博 士 論 文

暗号モジュールの安全性評価手法に関する研究

通信工学専攻
伊東 燦

Security Evaluation Methods of Cryptographic Modules

by

Akira Ito

Graduate School of Engineering, Tohoku University, 2022

Abstract

Cryptography is expanding to realize information security functions such as secret communication, authentication, and digital signature. This trend is expected to intensify in the next-generation information and communication systems such as the Internet of Things (IoT). In particular, IoT systems include not only servers and personal computers (PCs) but also embedded systems such as mobile terminals and in-vehicle systems. Thus, the efficient implementation of dedicated modules for cryptographic computation (cryptographic modules) is essential for embedded devices.

Countermeasures against tampering attacks are important because cryptographic modules in embedded devices are the starting point of trust in information systems. Physical attacks on cryptographic modules can be a serious problem in IoT systems with numerous unmonitored devices. Recently, a side-channel attack has emerged as a severe threat. A side-channel attack is an attack that exploits the leakage of information related to cryptographic operations from physical information (e.g., power consumption, electromagnetic radiation, processing time) that occurs as a side effect of cryptographic operations. Side-channel attacks are considered as a practical threat because an adversary does not need expensive equipment to conduct these attacks and will leave no traces. In fact, side-channel attacks have attracted a great deal of attention at conferences related to information security, and there have been many studies on detection and countermeasure methods. Previous studies show that most of the leakage from side-channel information is caused by the data path that constitutes the cryptographic module.

Meanwhile, it is not easy to implement cryptographic modules securely. Countermeasures against side-channel attacks require that the cryptographic modules are designed without errors. If even one input of a cryptographic module causes a failure, it leads to the leakage of secret information. Therefore, it is necessary to remove bugs completely that may cause vulnerabilities in the functional design of cryptographic modules. However, the number system used by cryptography makes the design of

cryptographic modules difficult. For instance, Galois field (finite field) arithmetic is employed to realize widely used cryptographic algorithms, such as the ISO/IEC standard cipher AES (Advanced Encryption Standard) and ECC (Elliptic Curve Cryptography). In the case of cryptographic hardware design, the current LSI design automation (EDA) technology does not provide functions for high-level synthesis and design support for Galois-field arithmetic. Therefore, it is necessary to manually convert the arithmetic algorithms represented by Galois field into logical formulas, which requires a great deal of effort. This makes it hard to design and optimize the cryptographic modules.

It is also challenging to evaluate the security of the designed cryptographic modules against side-channel attacks. Recently, a new type of attack method called DL-SCA (Deep-Learning based Side-Channel Attack) has been proposed. It consists of a profiling phase and an attack phase. In the profiling phase, the attacker obtains side-channel information from a device with the same model number as the target device, and extracts device-specific features using deep learning. In the attack phase, the attacker uses the model created in the profiling phase to efficiently estimate the secret key from the side-channel information of the target device. DL-SCA supplements the information that is difficult to use in conventional side-channel attacks (e.g., assumptions about noise and knowledge about the attacked device) by learning during the profiling phase, thus enabling robust and powerful attacks in cryptographic module environments. DL-SCA is applicable to provable-secure cryptographic modules that are difficult to attack by conventional side-channel attacks. However, DL-SCA was proposed only a few years ago, and its applicability is still unclear. It is required to accurately understand the threat of DL-SCA in the security (vulnerability) assessment of cryptographic modules because adversaries can easily profile cryptographic modules due to the IoT.

In this dissertation, we aim to establish a method to evaluate the security of cryptographic modules which are protected against side-channel attacks. First, (i) for logical security evaluation, we develop an efficient verification technique for cryptographic hardware based on Galois arithmetic. Second, (ii) for physical security evaluation, we propose a method for evaluating the security of cryptographic modules using DL-SