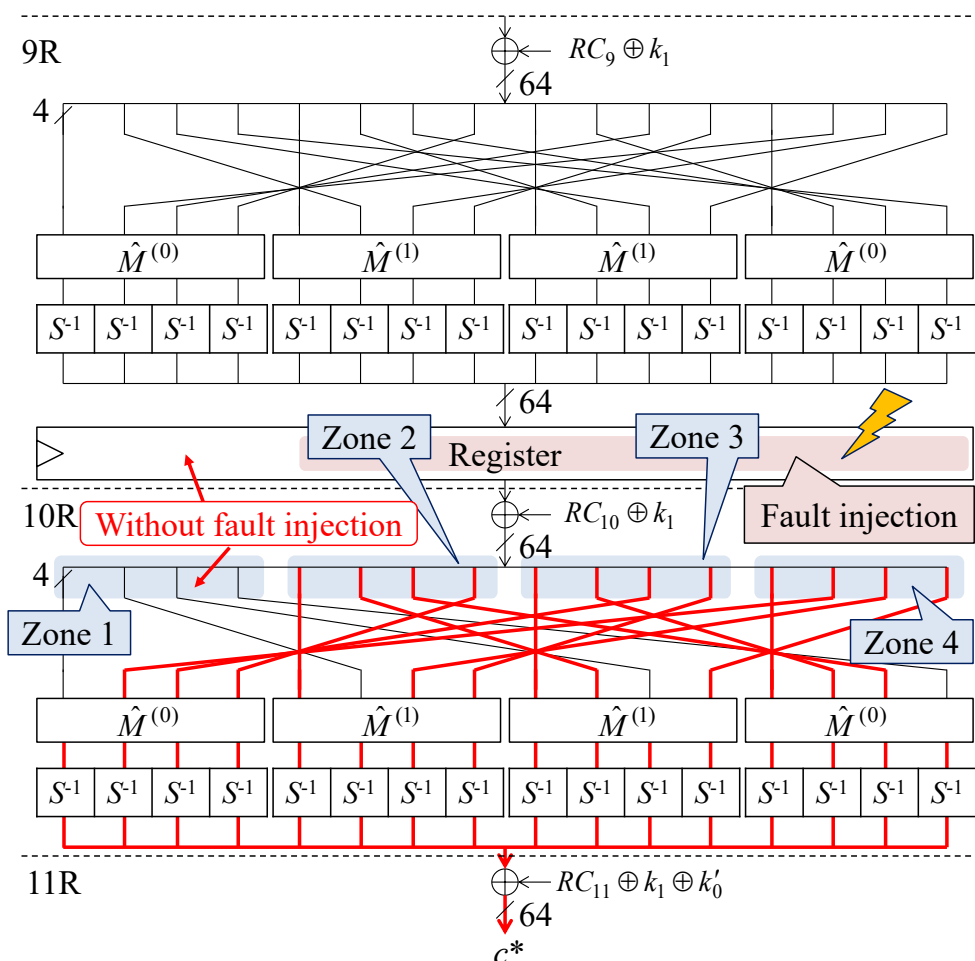


図 12.1 フォールトモデル I

Fig. 12.1 Fault model I

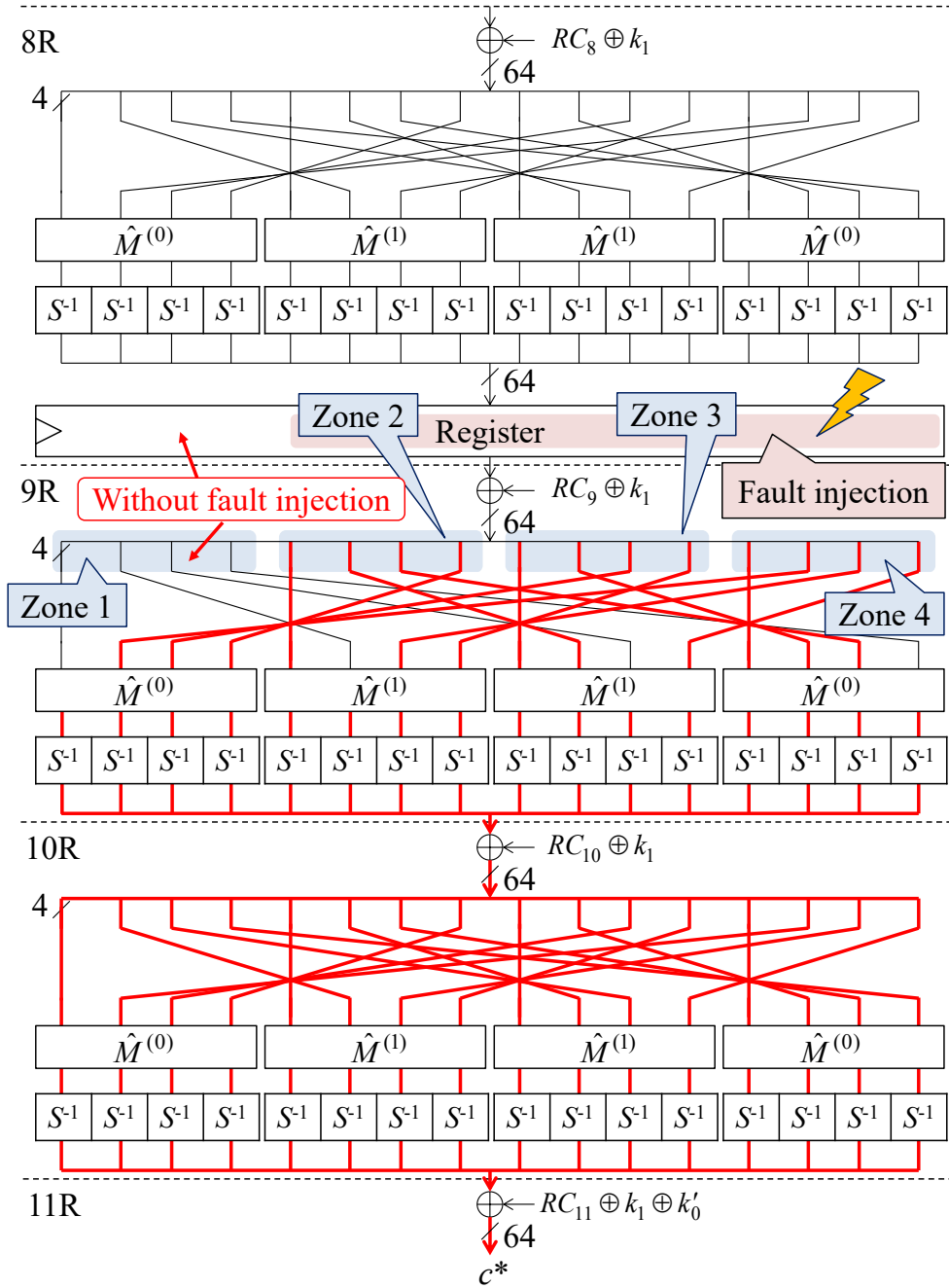


図 12.2 フォールトモデル II

Fig. 12.2 Fault model II

12.3 提案するフォールト解析手法

12.3.1 ベースとなる解析手法の概要

提案するフォールト解析手法では、通常の暗号文とフォールト入り暗号文の任意のゾーンを構成するビット列（以後、注目値と呼ぶ）のハミング距離を利用する解析モデルを導入する。まず、注目値のビット位置とその性質について、図 12.3 を用いて説明する。図 12.3 はフォールトモデル I の場合で、提案するフォールト解析手法では、まず、通常の暗号文とフォールト入り暗号文からそれぞれ 10R の注目値（フォールトモデル I の注目値）を計算する。図 12.3 に示す例は、フォールト入り暗号文での計算例であり、10R の計算における SR^{-1} 処理の入力値（64bit）の上位 16bit、すなわち、ゾーン 1 を 4bit ずつに分割して注目値とした場合である。この注目値をそれぞれ α_1^* , α_2^* , α_3^* , α_4^* （フォールト入り暗号文から計算した注目値）とする。また、通常の暗号文においても同様に、注目値 α_1 , α_2 , α_3 , α_4 を求める。このとき、注目値 α_1^* , α_2^* , α_3^* , α_4^* には、図 12.1 のフォールトモデル I に示すように、フォールトは混入しない。したがって、注目値 α_1 , α_2 , α_3 , α_4 と α_1^* , α_2^* , α_3^* , α_4^* は同じ値となる（ $\alpha_1^* = \alpha_1$, $\alpha_2^* = \alpha_2$, $\alpha_3^* = \alpha_3$, $\alpha_4^* = \alpha_4$ ）。

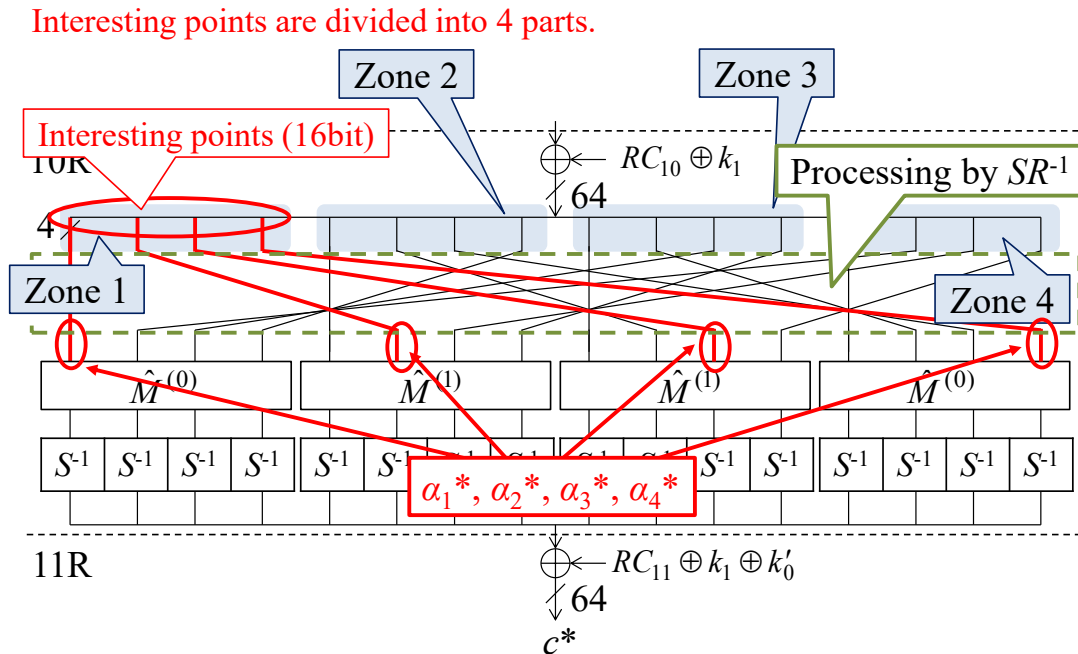


図 12.3 解析モデルにおける注目値

Fig. 12.3 Interesting points for the analysis model

次に、注目値の計算方法について図 12.4 を用いて説明する。ここでは、注目値 α_1 と α_1^* を計算する場合について説明する。提案するフォールト解析手法では、図 12.4 の(i)に示すように、既知である暗号文 c と部分鍵 $k_1 \oplus k'_0$ の予測値を利用した逆算処理（詳細は 12.3.2 項で説明する）によって 4bit の注目値 α_1 を計算する。このとき、部分鍵 $k_1 \oplus k'_0$ を K' とおく。図 12.4 の(i)に示す例では、部分鍵 $K'_{(1-16)}$ の予測値は 16bit であるため、0 から 65535 の 65536 通りの鍵候補を試す。同様に、既知のフォールト入り暗号文 c^* と部分鍵 $K'_{(1-16)}$ の予測値を利用した逆算処理によって 4bit の注目値 α_1^* を計算する。そして、図 12.4 の(i)に示すように、注目値 α_1 と α_1^* のハミング距離を計算する。

次に、図 12.4 の(ii)に示すように、図 12.4 の(i)に示す処理を、通常の暗号文とフォールト入り暗号文のペアの数 n だけ繰り返し、ハミング距離の平均値を求める。

このとき、正しい部分鍵の予測値に基づいて注目値 α_1 , α_1^* を計算した場合、注目値 α_1 と α_1^* のハミング距離の平均値は 0 となる。一方で、間違った部分鍵の予測値に基づいて注目値 α_1 , α_1^* を計算した場合、注目値 α_1 と α_1^* のハミング距離の平均値は 0 よりも大きい値となる。

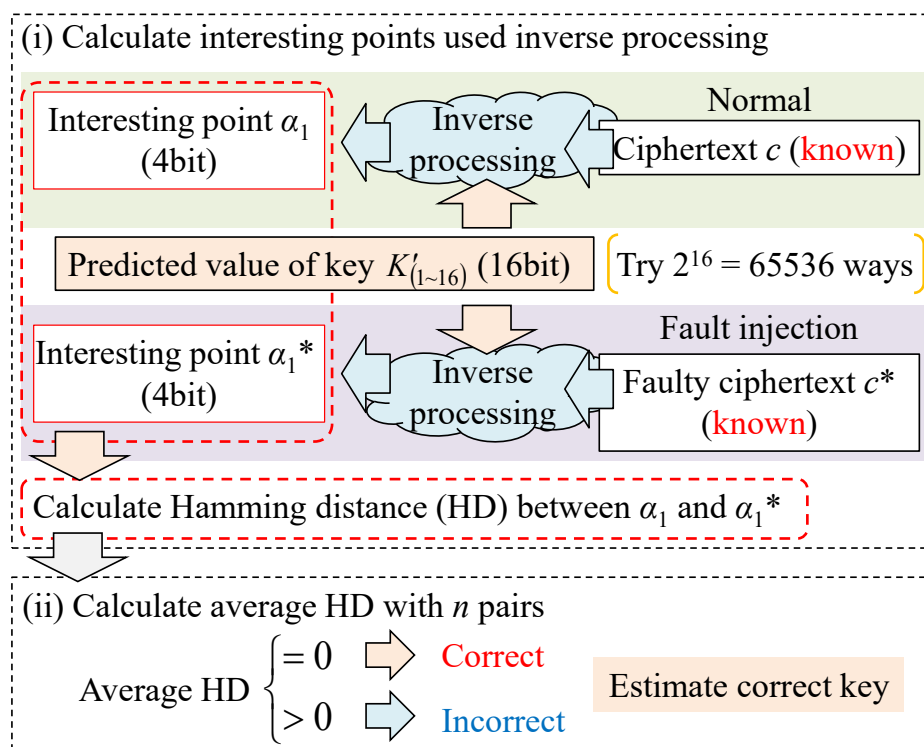


図 12.4 注目値の計算手順

Fig. 12.4 Calculation procedure for interesting points

したがって、提案するフォールト解析手法では図 12.4 の(ii)に示すように、通常の暗号文 c から計算した注目値 α_1 と、フォールト入り暗号文 c^* から計算した注目値 α_1^* のハミング距離の平均値が 0 となる部分鍵の予測値を正解鍵と推定する。また、他の注目値 α_2 と α_2^* 、 α_3 と α_3^* 、 α_4 と α_4^* についても同様に計算し、残りの部分鍵 $K'_{(17\sim32)}$ 、 $K'_{(33\sim48)}$ 、 $K'_{(49\sim64)}$ 、すなわち 48bit 分の部分鍵 $K'_{(17\sim64)}$ を推定する。

12.3.2 解析手法における逆算処理

図 12.4 の(i)に示す処理で行う逆算処理の詳細について説明する。逆算処理は、逆算処理 I と逆算処理 II の 2 段階で構成する。まず、逆算処理 I について説明する。逆算処理 I では、図 12.5 に示すように 10R における $\hat{M}^{(0)}$ 、 $\hat{M}^{(1)}$ の行列演算 (A.4 節を参照) の 16bit の入力値をそれぞれ計算する。この入力値をそれぞれ p_1^* 、 p_2^* 、 p_3^* 、 p_4^* とする。このとき、図 12.5 に示すように p_1 、 p_2 、 p_3 、 p_4 と注目値 α_1^* 、 α_2^* 、 α_3^* 、 α_4^* には式(12.1)に示す関係が成り立つ。同様に、通常の暗号文における、10R の $\hat{M}^{(0)}$ 、 $\hat{M}^{(1)}$ の行列演算の 16bit の入力値 p_1 、 p_2 、 p_3 、 p_4 と、注目値 α_1 、 α_2 、 α_3 、 α_4 には式(12.2)に示す関係が成り立つ。

$$\begin{aligned}\alpha_1^* &= p_{1(1\sim4)}^* \\ \alpha_2^* &= p_{2(5\sim8)}^* \\ \alpha_3^* &= p_{3(9\sim12)}^* \\ \alpha_4^* &= p_{4(13\sim16)}^*\end{aligned}\tag{12.1}$$

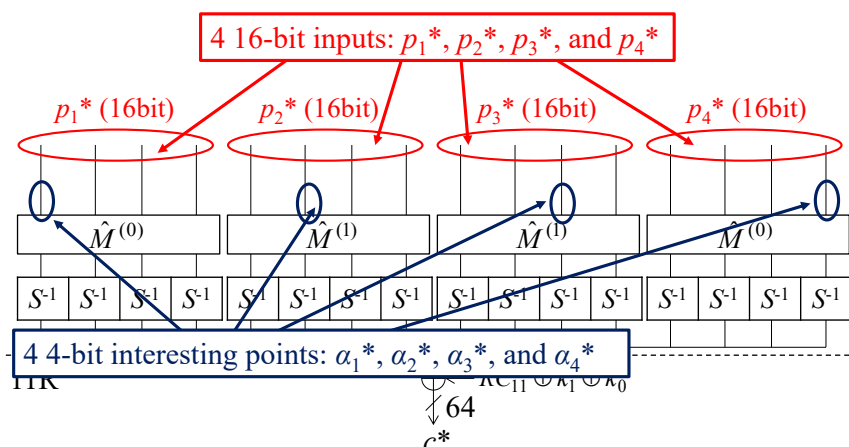


図 12.5 逆算処理で求める値

Fig. 12.5 Values calculated by the inverse processing

$$\begin{aligned}
\alpha_1 &= p_{1(1\sim4)} \\
\alpha_2 &= p_{2(5\sim8)} \\
\alpha_3 &= p_{3(9\sim12)} \\
\alpha_4 &= p_{4(13\sim16)}
\end{aligned} \tag{12.2}$$

逆算処理 I における入力値 p_1^* の計算方法を図 12.6 に示す。図 12.6 に示すように、逆算処理 I での S^{-1} の処理は S^{-1} の逆関数、すなわち $(S^{-1})^{-1} = S$ から、 S を利用する。また、 $\hat{M}^{(0)}$ の計算は、式(A.19)に示す関係から、 $\hat{M}^{(0)}$ をそのまま適用する。したがって、 $\hat{M}^{(0)}$ の行列演算の入力値 p_1^* は式(12.3)で計算できる。

$$p_1^* = S((c_{(1\sim16)}^* \oplus RC_{11(1\sim16)} \oplus K'_{(1\sim16)}) \cdot \hat{M}^{(0)}) \tag{12.3}$$

また、通常の暗号文においても同様にして、 p_1 を計算することができる。この計算式を式(12.4)に示す。

$$p_1 = S((c_{(1\sim16)} \oplus RC_{11(1\sim16)} \oplus K'_{(1\sim16)}) \cdot \hat{M}^{(0)}) \tag{12.4}$$

次に、逆算処理 II について説明する。逆算処理 II では、逆算処理 I で計算した p_1^* を利用して、注目値 α_1^* を算出する。具体的には、式(12.1)に示す関係を用いて、注目値 α_1^* を求める。また、通常の暗号文においても同様に、逆算処理 I で計算した p_1 から、式(12.2)に示す関係を用いて、注目値 α_1 を計算する。

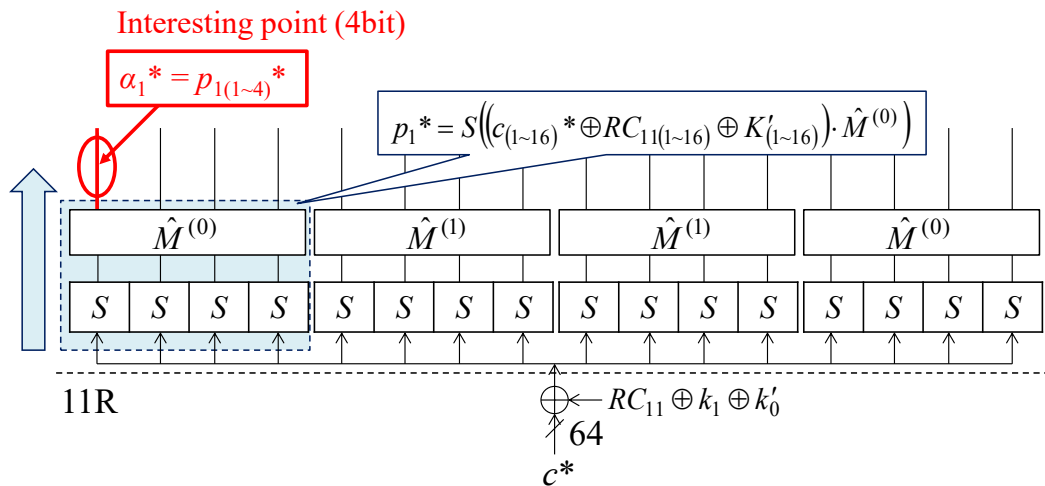


図 12.6 逆算処理の計算例

Fig. 12.6 Calculation example of the inverse processing

そして、 p_2^* , p_3^* , p_4^* と p_2 , p_3 , p_4 についても、逆算処理Iによって計算する。また、注目値 α_2^* , α_3^* , α_4^* と α_2 , α_3 , α_4 は逆算処理IIによって計算する。

12.3.3 解析手法における鍵の推定

ここでは、12.3.2 項の逆算処理によって導出した注目値を利用して部分鍵を推定する。式(12.3)と式(12.4)において、通常の暗号文 $c_{(1\sim16)}$ ，フォールト入り暗号文 $c_{(1\sim16)}^*$ ，ラウンド定数 $RC_{11(1\sim16)}$ は既知の値であるが、部分鍵 $K'_{(1\sim16)}$ は未知の値である。ここで、16bit の部分鍵 $K'_{(1\sim16)}$ に予測値として pK ($0 \leq pK \leq 65535$) を与え、式(12.3)と式(12.4)を計算し、 p_1^* , p_1 を求める。この結果を pX^* , pX とすると、式(12.5)が得られる。

$$\begin{aligned} pX^* &= S((c_{(1\sim16)}^* \oplus RC_{11(1\sim16)} \oplus pK) \cdot \hat{M}^{(0)}) \\ pX &= S((c_{(1\sim16)} \oplus RC_{11(1\sim16)} \oplus pK) \cdot \hat{M}^{(0)}) \end{aligned} \quad (12.5)$$

このとき、 pX^* , pX はそれぞれ、 p_1^* , p_1 と対応しているので、式(12.1)と式(12.2)の関係を利用して、注目値を求める。この注目値をそれぞれ $pX_{(1\sim4)}^*$, $pX_{(1\sim4)}$ とすると、 $pX_{(1\sim4)}^*$, $pX_{(1\sim4)}$ と α_1^* , α_1 には式(12.6)に示す関係が成り立つ。

$$\alpha_1 \begin{cases} = pX_{(1\sim4)} & (pK = K'_{(1\sim16)}) \\ \neq pX_{(1\sim4)} & (pK \neq K'_{(1\sim16)}) \end{cases}, \quad \alpha_1^* \begin{cases} = pX_{(1\sim4)}^* & (pK = K'_{(1\sim16)}) \\ \neq pX_{(1\sim4)}^* & (pK \neq K'_{(1\sim16)}) \end{cases} \quad (12.6)$$

ここで、 a と b のハミング距離を計算する関数を $HD(a, b)$ として、 α_1 と α_1^* , $pX_{(1\sim4)}$ と $pX_{(1\sim4)}^*$ のハミング距離をそれぞれ h , hpX とすると式(12.7)が成り立つ。

$$h \begin{cases} = hpX & (pK = K'_{(1\sim16)}) \\ \neq hpX & (pK \neq K'_{(1\sim16)}) \end{cases} \quad (12.7)$$

このとき、部分鍵の予測値 pK が正解値でない場合、すなわち、 $pK \neq K'_{(1\sim16)}$ の場合、注目値 $pX_{(1\sim4)}$ と $pX_{(1\sim4)}^*$ の間には相関は無く、そのハミング距離の平均値は 0 よりも大きい値となる。

一方で、部分鍵の予測値 pK が正解値の場合、すなわち、 $pK = K'_{(1\sim16)}$ の場合、注目値 $pX_{(1\sim4)}$ と $pX_{(1\sim4)}^*$ は等しくなるため、そのハミング距離の平均値は 0 となる。

したがって、 N 組の通常の暗号文 (c) とフォールト入り暗号文 (c^*) のペア (c_1 , c_1^*), ..., (c_N , c_N^*) を準備し、 N 個のハミング距離 hpX_1 , ..., hpX_N を計算し、ハミング距離の

平均値 $\overline{HD}$ を式(12.8)によって算出する。

$$\overline{HD} = \frac{1}{N} \sum_{n=1}^N hpX_n \begin{cases} = 0 & (pK = K'_{(1\sim 16)}) \\ > 0 & (pK \neq K'_{(1\sim 16)}) \end{cases} \quad (12.8)$$

以上のように、十分な数 N のとき、あるハミング距離の平均値が 0 となる部分鍵の予測値 pK を正解鍵と推定する。この操作によって部分鍵 $K'_{(1\sim 16)}$ 、すなわち、64bit の部分鍵 K' のうち 16bit の部分鍵を推定することができる。

提案するフォールト解析手法では、以上の操作を残りの注目値 α_2^* 、 α_3^* 、 α_4^* と α_2 、 α_3 、 α_4 の計算にも適用する。したがって、残りの 48bit の部分鍵を推定することが可能であるため、64bit の部分鍵 K' を全て推定することができる。

12.3.4 階層的な解析モデル

12.3.1 項から 12.3.3 項の統計処理を用いた解析モデルで推定した部分鍵 K' は、 $k_1 \oplus k'_0$ であるため、 K' から秘密鍵 k を求めることはできない。そこで提案するフォールト解析手法では、部分鍵 $k_1 \oplus k'_0$ を用いて階層的に解析手法を適用すること（以後、階層的な解析モデルと呼ぶ）で、秘密鍵 k を推定する。

階層的な解析モデルの概要を図 12.7 に示す。図 12.7 の(i)はフォールトモデルⅠに対して、12.3.1 項から 12.3.4 項で説明した解析モデルを適用した場合を示している。図 12.7 の(i)に示すように、この処理では、部分鍵 $k_1 \oplus k'_0$ を推定する。

次に、図 12.7 の(ii)はフォールトモデルⅡに対して、解析モデルを適用した場合を示したものである。図 12.7 の(ii)に示す処理では、図 12.7 の(i)に示す処理で推定した部分鍵 $k_1 \oplus k'_0$ を利用することで、部分鍵 k_1 を推定する。ここでは、フォールト入り暗号文を用いた場合について説明する。具体的には、図 12.2 に示すフォールトモデルⅡに対して解析モデルを適用することで、図 12.8 に示すように 9R の注目値（フォールトモデルⅡの注目値）を算出する。これをそれぞれ β_1^* 、 β_2^* 、 β_3^* 、 β_4^* とする。

また、このとき、注目値 β_1^* 、 β_2^* 、 β_3^* 、 β_4^* は図 12.8 に示すフォールト入り暗号中間値 X' を用いた逆算処理（〈3・2・2〉の逆算処理）によって求めることができる。ここで、フォールト入り暗号中間値 X^* は、図 12.7 の(i)に示す処理で推定した部分鍵 $k_1 \oplus k'_0$ とフォールト入り暗号文 c^* を用いて、式(12.9)から求めることができる。

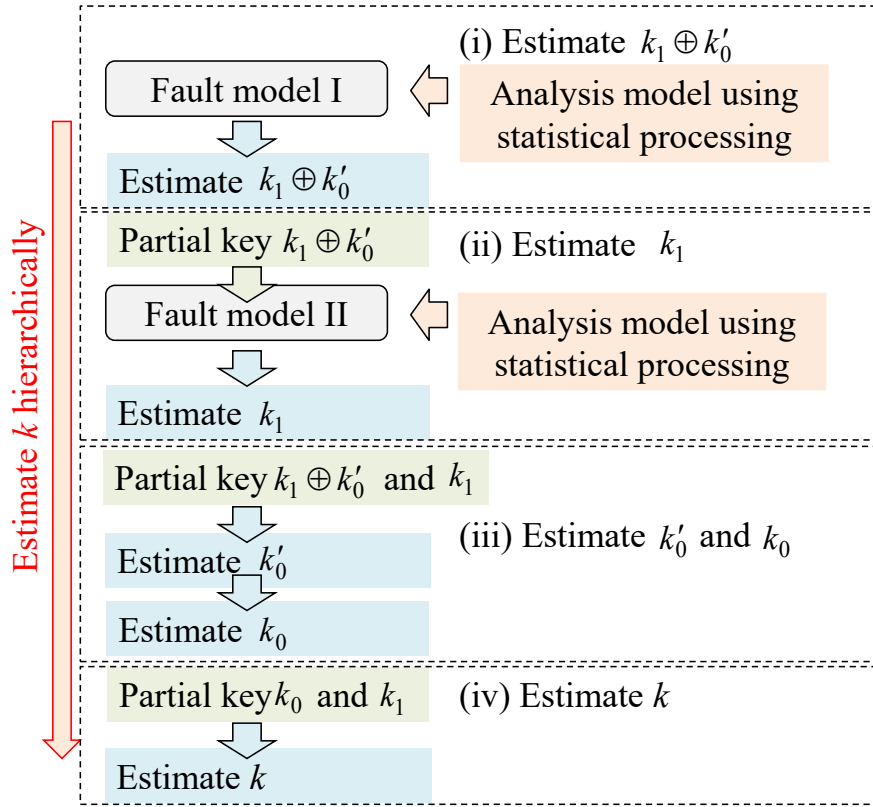


図 12.7 階層的な解析モデルの概要

Fig. 12.7 Outline of the hierarchical analysis model

$$X^* = SR(S(c^* \oplus RC_{11} \oplus k_1 \oplus k'_0) \cdot M') \quad (12.9)$$

具体的には、注目値 β_1^* は式(12.1)と式(12.3)を用いて計算する。このとき、 α_1^* には β_1^* を、 $c_{(1 \sim 16)}^*$ には $X_{(1 \sim 16)}^*$ を、 $RC_{11(1 \sim 16)}$ には $RC_{10(1 \sim 16)}$ を、 $K'_{(1 \sim 16)}$ には $k_{1(1 \sim 16)}$ を代入して計算する。また、通常の暗号文 c から算出する注目値 β_1 についても同様に計算を行う。そして、注目値 β_1 と β_1^* のハミング距離を計算し、部分鍵 $k_{1(1 \sim 16)}$ を推定する。残りの注目値 $\beta_2, \beta_3, \beta_4$ と $\beta_2^*, \beta_3^*, \beta_4^*$ についても同様の計算によって求め、部分鍵 $k_{1(17 \sim 32)}, k_{1(33 \sim 48)}, k_{1(49 \sim 64)}$ を推定する。以上より、64bit の部分鍵 k_1 を推定することが可能である。

そして、図 12.7 の(iii)に示す処理では、図 12.7 の(i)(ii)で示す処理で推定した部分鍵 $k_1 \oplus k'_0$ と k_1 を利用することで、部分鍵 k'_0 を推定する。この計算式を式(12.10)に示す。

$$k'_0 = (k_1 \oplus k'_0) \oplus k_1 \quad (12.10)$$

A 16-bit interesting point are divided into 4 4-bit values.

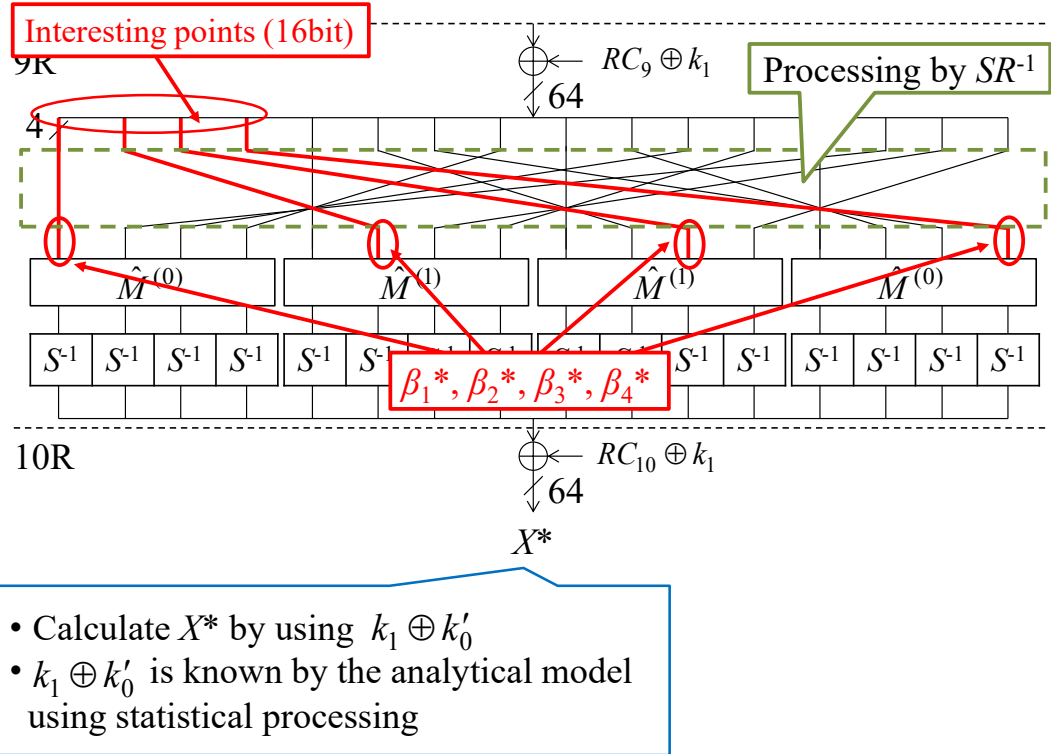


図 12.8 階層的な解析モデルにおける注目値

Fig. 12.8 Interesting points for the hierarchical analysis model

更に、式(12.10)で導出した部分鍵 k'_0 を利用して部分鍵 k_0 を推定する。部分鍵 k_0 の推定方法を図 12.9 に示す。このとき、 k'_0 は式(A.27)から、図 12.9 の(i)に示す計算で求まる。また、 k'_0 が既知であることから、図 12.9 の(ii)に示すように $k_{0(1)}, \dots, k_{0(62)}, k_{0(64)}, k_{0(63)} \oplus k_{0(1)}$ は既知の値、 $k_{0(63)}$ は未知の値である。ここで、 $k_{0(63)}$ は $k_{0(63)} \oplus k_{0(1)}$ と $k_{0(1)}$ が既知であることから、式(12.11)で計算できる。

$$k_{0(63)} = (k_{0(63)} \oplus k_{0(1)}) \oplus k_{0(1)} \quad (12.11)$$

最後に、図 12.7 の(iv)に示すように、秘密鍵を推定する。具体的には図 12.7 の(iv)に示すように、図 12.7 の(ii)(iii)に示す処理で推定した部分鍵 k_0, k_1 を利用して、秘密鍵を推定する。

以上より、128bit の秘密鍵 k を全て導出することが可能である。

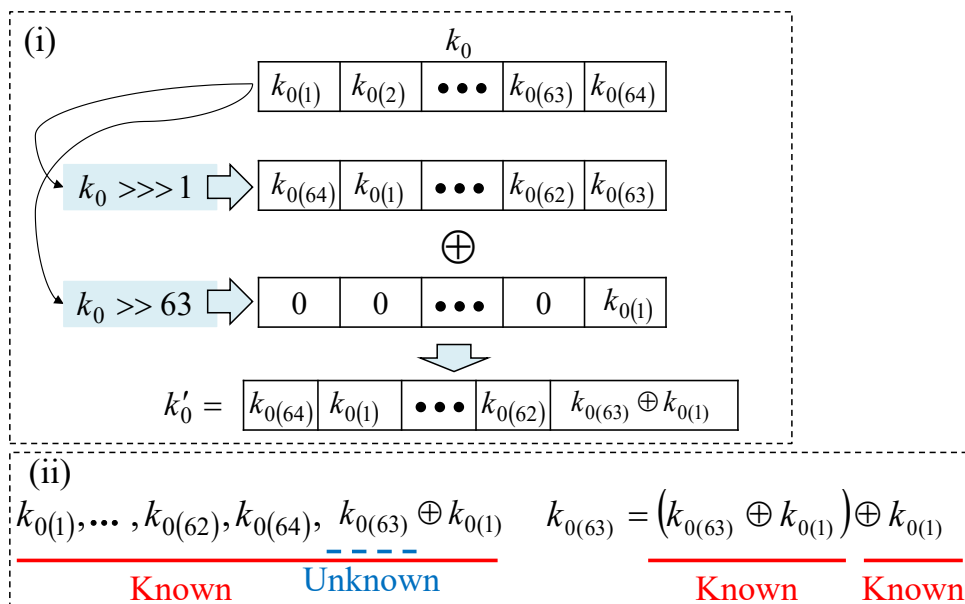


図 12.9 鍵 k_0 の解析方法

Fig. 12.9 Analytical method for the key k_0

12.3.5 枝刈りアルゴリズム

12.3.1 項から 12.3.3 項で説明した解析モデルは、部分鍵を 16bit ずつ計算しているため、65536 通りの鍵候補を暗号文ペア（通常の暗号文とフォールト入り暗号文の組）の数だけ計算する必要がある。耐タンパ性を効率的に評価するためには、解析に必要な処理時間を短縮することは重要である。そこで提案するフォールト解析手法では、処理時間の短縮や計算量の削減を実現するために、部分鍵の推定において、ハミング距離が 0 以外の鍵候補を切り捨てる枝刈りアルゴリズムを導入する。

枝刈りアルゴリズムについて、注目値 α_1 と α_1^* を計算する場合を例に説明する。図 12.10 に枝刈りアルゴリズムの処理を示す。ここでは、鍵候補リストを作成し、鍵候補リストに登録された鍵候補に対して解析モデルを適用する。まず、図 12.10 に示すように、1 組目の暗号文ペアを用いた計算では、65536 通りの鍵候補が存在する鍵候補リストから 1 つずつ鍵候補を試していく。そして、注目値 α_1 と α_1^* のハミング距離を導出し、このハミング距離が 0 の場合、この鍵候補を鍵候補リスト A_1 に加える。

一方で、ハミング距離が 0 でない場合、試行した鍵候補は正解鍵ではないため、鍵候補リスト A_1 に鍵候補を加えない。この処理を繰り返すことで、試行する鍵候補の数を減らす。

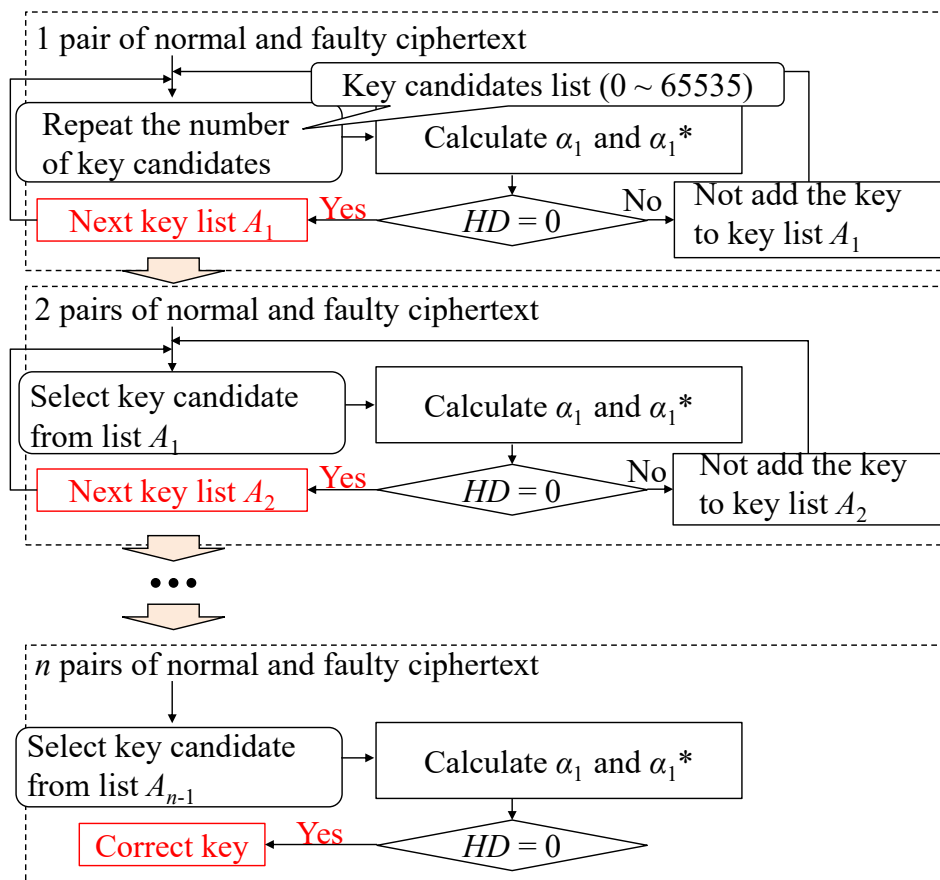


図 12.10 枝刈りアルゴリズム

Fig. 12.10 Pruning algorithm

そして、この処理を十分な数である N 回繰り返すことで、鍵候補を 1 つに絞り込むことが可能である。したがって、この処理で最後まで残った鍵候補を正解鍵と推定することができる。

12.4 評価実験

12.4.1 実験結果

ここでは、各フォールトモデルに対して、ランダムにフォールトを混入させる。具体的には、乱数によって決定した位置のビット値を反転させる。

まず、最初の実験では、フォールトの混入個数は 1 ビットとする。そして、乱数で通常の暗号文とフォールト入り暗号文のペアを 70 組生成する。提案するフォールト解析手法を

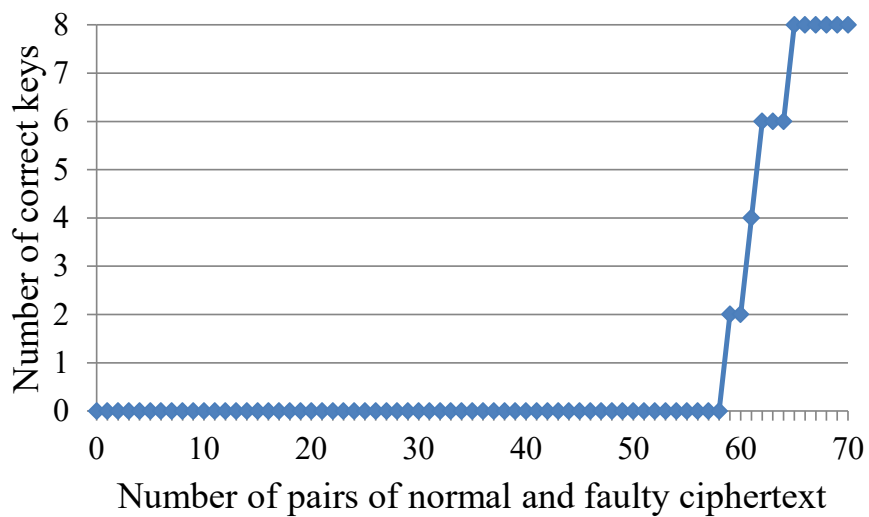


図 12.11 実験結果

Fig. 12.11 Simulation result

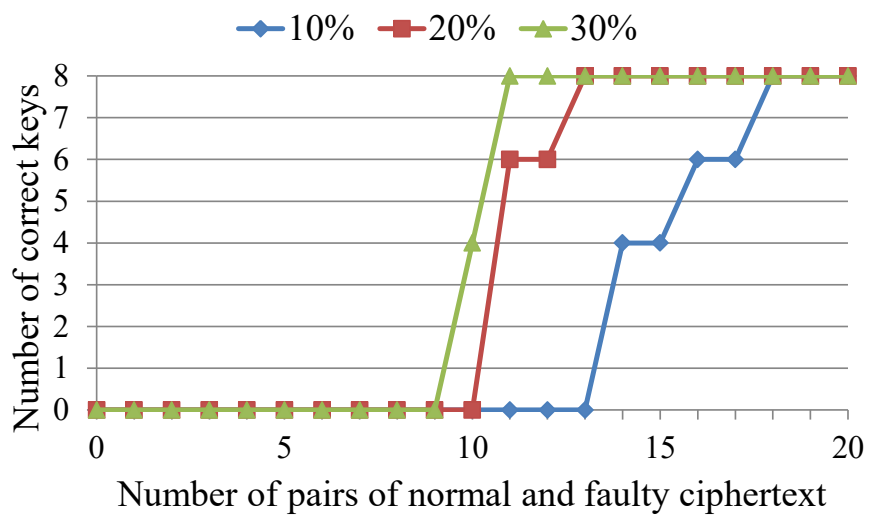


図 12.12 フォールト混入数の違いによる解析結果

Fig. 12.12 Simulation results with different number of faults

用いて、秘密鍵 k の解析を行った結果を図 12.11 に示す。図の縦軸（正解部分鍵数）は秘密鍵 128bit のうち、正解した部分鍵の数（1 つの部分鍵は 16bit）を示したものである。また、横軸は解析に使用した暗号文ペア数である。図 12.11 に示すように、暗号文ペア数が 65 組のとき、全ての秘密鍵の解析に成功していることから、提案するフォールト解析手法の有効性を確認できる。

次に、フォールト混入個数が解析結果に与える影響について検証した。ここでは、フォールト混入個数をそれぞれ、対象とする暗号中間値の 10%、20%、30% のビット数を変化させた場合について実験を行った。この結果を図 12.12 に示す。図 12.12 に示すように、フォールト混入個数が 10% のとき、暗号文ペア数が 18 組で全ての秘密鍵の推定に成功した。一方で、フォールト混入個数が 30% では、暗号文ペア数が 10 組で全ての秘密鍵の推定に成功した。したがって、フォールト混入個数に関係なく、解析を行うことができることが分かる。

以上より、提案するフォールト解析手法を用いることで、PRINCE の耐タンパ性を検証することができる。

12.4.2 ハミング距離の比較結果

各鍵候補別の平均ハミング距離について比較した。ここでは、平均ハミング距離が小さいものを顕著にするために、ハミング距離の特徴量を計算する。ハミング距離の特徴量は、平均ハミング距離とハミング距離の中央値との差の 2 乗を計算したものである。ここで、中央値は注目値 4bit の半分の 2 としている。平均ハミング距離が小さくなるほど、この特徴量は大きくなる。また、平均ハミング距離が中央値に近づくほど、ハミング距離の特徴量は小さくなる。

実験結果を図 12.13 に示す。図の横軸は鍵候補を、縦軸はハミング距離の特徴量を、赤色は正解鍵を、青色は不正解鍵をそれぞれ示している。図 12.13 に示すように、正解鍵においてハミング距離の特徴量が大きくなっている。このことは他の鍵候補と比較して、ハミング距離が小さくなっていることを表している。一方で、不正解鍵ではハミング距離の特徴量は小さくなっている。このことは平均ハミング距離が中央値 2 に近づいていることを示している。

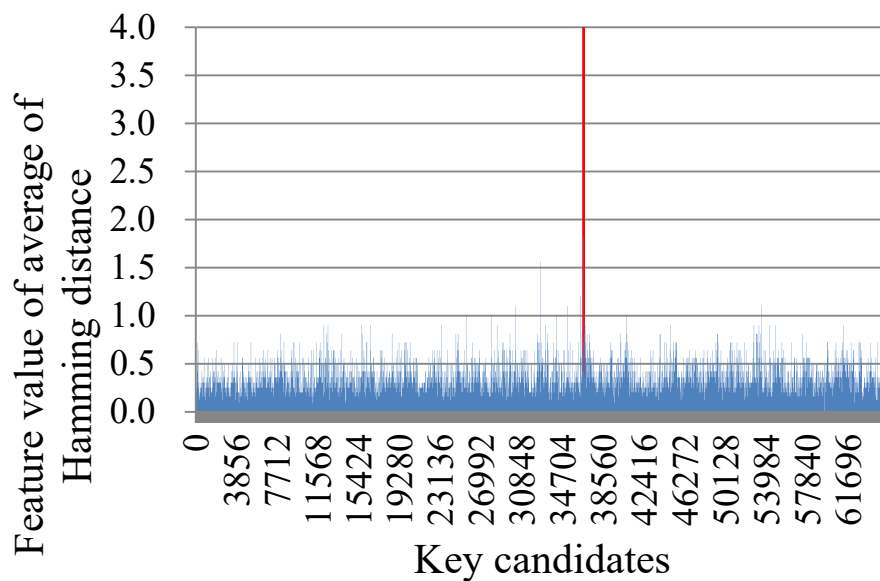


図 12.13 各鍵候補でのハミング距離の特徴量

Fig. 12.13 Feature value of average of Hamming distance for each key candidate

12.4.3 処理時間の比較

枝刈りアルゴリズムの有効性を検証するために、処理時間を比較した。実験では、枝刈りアルゴリズムを使用する場合と、使用しない場合を比較した。具体的には、暗号文ペア数を 20 組用意し、1 つの部分鍵 16bit の予測における処理時間と計算量を比較した。比較結果を表 12.1 に示す。

まず、処理時間については、枝刈りアルゴリズムを使用した場合は 28 [sec]となっている。そのため、残りの 7 つの部分鍵を含んだ全ての部分鍵の解析には、 $28 \times 8 = 224$ [sec]の処理が必要となる。一方で、枝刈りアルゴリズムを使用しない場合、1 つの部分鍵における処理時間は 471 [sec]となっている。そのため、全ての部分鍵の解析には、 $471 \times 8 = 3768$ [sec]の処理が必要となる。したがって、枝刈りアルゴリズムを使用することで、約 17 倍の高速化を実現し、耐タンパ性を効率的に評価することが可能である。

また、計算量については、枝刈りアルゴリズムを使用しない場合は、65536 通りの鍵候補を暗号文ペア 20 組分の計算を行う必要がある。一方で、枝刈りアルゴリズムを使用した場合、その計算量は、表 12.1 に示すように 69904 回となっており、大幅に計算量を削減できる。

表 12.1 処理時間の比較結果

Table 12.1 Comparison result of processing time

	Processing time	Computational complexity
Pruning algorithm	471 [sec]	65536×20
Without pruning algorithm	28 [sec]	69904

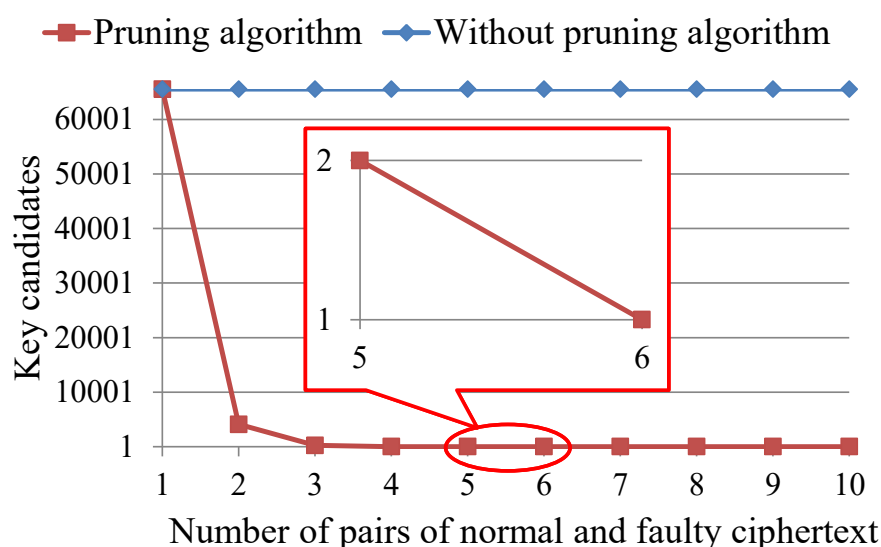


図 12.14 枝刈りアルゴリズムを使用したときの鍵候補数の推移

Fig. 12.14 Transition of the number of key candidates using pruning algorithm

更に、枝刈りアルゴリズムによる部分鍵の鍵候補数の推移について検証した結果を図 12.14 に示す。図 12.14 に示すように、1 組目の暗号文ペアでは、鍵候補数は 65536 通りであるが、暗号文ペア数が増えていくにしたがい、鍵候補数が少なくなっていく。そして、暗号文ペア数が 6 組のとき、鍵候補を 1 つに絞り込めていることが分かる。

以上より、枝刈りアルゴリズムを導入することで、計算量・処理時間を大幅に削減することができ、枝刈りアルゴリズムが有効であることを確認した。

12.5 結言

本章では、PRINCE に対するフォールト解析手法を提案し、シミュレーションによる評

価実験によってその有効性を示した。提案したフォールト解析では、統計的な解析モデルを導入することで、フォールトの混入する位置を精密に制御できない場合での解析を可能にした。また、階層的な解析モデルを導入することで、全ての秘密鍵が導出できることを示した。更に、枝刈りアルゴリズムを導入することで解析に必要な計算量を削減し、解析時間の約 17 倍の高速化を実現した。

以上より、提案したフォールト解析手法を用いることで、PRINCE の耐タンパ性を評価できることを示した。

参考文献

- [1] 野崎佑典, 野原康平, 松久僚真, 旭 健作, 吉川雅弥, “PRINCE に対する統計的フォールト解析とその評価,” 電気学会論文誌 C, vol. 135, no. 12, pp. 1442–1452, Dec. 2015.
- [2] J. Borghoff, A. Canteaut, T. Güneysu, E. B. Kavum, M. Knežević, L. R. Knudsen, G. Leander, V. Nikov, C. Paar, C. Rechberger, P. Rombouts, S. S. Thomsen, and T. Yalçin, “PRINCE - A Low-latency Block Cipher for Pervasive Computing Applications,” Proc. ASIACRYPT 2012, LNCS vol. 7658, pp. 208–225, Springer, Dec. 2012.
- [3] CRYPTREC, “CRYPTREC Cryptographic Technology Guideline (Lightweight Cryptography),” Mar. 2017.
<http://www.cryptrec.go.jp/report/cryptrec-gl-0001-2016-e.pdf>
- [4] L. Song and L. Hu, “Differential Fault Attack on the PRINCE Block Cipher,” Proc. Second Int. Workshop on Lightweight Cryptography for Security & Privacy (LightSec 2013), LNCS vol. 8162, pp. 43–54, Springer-Verlag, May 2013.
- [5] P. Jovanovic, M. Kreuzer, and I. Polian, “Multi-Stage Fault Attacks on Block Ciphers,” IACR Cryptology ePrint Archive: Report 2013/778, pp. 1–6, Nov. 2013.
- [6] S. P. Skorobogatov and R. J. Anderson, “Optical Fault Induction Attacks,” Proc. 4th Int. Workshop on Cryptographic Hardware and Embedded Systems (CHES 2002), LNCS 2523, pp. 2–12, Springer-Verlag, Aug. 2003.
- [7] 藤原 充, 諸藤 力, 成吉雄一郎, 塚本研一, 山梨 晃, 川村信一, “レーザーアタ

ックの優位的評価手法について,” 2011 年暗号と情報セキュリティシンポジウム講演
論文集, 3D3-6, pp. 1–7, Jan. 2011.

第 13 章

結論

本論文では、軽量暗号の安全性評価に必要なサイドチャネル解析手法を開発することを目的として、軽量暗号の 5 大要素（商用利用・小回路規模実装・低消費電力実装・低遅延実装・認証付暗号）の代表的なアルゴリズム TWINE, SIMECK, Midori, PRINCE, Minalpher, TWINE-OTR に対するサイドチャネル解析手法を提案し、評価実験によってその有効性を示した。以下に得られた研究成果を要約する。

第 2 章では、まず、基本的な暗号技術の分類や実装について述べた。そして、本論文で対象とした軽量暗号の 5 大要素の代表的なアルゴリズム TWINE, SIMECK, Midori, PRINCE, Minalpher, TWINE-OTR について概説した。

第 3 章では、暗号回路に対する物理解析について、特に、サイドチャネル解析を中心に概説した。

第 4 章から第 9 章では、軽量暗号に対する電力・電磁波解析手法を提案した。

第 4 章では、TWINE を対象とした電力解析手法を提案し、FPGA を用いた評価実験によってその有効性を示した。提案した TWINE に対する電力解析手法では、秘密鍵を全て推定するために、相関電力解析をベースとした解析を複数ラウンドに対して適用した。また、複数ラウンドに対する電力解析では、他のラウンドで解析済の部分鍵を組み込む処理を導入することで、ノイズ成分による影響を減らし、解析効率を向上させた。更に、時間軸上で生じる消費電力値の位相ずれ誤差の影響を軽減するために、周波数領域での解析を導入し、解析効率を向上させた。

第 5 章では、SIMECK を対象とした電磁波解析手法を提案し、FPGA を用いた評価実験によってその有効性を示した。提案した SIMECK に対する電磁波解析手法では、SIMECK

の構造に着目した複数のラウンドの情報を利用する解析を導入することで、通常よりも多くの情報を利用し、解析効率を向上させた。また、解析において既知となる情報を組み込む処理を導入することで、ノイズ成分の影響を軽減し、解析効率を向上させた。

第 6 章では、Midori を対象とした電力解析手法を提案し、FPGA を用いた評価実験によってその有効性を示した。提案した Midori に対する電力解析手法では、Midori の 16R と 2R を対象とした階層的な電力解析手法を導入することで、全ての秘密鍵を推定した。また、Midori のアルゴリズムに着目した、解析の効率化手法を導入することで、その計算量を約 7.0×10^{13} 通りから 16,448 通りまで削減することに成功した。

第 7 章では、Minalpher を対象とした電力解析手法を提案し、FPGA を用いた評価実験によってその有効性を示した。提案した Minalpher に対する電力解析手法では、オフセット値による排他的論理和演算を外すために、オフセット値生成部分と暗号文生成部分を対象とした 2 段の階層的な電力解析手法を導入することで、全ての秘密鍵の推定に成功した。また、周波数領域での解析手法を展開することで、時間軸上で生じる位相ずれ誤差の影響を軽減し、解析効率を向上させた。

第 8 章では、TWINE-OTR を対象とした電力解析手法を提案し、FPGA を用いた評価実験によってその有効性を示した。提案した TWINE-OTR に対する電力解析手法では、入力メッセージによる排他的論理和演算を外すために、暗号文生成部分を対象とした 2 段階の電力解析提案手法を導入し、鍵の推定を成功させた。また、TWINE-OTR のアルゴリズムに着目した複数の暗号文生成部分を対象とした解析を導入することで、解析効率を向上させた。

第 9 章から第 12 章では、軽量暗号に対するフォールト解析手法を提案した。

第 9 章では、TWINE を対象としたフォールト解析手法を提案し、FPGA を用いた評価実験によってその有効性を示した。提案した TWINE に対するフォールト解析手法では、クロックグリッチに基づくフォールト混入を想定した、統計的な解析手法を導入することで、複数のフォールトがランダムな位置に混入した場合でも解析を可能にした。

第 10 章では、SIMECK を対象としたフォールト解析手法を提案し、シミュレーション実験によってその有効性を示した。提案した SIMECK に対するフォールト解析手法では、クロックグリッチによるランダムな位置に複数のフォールトが混入するフォールトモデルを導入した。そして、SIMECK のフォールト伝搬に着目した統計的な解析手法によってラウンド鍵を推定した。

第 11 章では、Midori を対象としたフォールト解析手法を提案し、シミュレーション実験によってその有効性を示した。提案した Midori に対するフォールト解析手法では、クロックグリッチによって複数のフォールトがランダムに混入するフォールトモデルを想定し、統計的なフォールト解析手法を導入した。また、Midori の 15R と 16R に対する階層的なフォールト解析を導入することで、全ての秘密鍵を推定した。更に、Isim を用いた遅延付きシミュレーションによってクロックグリッチによるフォールト混入の有効性を確認した。

第 12 章では、PRINCE に対するフォールト解析手法を提案し、シミュレーション実験によってその有効性を示した。提案した PRINCE に対するフォールト解析手法では、フォールト混入位置を精密に制御できない場合でも解析を行うために、統計的な解析モデルを導入した。また、階層的な解析モデルを導入することで、PRINCE の秘密鍵を全て推定することに成功した。更に、枝刈りアルゴリズムを導入することで、解析時間の約 17 倍の高速化を実現した。

以上、本論文では軽量暗号に対するサイドチャネル解析手法を提案し、評価実験によってその有効性を示した。そして、アルゴリズムにおいてどの部分が解析に対して脆弱であるかを明らかにし、耐タンパ実装に必要な設計指針を示した。

今後の展望としては、本論文で明らかにした軽量暗号の脆弱性を保護するための耐タンパ実装の研究が挙げられる。本論文で提案したサイドチャネル解析手法を用いることで、耐タンパ実装の安全性を検証することができる。また耐タンパ実装に関して、これまでに AES 暗号などを対象に様々な実装方式が報告されており、耐タンパ性と対策コスト（回路規模、消費電力、遅延）にはトレードオフの関係があることが知られている。軽量暗号は対象とするアプリケーションや用途によって、満たすべき制約（回路規模、消費電力、遅延）は異なる。そのため軽量暗号の耐タンパ実装では、利用する暗号（小回路規模実装・低消費電力実装・低遅延実装など）に合わせて、対策コストを最小化するような実装が求められる。他にも、サイドチャネル解析に必要な情報が漏洩することを前提とした対策システムの開発も課題である。絶対に安全な対策を施すことは難しいため、情報が漏洩することを前提として、システム全体で対策コストを最小化するような手法の確立が挙げられる。

謝辞

本論文は、著者が名城大学 大学院 理工学研究科 計算機工学研究室で行った研究を取りまとめたものです。本研究の遂行および本学位論文の執筆に関して、多くの方々の御協力・御助言を頂きました。ここに厚く御礼申し上げます。指導教員の吉川雅弥教授には、著者が研究室に配属されて以来、終始研究に対する熱心な御指導、叱咤激励を頂きました。先生には、楽しんで研究に取り組む姿勢を学ばせて頂きました。また、自由な発想のもと研究に取り組むことのできる環境を用意して頂いただけでなく、国内外での多くの学会発表の機会を頂きました。改めて深く感謝の意を表します。

名城大学 理工学部 情報工学科の柳田康幸教授、山田宗男教授、山本修身教授には博士論文副査として、ご丁寧な査読をして頂きました。深く感謝の意を表します。

本学位論文第4章の研究を遂行するに当たり、名城大学 理工学部 情報工学科の旭健作准教授には貴重な御助言を頂きました。この場をお借りして御礼申し上げます。

計算機工学研究室の学生および卒業生の皆様とは、研究だけでなく様々な面でお世話になりました。特に、研究員の浅井稔也氏には、実験環境の構築や解析手法に関して、多くの助言を頂きました。また、2013年度修士課程修了の塚平峻矢氏、後藤輝氏、2014年度修士課程修了の杉岡恭太氏には、著者が研究室に配属された当初、サイドチャンネル解析について、熱心に御指導頂きました。更に、同輩の野原康平氏、松久僚真氏、ポウデルブラカシュ氏、後輩の岩瀬貴都氏、池崎良哉氏とは、日ごろの研究活動をはじめ多くの有意義な時間を共有し、共に切磋琢磨することができました。ここに感謝の意を表します。

本学位論文第5章、第11章および第12章の研究を遂行するに当たり、2015年度修士課程修了の野原康平氏、松久僚真氏、2017年度修士課程修了の池崎良哉氏の協力を頂きました。ここに感謝の意を表します。

日本学術振興会の特別研究員制度(DC2)による助成をして頂いたことを感謝致します。

最後に、いつも暖かく見守ってくれた家族に感謝の意を表しまして、謝辞とさせて頂き

ます。

2018 年 12 月 1 日

付録

軽量暗号アルゴリズム

A.1 TWINE

TWINE [1]の概要を表 A.1 と図 A.1 に示す。TWINE は 16 分割一般化 Feistel 構造を持つ軽量暗号であり、ブロック長は 64bit、鍵長は 80bit と 128bit の 2 種類から選択が可能である。合計で 36 回のラウンド処理を行い、暗号化する。TWINE は図 A.1 に示す暗号処理部と鍵スケジュール部で構成する。A.1.1 項では暗号処理部について、A.1.2 項では鍵スケジュール部について説明する。

表 A.1 TWINE の概要

Table A.1 Outline of TWINE

Structure	Generalised Feistel structure (GFS) with 16 4-bits branches
Block size [bit]	64
Key size [bit]	80 or 128
Number of rounds	36

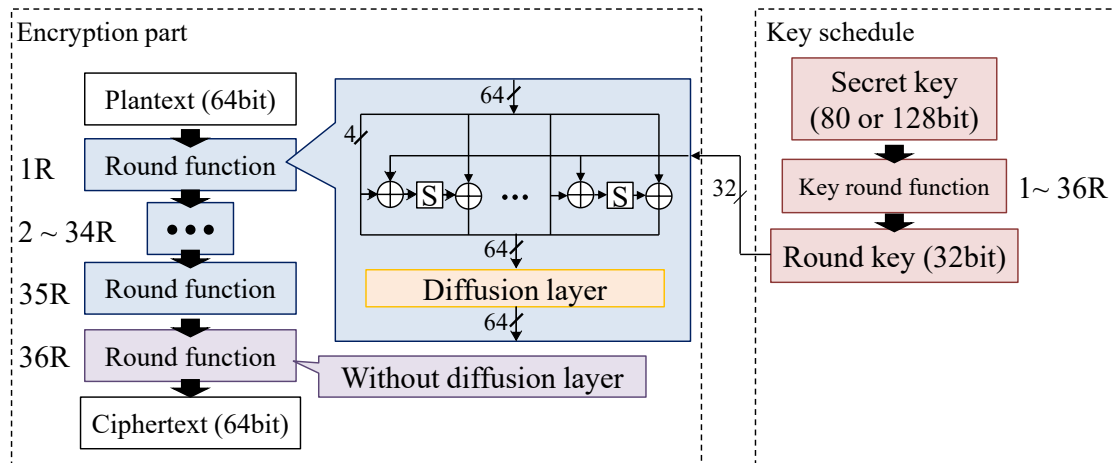


図 A.1 TWINE の概要

Fig. A.1 Outline of TWINE

A.1.1 暗号処理部

まず，暗号処理部について図 A.2 を用いて説明する。ここでは， r ラウンド目 (rR) の j 番目のデータ x を x_j^r と表記する。図 A.2 に示すように，TWINE は 64bit の平文を 4bit ずつの 16 個のデータ ($x_0^1, \dots, x_{15}^1$) に分割して処理する。1R から 35R までは，ラウンド関数による処理を繰り返し行う。この処理では，図 A.2 に示す F 関数による処理と拡散層による処理を行う。そして，最終ラウンドである 36R の処理では，1R から 35R まで繰り返した処理から拡散層を除いた処理を行う。

ここで，暗号処理の詳細について説明する。F 関数は，ラウンド鍵との排他的論理和処理と，S-BOX による置換処理で構成する。このとき，使用するラウンド鍵 RK は，A.2.2 項で説明する鍵スケジュール部で生成したものを使用し，F 関数では 32bit のラウンド鍵 RK を 4bit ずつの 8 個の部分鍵に分割したものを用いる。次に，S-BOX による置換表を表 A.2 に示す。S-BOX による置換処理では，表 A.2 を用いて入力値 x を置換する。また，S-BOX による置換処理を $S(x)$ ，排他的論理和処理を $\oplus$ とすると， rR のラウンド関数の処理は式 (A.1)，(A.2) で表すことができる。

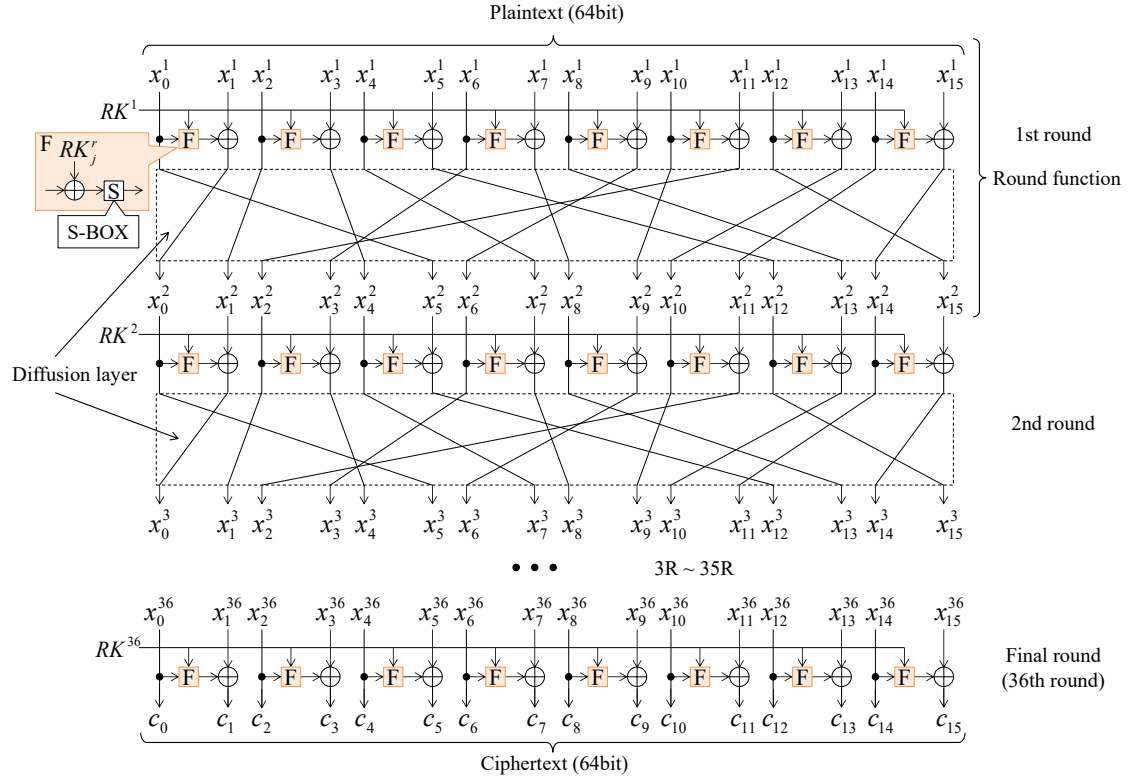


図 A.2 TWINE の暗号処理部

Fig. A.2 Encryption processing part of TWINE

$$\begin{aligned} y_{2j}^r &= x_{2j}^r \\ y_{2j+1}^r &= S(x_{2j}^r \oplus RK_j^r) \oplus x_{2j+1}^r \end{aligned} \quad (A.1)$$

$$\begin{aligned} x_{\pi(2j)}^{r+1} &= y_{2j}^r \\ x_{\pi(2j+1)}^{r+1} &= y_{2j+1}^r \end{aligned} \quad (A.2)$$

ここで、 $\pi()$ は表 A.3 に示す拡散層によって決定するビット位置を表している。また、 r は $1 \leq r \leq 35$ 、 j は $0 \leq j \leq 7$ である。最後に、最終ラウンドである 36R では式(A.1)、(A.2)の処理から拡散層を除いた処理を行う。すなわち、暗号文 c は式(A.3)で計算される。

$$\begin{aligned} c_{2j} &= x_{2j}^{36} \\ c_{2j+1} &= S(x_{2j}^r \oplus RK_j^{36}) \oplus x_{2j+1}^{36} \end{aligned} \quad (A.3)$$

表 A.2 TWINE の S-BOX

Table A.2 S-BOX of TWINE

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S(x)$	C	0	F	A	2	B	9	5	8	3	D	7	1	E	6	4

表 A.3 拡散層

Table A.3 Diffusion layer

h	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$\pi(h)$	5	0	1	4	7	12	3	8	13	6	9	2	15	10	11	14

A.1.2 鍵スケジュール部

鍵スケジュール部について、鍵長が 80bit の場合を例に図 A.3 を用いて説明する。鍵スケジュールでは、80bit の値を 16bit ずつに分割して処理する。そして、S-BOX による置換処理や CON との排他的論理和演算を行うことで、暗号処理で使用するラウンド鍵 RK (8 個の部分鍵) を生成する。このとき、使用する CON の各値を表 A.4 に示す。 CON^r は、各ラウンド r で変化する 6bit の定数値であり、 CON_H^r 、 CON_L^r はそれぞれ CON^r の上位 3bit、下位 3bit を表している。

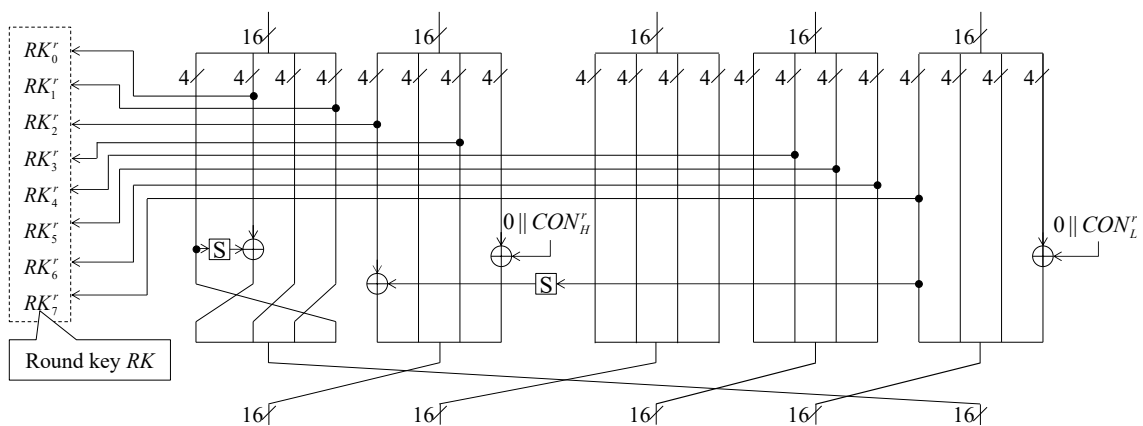


図 A.3 鍵スケジュール部 (鍵長 : 80bit)

Fig. A.3 Key schedule of TWINE (key size: 80bit)

表 A.4 CON

Table A.4 CON

r	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
CON^r	01	02	04	08	10	20	03	06	0C	18	30	23	05	0A	14	28	13	26
r	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	
CON^r	0F	1E	3C	3B	35	29	11	22	07	0E	1C	38	33	25	09	12	24	

A.1.3 実装方法

ここでは、本論文で用いた TWINE の一般的なハードウェア実装について述べる。ASIC や FPGA などのハードウェア実装を行う場合、ループ型アーキテクチャが一般的に広く用いられる。TWINE のループ型アーキテクチャを図 A.4 に示す。図 A.4 に示すように、ループ型アーキテクチャでは、同一のレジスタに対して暗号中間値を格納して、繰り返し処理を行う。

次に、TWINE の S-BOX の構成方法について説明する。A.2.1 項の表 A.2 に示した S-BOX 表は、ガロア体 $GF(2^4)$ の乗法逆元演算（非線形変換）と affine 変換（線形変換）によって導出される。このとき使用する既約多項式を式(A.4)に、affine 変換 θ を式(A.5)に示す。

$$x^4 + x + 1 \quad (A.4)$$

$$\theta = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}_2 \quad (A.5)$$

S-BOX の代表的な構成方法には、小面積で実装が可能な合成体方式 [2]–[4]と、計算結果を真理値表などのテーブルで記述するテーブル方式がある。文献 [1]では、TWINE の S-BOX の構成方法に合成体方式を利用している。合成体方式による S-BOX の構成方法を図 A.5 に示す。図 A.5 に示すように合成体方式では、ガロア体 $GF(2^4)$ の乗法逆元演算を合成体 $GF((2^2)^2)$ 上で行う。この合成体は式(A.6)、(A.7)に示す既約多項式を元に 2 次拡大を繰り返したものを使用する。式(A.6)はガロア体 $GF(2)$ の 2 次拡大を行うときに使用した既約多項式であり、式(A.7)はガロア体 $GF(2)$ の 2 次拡大の 2 次拡大を行うときに使用した既約多項式である。

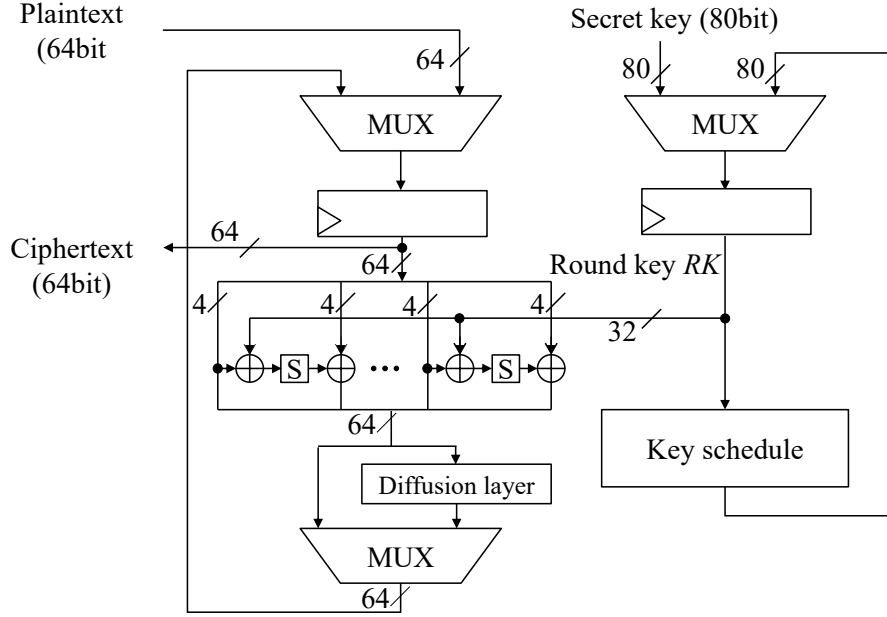


図 A.4 TWINE のループ型アーキテクチャ

Fig. A.4 Loop architecture of TWINE

$$x^2 + x + 1 \quad (\text{A.6})$$

$$x^2 + x + \varphi \quad (\text{A.7})$$

ただし、 φ は $\varphi = (10)_2$ である。

また、ガロア体 $\text{GF}(2^4)$ の元を合成体 $\text{GF}((2^2)^2)$ 上にマッピングする同型写像 δ と、合成体 $\text{GF}((2^2)^2)$ 上の元をガロア体 $\text{GF}(2^4)$ にマッピングし直す逆同型写像 δ^{-1} は、式(A.8)、(A.9)に示すものを使用する。

$$\delta = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_2 \quad (\text{A.8})$$

$$\delta^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_2 \quad (\text{A.9})$$

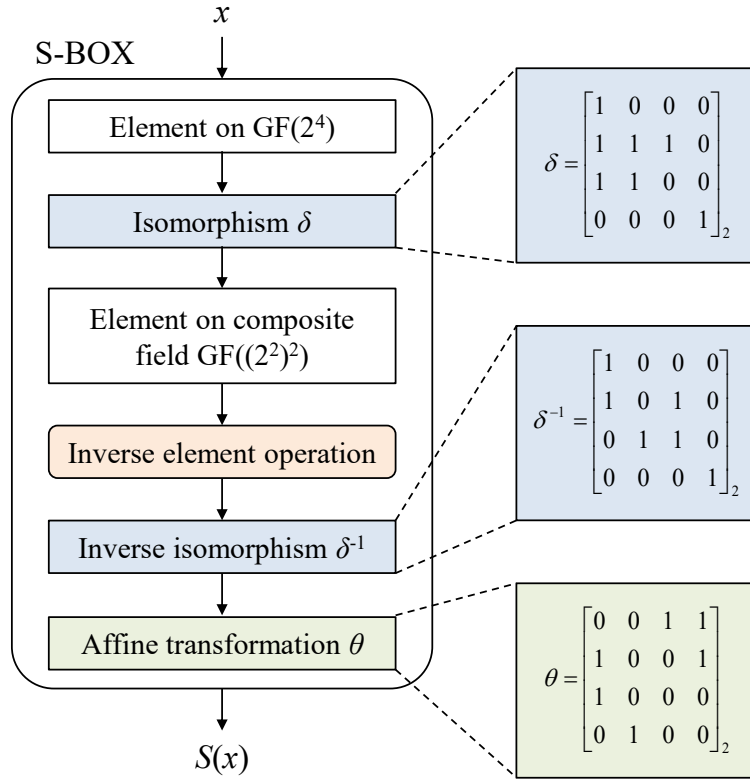


図 A.5 合成体方式による S-BOX

Fig. A.5 S-BOX by composite field

A.2 SIMECK

SIMECK [5]は 2015 年に提案された軽量暗号であり，現在報告されている軽量暗号の中で最も小回路規模でハードウェア実装することができる。その構造は NSA によって公開された SIMON/SPECK [6]をベースとしたものとなっている。

SIMECK のワード長 m は， $m = 16, 24, 48$ であり，ブロック長は $2m$ ，鍵長は $4m$ で定義される。したがって，ブロック長と鍵長の組み合わせはそれぞれ， $(2m, 4m) = (32, 64), (48, 96), (64, 128)$ となる。また，このときラウンド数は $T-1$ で表され， T は各ブロック長と鍵長の組み合わせに合わせて，それぞれ， $T = 33, 37, 45$ と変化する。SIMECK の概要を図 A.6 に示す。図 A.6 に示すように Feistel 構造であり，暗号中間値を左半分の値 L と右半分の値 R に分割して処理する。暗号処理は，主に排他的論理和演算と論理積演算，左ローテーション処理で構成し，それぞれビット単位での演算を行う。そして，これらの処理を繰り返し適用することで暗号化する。暗号中間値 L, R は，それぞれ式(A.10)と式(A.11)で表される。

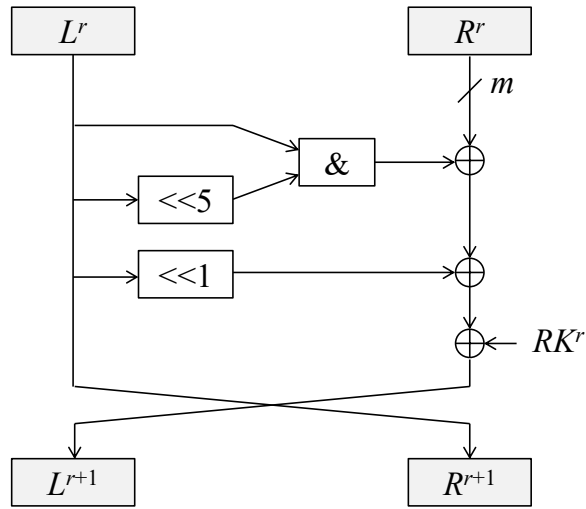


図 A.6 SIMECK の暗号アルゴリズム

Fig. A.6 Encryption algorithm of SIMECK

$$L^{r+1} = R^r \oplus f(L^r) \oplus RK^r \quad (\text{A.10})$$

$$R^{r+1} = L^r \quad (\text{A.11})$$

このとき、 $\oplus$ は排他的論理和演算、 RK はラウンド鍵、 r はラウンドを表している。また、式(A.10)の $f()$ では式(A.12)に示す計算を行う。

$$f(x) = (x \& (x \ll 5)) \oplus (x \ll 1) \quad (\text{A.12})$$

ここで、 $\&$ は論理積演算、 $\ll j$ は j bit の左ローテーション処理を行う。

A.3 Midori

Midori [7]の暗号アルゴリズムの概要を図 A.7 に示す。図 A.7 に示すように、合計で 17 ラウンドの処理を行う。0 ラウンド目 (0R) では平文 P と鍵 WK との排他的論理和演算を行う。次に、1R から 15R までは、ラウンド関数を用いた処理を行う。最後に、16R では SubCell による処理を行い、その後、鍵 WK と排他的論理和演算を行う。ラウンド関数は、SubCell, ShuffleCell, MixColumn, AddKey の 4 つの処理で構成する。各処理の詳細を以下に示す。

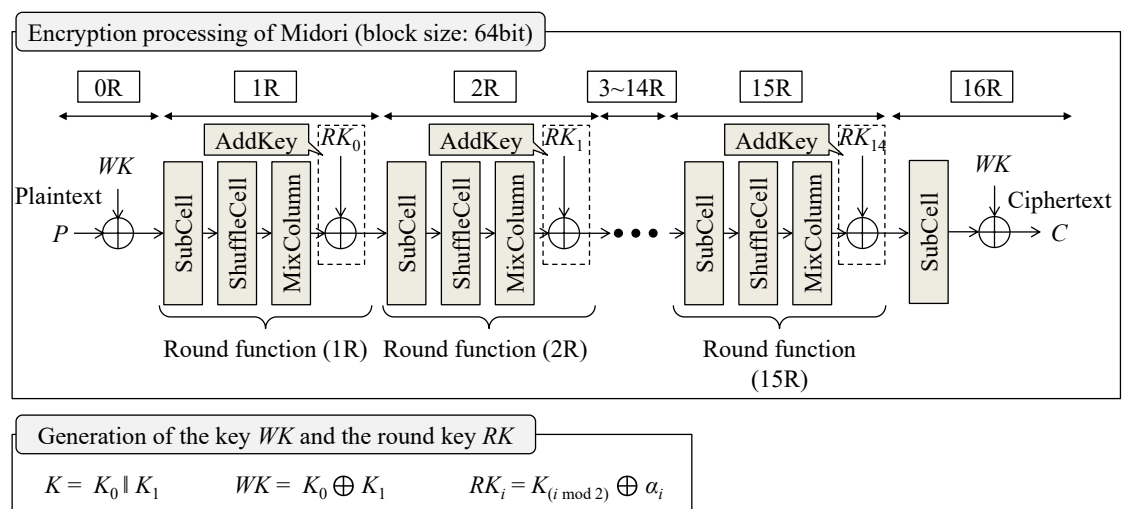


図 A.7 Midori の概要

Fig. A.7 Outline of Midori

表 A.5 Midori の S-BOX

Table A.5 S-BOX of Midori

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S(x)$	C	A	D	3	E	B	F	7	8	9	1	5	0	2	4	6

表 A.6 シャッフル表

Table A.6 Shuffle table

j	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$\pi(j)$	0	7	14	9	5	2	11	12	15	8	1	6	10	13	4	3

SubCell

SubCell は 4bit 単位で処理を行い、表 A.5 に示す S-BOX による非線形置換を行う。また、SubCell の逆関数 S^{-1} には、 $S = S^{-1}$ の関係が成り立つ。

ShuffleCell

ShuffleCell は 4bit 単位で処理を行い、表 A.6 に示す shuffle 表による値の転置を行う。表 A.6 では、 j 番目の値は $\pi(j)$ 番目に転置される。

MixColumn

MixColumn は、16bit 単位での行列演算を行う。MixColumn の計算で使用する行列 M の式を式(A.13)に示す。また、式(A.13)の M は Involution 性を持つため、 $M = M^1$ が成り立つ。

$$M = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \quad (\text{A.13})$$

表 A.7 Midori のラウンド定数

Table A.7 Round constants of Midori

Round ($r - 1$)	Round constants (α)
0	243F
1	6A88
2	85A3
3	08D3
4	1319
5	8A2E
6	0307
7	7344
8	A409
9	3822
10	299F
11	31D0
12	082E
13	FA98
14	EC4E

KeyAdd

KeyAdd は、暗号中間値とラウンド鍵 RK との排他的論理和演算を行う。ここで、Midori は鍵スケジュール部を持たず、暗号処理で使用する鍵 WK やラウンド鍵 RK は、128bit の秘密鍵 K を 64bit ずつに分割した 2 つの鍵 K_0 と K_1 ($K = K_0 \parallel K_1$) から生成する。具体的には、鍵 WK と RK は式(A.14)と式(A.15)で計算する。ここで、 r はラウンド、 α は各ラウンドで変化する定数（ラウンド定数）である。また、ラウンド定数は、表 A.7 に示す円周率の小数部が使用される。

$$WK = K_0 \oplus K_1 \quad (\text{A.14})$$

$$RK_{r-1} = K_{(r-1 \bmod 2)} \oplus \alpha_{r-1} \quad (\text{A.15})$$

A.4 PRINCE

PRINCE [8] は SPN 構造を持つ軽量暗号であり、ブロック長は 64bit、秘密鍵は 128bit である。鍵スケジュール部を持たず、128bit の秘密鍵 k をそれぞれ 64bit の部分鍵 k_0 , k_1 に分割して利用する。

PRINCE の暗号化処理の概要を図 A.8 に示す。暗号化処理は、図 A.8 に示すように、0 ラウンド目（0R）の定数加算処理、1R から 5R までのラウンド関数 F による処理、中間処

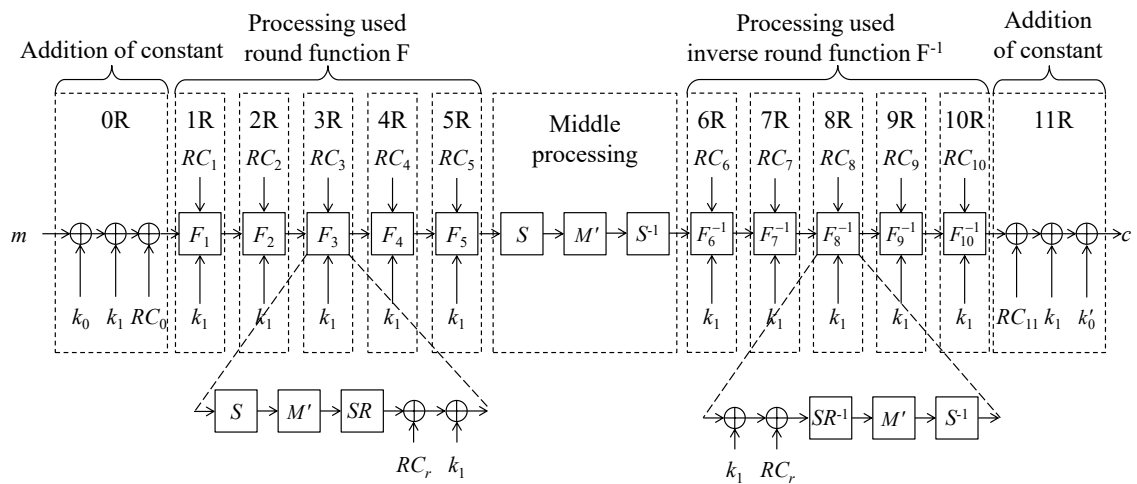


図 A.8 PRINCE の概要

Fig. A.8 Outline of PRINCE

処理, 6R から 10R までの逆ラウンド関数 F^{-1} による処理, 11R の定数加算処理で構成する。

各処理の詳細を以下に示す。

0R の定数加算処理

この処理では、平文 m に対して、部分鍵 k_0 , k_1 とラウンド定数 RC_0 による排他的論理和演算を行う。ここで、 RC は表 A.8 に示すラウンド毎に変化する定数値である。

ラウンド関数 F（1R から 5R までの処理）

ラウンド関数 F の処理では、図 A.9 に示すように S , M' , SR による処理を行い、 RC_r と部分鍵 k_1 による排他的論理和演算を行う。このとき、 RC_r は r ラウンド目における RC の値である。

S は S-BOX による置換処理であり、表 A.9 に従い、4bit のデータ x ごとに置換する。

表 A.8 PRINCE のラウンド定数

Table A.8 Round constants of PRINCE

Round (r)	Round constants (RC_r)
0	0000000000000000
1	13198A2E03707344
2	A4093822299F31D0
3	082EFA98EC4E6C89
4	452821E638D01377
5	BE5466CF34E90C6C
6	7EF84F78FD955CB1
7	85840851F1AC43AA
8	C882D32F25323C54
9	64A51195E0E3610D
10	D3B5A399CA0C2399
11	C0AC29B7C97C50DD

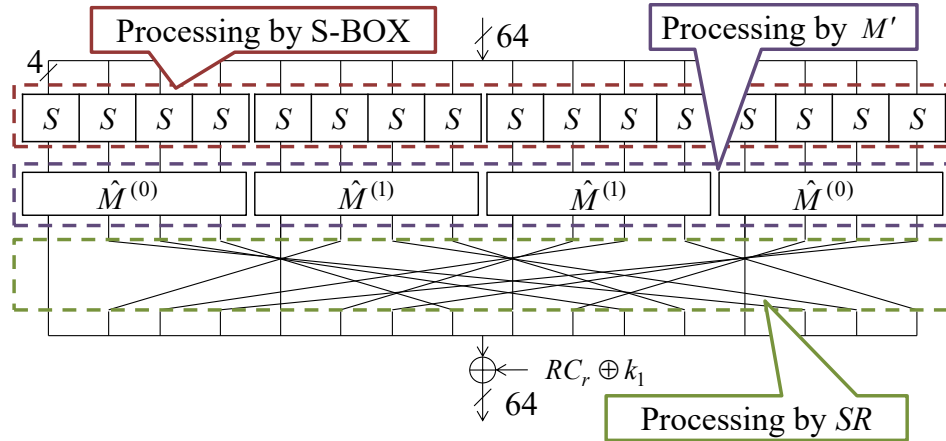


図 A.9 ラウンド関数 F の処理

Fig. A.9 Processing of round function F

表 A.9 PRINCE の S-BOX

Table A.9 S-BOX of PRINCE

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S(x)$	B	F	3	2	A	C	9	1	6	7	8	0	E	5	D	4
$S^{-1}(x)$	B	7	2	3	F	D	8	9	A	6	4	0	5	E	C	1

次に、 M' では 64bit 単位での行列演算を行う。 M' を式(A.16)に示す。

$$M' = \begin{pmatrix} \hat{M}^{(0)} & 0 & 0 & 0 \\ 0 & \hat{M}^{(1)} & 0 & 0 \\ 0 & 0 & \hat{M}^{(1)} & 0 \\ 0 & 0 & 0 & \hat{M}^{(0)} \end{pmatrix} \quad (\text{A.16})$$

M' には $M' = M'^{-1}$ という関係が成り立ち、 M' は $\hat{M}^{(0)}$ と $\hat{M}^{(1)}$ で構成する。 $\hat{M}^{(0)}$ 、 $\hat{M}^{(1)}$ をそれぞれ式(A.17)と式(A.18)に示す。

$$\hat{M}^{(0)} = \begin{pmatrix} M_0 & M_1 & M_2 & M_3 \\ M_1 & M_2 & M_3 & M_0 \\ M_2 & M_3 & M_0 & M_1 \\ M_3 & M_0 & M_1 & M_2 \end{pmatrix} \quad (\text{A.17})$$

$$\hat{M}^{(1)} = \begin{pmatrix} M_1 & M_2 & M_3 & M_0 \\ M_2 & M_3 & M_0 & M_1 \\ M_3 & M_0 & M_1 & M_2 \\ M_0 & M_1 & M_2 & M_3 \end{pmatrix} \quad (\text{A.18})$$

ここで、 $\hat{M}^{(0)}$ 、 $\hat{M}^{(1)}$ はそれぞれ 16bit 単位での行列演算を行う。また $\hat{M}^{(0)}$ 、 $\hat{M}^{(1)}$ には、式(A.19)に示す関係が成り立つ。

$$\begin{aligned} \hat{M}^{(0)} &= (\hat{M}^{(0)})^{-1} \\ \hat{M}^{(1)} &= (\hat{M}^{(1)})^{-1} \end{aligned} \quad (\text{A.19})$$

次に、 $\hat{M}^{(0)}$ 、 $\hat{M}^{(1)}$ を構成する M_0 、 M_1 、 M_2 、 M_3 をそれぞれ式(A.20)から式(A.23)に示す。

$$M_0 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (\text{A.20})$$

$$M_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (\text{A.21})$$

$$M_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (\text{A.22})$$

$$M_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (\text{A.23})$$

SR は AES 暗号で利用される ShiftRows 処理と同様の処理を行う。具体的に、ShiftRows 処理では図 A.10 に示すように、64bit のデータを 4bit ずつの 16 個のデータに分割し、 4×4 の配列として処理する。そして、この配列の各行でそれぞれ左シフト処理を行う。

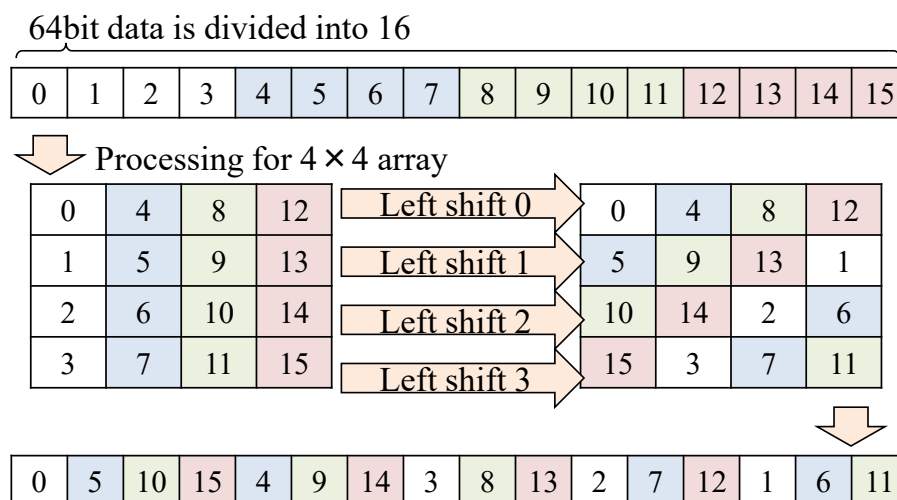


図 A.10 SR の処理

Fig. A.10 Processing of SR

以上が、1R から 5R までのラウンド関数 F を構成する各処理についての説明である。ここで、 M' による行列演算では、64bit のデータを用いて計算する必要がある。しかし、実際には図 A.9 に示すように、 M' を構成する $\hat{M}^{(0)}$ 、 $\hat{M}^{(1)}$ に分割して計算することが可能である。したがって、16bit 単位の行列演算によって計算できる。

中間処理

この処理では、 S 、 M' 、 S^{-1} による処理を行う。 S 、 S^{-1} では、それぞれ表 A.9 に示す S-BOX の置換表・逆置換表による置換処理を行う。また、 M' はラウンド関数 F と同様に行列演算を行う。

逆ラウンド関数 F^{-1} (6R から 10R までの処理)

この処理では、図 A.11 に示すように、部分鍵 k_1 と RC_r による排他的論理和演算を行い、 SR^{-1} 、 M' 、 S^{-1} による処理を行う。 SR^{-1} は SR の逆関数、すなわち、AES 暗号における InvShiftRows 処理と同じ処理を行う。ラウンド関数 F と同様に、 M' の行列演算は図 A.11 に示すように、 $\hat{M}^{(0)}$ 、 $\hat{M}^{(1)}$ を用いて 16bit 単位で計算することが可能である。

ここで、 r ラウンド目の入力値を x^r 、 SR^{-1} の処理を $SR^{-1}(x)$ 、 S^{-1} の処理を $S^{-1}(x)$ と表記すると、 x^{r+1} は式(A.24)で計算できる。

$$x^{r+1} = S^{-1}(SR^{-1}(x^r \oplus RC_r \oplus k_1) \cdot M') \quad (\text{A.24})$$

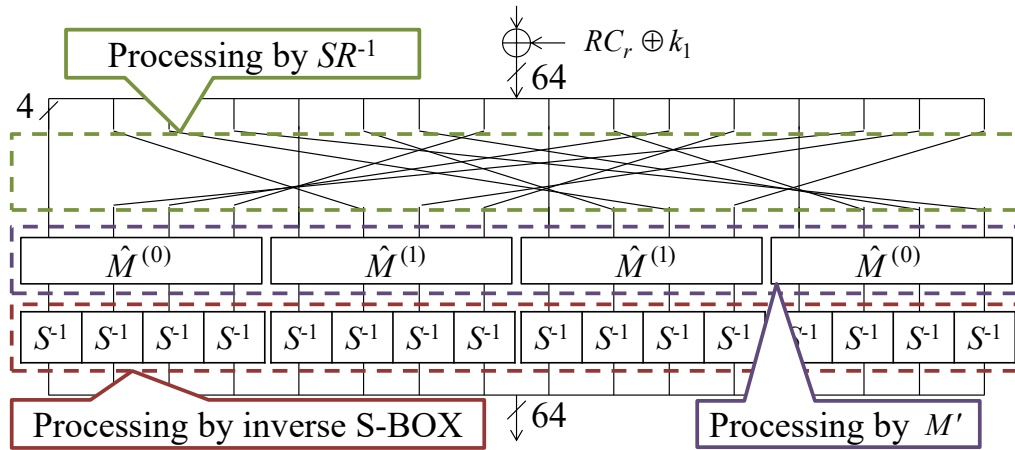


図 A.11 ラウンド関数 F^{-1} の処理

Fig. A.11 Processing of round function F^{-1}

このとき、 j bit 目のデータから k bit 目までのデータを $x_{(j-k)}$ と表記すると、 x^{r+1} を 16bit ずつに分割したデータは $x_{(1-16)}^{r+1}$ 、 $x_{(17-32)}^{r+1}$ 、 $x_{(33-48)}^{r+1}$ 、 $x_{(49-64)}^{r+1}$ と表記することができる。また、式(A.24)から $x_{(1-16)}^{r+1}$ は式(A.25)で計算できる。

$$x_{(1-16)}^{r+1} = S^{-1} (SR^{-1} (x^r \oplus RC_r \oplus k_1)_{(1-16)} \cdot \hat{M}^{(0)}) \quad (\text{A.25})$$

更に、4bit のデータ $x_{(1-4)}^{r+1}$ は式(A.26)で計算できる。

$$x_{(1-4)}^{r+1} = S^{-1} ((SR^{-1} (x^r \oplus RC_r \oplus k_1)_{(1-16)} \cdot \hat{M}^{(0)})_{(1-4)}) \quad (\text{A.26})$$

11R の定数加算処理

この処理では、ラウンド定数 RC_{11} と部分鍵 k_1 、 k'_0 による排他的論理和演算を行うことで、暗号文 c を取得する。このとき、部分鍵 k'_0 の計算式を式(A.27)に示す。

$$k'_0 = (k_0 \gg 1) \oplus (k_0 \gg \gg 63) \quad (\text{A.27})$$

ここで、 $\gg$ は右ローテーション処理、 $\gg \gg$ は右シフト処理である。

A.5 Minalpher

Minalpher [9]–[11]に関して、特に Authenticated Encryption with Associated Data (AEAD) モードの暗号文生成部分を中心に説明する。Minalpher は図 A.12 に示すように、オフセット値生成部分と暗号文生成部分で構成する。そして、256bit の置換を行う P 関数を繰り返し適用する。オフセット値生成部分と暗号文生成部分の詳細を以下に示す。

オフセット値生成部分

オフセット値生成部分では、128bit の秘密鍵 K と 24bit の定数値 $flag_m$ 、104bit のナンス N の合計 256bit の値 $K\|flag_m\|N$ を P 関数に入力する。そして、P 関数の出力と $K\|flag_m\|N$ の排他的論理和である L を利用してオフセット値 φ_i ($i = 1, \dots, m$) を計算する。このときオフセット値は、式(A.28)に示す体上で計算する。式(A.28)の $g(y)$ と $f(x)$ を式(A.29)と式(A.30)に示す。また、オフセット値は文献[9][11]から図 A.13 に示す処理で計算できる。

$$GF(2^{256}) \cong (GF(2)[x]/(f(x)))[y]/(g(y)) \quad (\text{A.28})$$

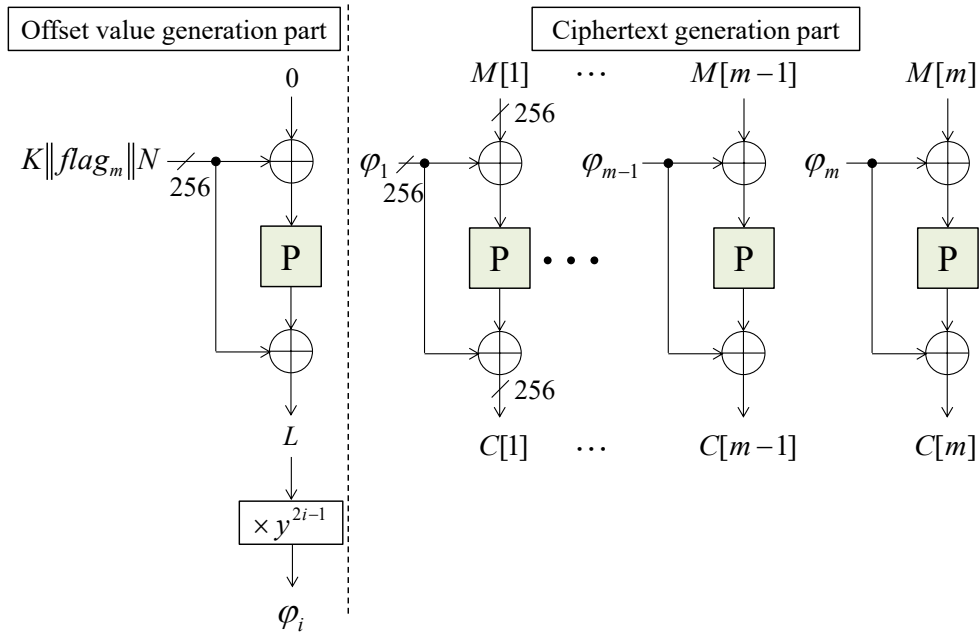


図 A.12 Minalpher の概要

Fig. A.12 Outline of Minalpher

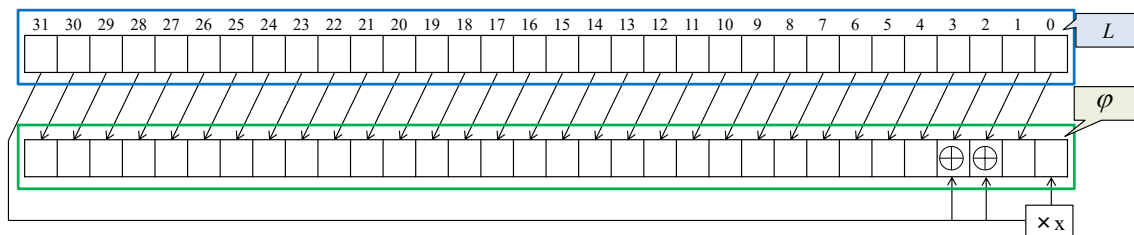


図 A.13 y の乗算の計算例 [9][11]

Fig. A.13 Calculation example for multiplication of y

$$g(y) = y^{32} + y^3 + y^2 + x \quad (\text{A.29})$$

$$f(x) = x^8 + x^7 + x^5 + x + 1 \quad (\text{A.30})$$

暗号文生成部分

暗号文生成部分では、P 関数にメッセージ M とオフセット値 φ_i の排他的論理和である $M \oplus \varphi_i$ を入力する。次に、得られた P 関数の出力とオフセット値 φ_i との排他的論理和演算を行うことで、暗号文 C を計算する。このとき、メッセージ M は 256bit を 1 つのブロックとして、 m 個のブロックに分割して処理する。

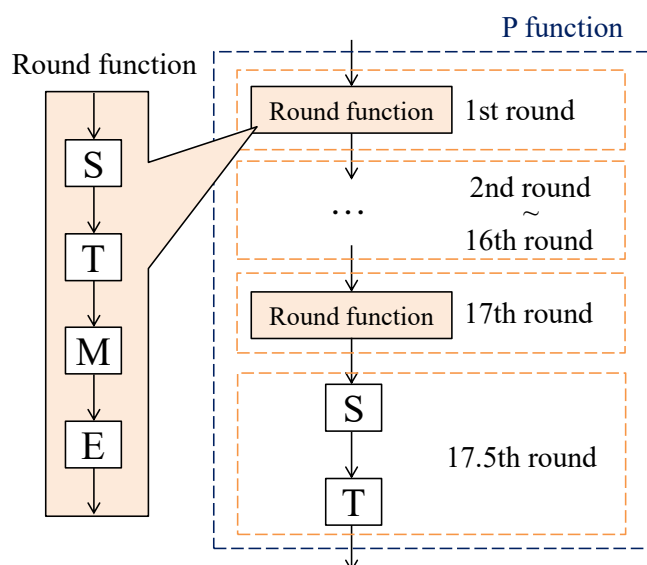


図 A.14 P 関数

Fig. A.14 P function

次に、P 関数について図 A.14 を用いて説明する。図 A.14 に示すように、合計で 17.5 ラウンドの処理を行う。ラウンド処理に関して、Minalpher は 4bit 単位で各処理を行っており、256bit の入力値を 4×8 の行列 A, B に分割して処理する。またラウンド関数は、S 関数、T 関数、M 関数、E 関数で構成しており、1 ラウンド目 (1R) から 17R までは、図 A.15 に示すラウンド関数による処理を、17.5R では S 関数と T 関数による処理を行う。各ラウンド処理の詳細を以下に示す。

S 関数

SubNibbles (SN) 関数による処理を行う。SN 関数では、行列 A, B に対して、表 A.10 に示す S-BOX 表による置換処理を 4bit 単位で適用する。

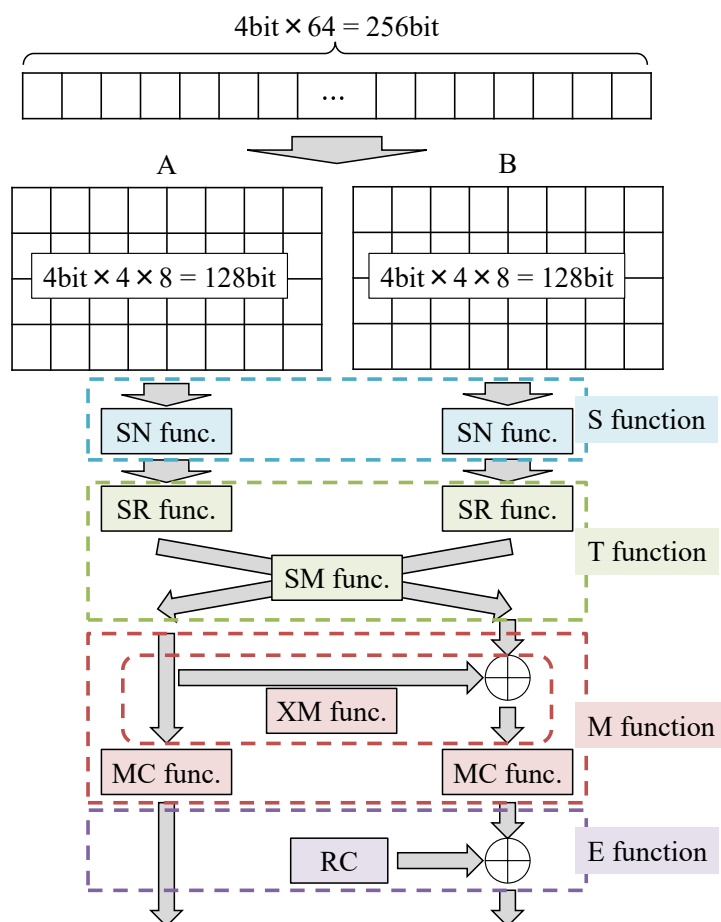


図 A.15 Minalpher のラウンド関数

Fig. A.15 Round function of Minalpher

表 A.10 Minalpher の S-BOX

Table A.10 S-BOX of Minalpher

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S(x)$	B	3	4	1	2	8	C	F	5	D	E	0	6	9	A	7

T 関数

Shuffle Rows (SR) 関数と Swap Matrices (SM) 関数による処理を行う。SR 関数では表 A.11 に示す SR 表による転置処理を行う。具体的には、図 A.16 に示すように、行列 A に対しては各行に対しそれぞれ SR_1 , SR_2 , SR_1^{-1} , SR_2^{-1} を、行列 B に対しては各行に対しそれぞれ SR_1^{-1} , SR_2^{-1} , SR_1 , SR_2 を適用する。SM 関数では、2 つの行列を入れ替える処理を行う。

M 関数

XorMatrix (XM) 関数と MixColumns (MC) 関数による処理を行う。XM 関数では行列 A と行列 B との排他的論理和演算を行う。また、MC 関数では式(A.31)に示す行列演算を行う。

$$\begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix} \quad (\text{A.31})$$

表 A.11 SR 表

Table A.11 SR table.

j	0	1	2	3	4	5	6	7
$SR_1(j)$	6	7	1	0	2	3	4	5
$SR_1^{-1}(j)$	3	2	4	5	6	7	0	1
$SR_2(j)$	4	5	0	1	7	6	2	3
$SR_2^{-1}(j)$	2	3	6	7	0	1	5	4

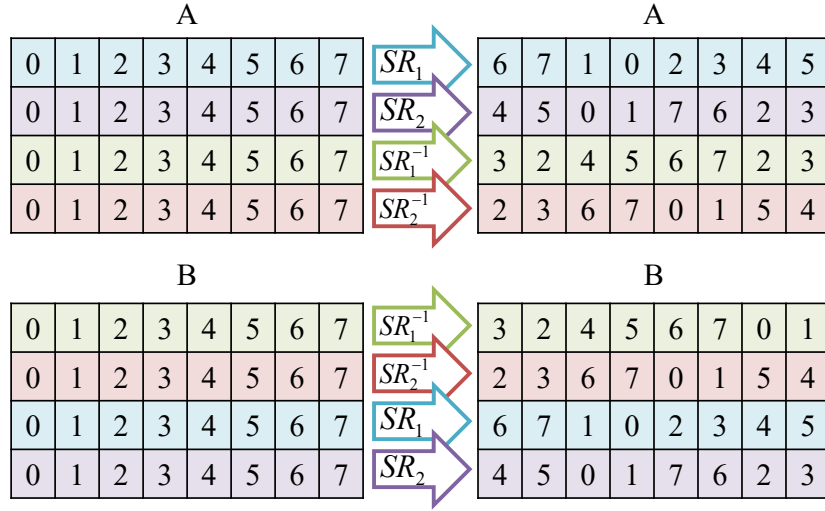


図 A.16 SR 関数

Fig. A.16 SR function

E 関数

行列 B とラウンド定数 (Round Constant: RC) との排他的論理和演算を行う。ここで、RC は式(A.32)に示す式で計算できる。式(A.32)の r はラウンド毎に変化する値であり、 n ラウンド目の r は $r = (n-1) \bmod 16$ で計算できる。

$$\begin{bmatrix} r \oplus 0 & r \oplus 1 & r \oplus 2 & r \oplus 3 & 0 & 0 & 0 & 0 \\ r \oplus 1 & r \oplus 0 & r \oplus 3 & r \oplus 2 & 0 & 0 & 0 & 0 \\ r \oplus 2 & r \oplus 3 & r \oplus 0 & r \oplus 1 & 0 & 0 & 0 & 0 \\ r \oplus 3 & r \oplus 2 & r \oplus 1 & r \oplus 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{A.32})$$

A.6 TWINE-OTR

TWINE-OTR [12]は、図 A.17 に示すようにオフセット値生成部分と暗号文生成部分で構成する。各処理では暗号化関数 E_k を繰り返し適用し、この暗号化関数 E_k には A.1 節で説明した TWINE を使用する。オフセット値生成部分と暗号文生成部分の詳細を以下に示す。

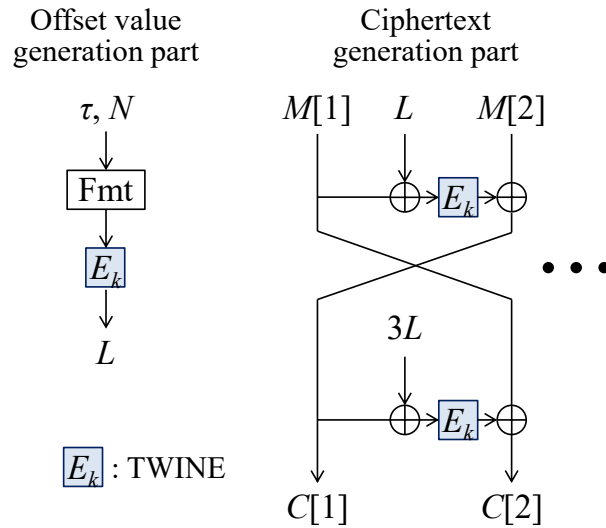


図 A.17 TWINE-OTR の概要

Fig. A.17 Outline of TWINE-OTR

オフセット値生成部分

オフセット値生成部分では、暗号文生成部分で使用するオフセット値 L を計算する。具体的には、 Fmt 関数にタグ長 τ とナンス N を入力する。そして、得られた出力を暗号化関数 E_k に入力し、オフセット値 L を得る。ここで、 Fmt 関数の詳細については、文献 [12] を参照されたい。

暗号文生成部分

暗号文生成部分では、メッセージ M を 64bit ずつの n 個のブロック $M[1], M[2], \dots, M[n]$ に分割して、それぞれ暗号化処理を行う。この暗号化について、暗号文 $C[2]$ を生成する場合を例に説明する。暗号文 $C[2]$ の生成では、まず暗号文 $C[1]$ とオフセット値 $3L$ との排他的論理和演算を行った値 ($C[1] \oplus 3L$) を暗号化関数 E_k に入力する。そして、 E_k の出力とメッセージ $M[1]$ との排他的論理和演算を行い、暗号文 $C[2]$ を計算する。ここで、64bit のメッセージ $M[1]$ を 4bit ずつに分割した 16 個の値を $m_0, \dots, m_{15}$ 、64bit の暗号文 $C[2]$ を 4bit ずつに分割した 16 個の値を $c_0, \dots, c_{15}$ とすると、暗号文 $C[2]$ は式(A.33)、(A.34)で計算できる。ここで、式(A.31)、(A.32)の x^{36} は暗号化関数 TWINE の 35R 終了後の暗号中間値であり、 x^{36} は式(A.2)から計算できる。

$$c_{2i} = x_{2i}^{36} \oplus m_{2i} \quad (\text{A.33})$$

$$c_{2i+1} = S(x_{2i}^{36} \oplus RK_i^{36}) \oplus x_{2i+1}^{36} \oplus m_{2i+1} \quad (\text{A.34})$$

参考文献

- [1] T. Suzaki, K. Minematsu, S. Morioka, and E. Kobayashi, “TWINE: A Lightweight, Versatile Blockcipher,” Proc. ECRYPT Workshop on Lightweight Cryptography (LC11), pp. 146–149, Nov. 2011.
- [2] 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, May 2003.
- [3] A. Satoh, S. Morioka, K. Takano, and S. Munetoh, “A Compact Rijndael Hardware Architecture with S-Box Optimization,” Proc. ASIACRYPT 2001, LNCS vol. 2248, pp. 239–254, Springer-Verlag, Dec. 2001.
- [4] 森岡澄夫, 佐藤 証, 高野光司, “GF(((2^2)^2)^2)上の演算を用いた AES の S-Box 構成法,” 情報処理学会第 63 回全国大会講演論文集, 3G-04, Sep. 2001.
- [5] G. Yang, B. Zhu, V. Suder, M. D. Aagaard, and G. Gong, “The Simeck Family of Lightweight Block Ciphers,” Proc. 17th Int. Workshop on Cryptographic Hardware and Embedded Systems (CHES 2015), LNCS vol. 9293, pp. 307–329, Springer, Sep. 2015.
- [6] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers, “The SIMON and SPECK Families of Lightweight Block Ciphers,” IACR Cryptography ePrint Archive: Report 2013/404, Jun. 2013.
- [7] S. Banik, A. Bogdanov, T. Isobe, K. Shibutani, H. Hiwatari, T. Akishita, and F. Regazzoni, “Midori: A Block Cipher for Low Energy,” Proc. ASIACRYPT 2015, LNCS vol. 9453, pp. 411–436, Springer, Dec. 2015.
- [8] J. Borghoff, A. Canteaut, T. Güneysu, E. B. Kavum, M. Knežević, L. R. Knudsen, G. Leander, V. Nikov, C. Paar, C. Rechberger, P. Rombouts, S. S. Thomsen, and T. Yalçin, “PRINCE - A Low-latency Block Cipher for Pervasive Computing Applications,” Proc. ASIACRYPT 2012, LNCS vol. 7658, pp. 208–225, Springer, Dec. 2012.

- [9] Y. Sasaki, Y. Todo, K. Aoki, Y. Naito, T. Sugawara, Y. Murakami, M. Matsui, and S. Hirose, “Minalpher v1.1,” Aug. 2015. <http://info.isl.ntt.co.jp/crypt/minalpher/files/minalpherv11.pdf>
- [10] 佐々木悠, 藤堂洋介, 青木和麻呂, 内藤祐介, 菅原 健, 村上ユミコ, 松井 充, 廣瀬勝一, 高橋克己, “改ざん検知暗号 Minalpher,” 2015 年暗号と情報セキュリティシンポジウム講演論文集, 2E-1, pp. 1–4, Jan. 2015.
- [11] 青木和麻呂, 藤堂洋介, “Minalpher の x86_64 最適化実装,” 2015 年暗号と情報セキュリティシンポジウム講演論文集, 1B1-4, pp. 1–8, Jan. 2015.
- [12] K. Minematsu, “AES-OTR v3.1,” Sep. 2016.
<http://competitions.cr.yp.to/round3/aesotrv31.pdf>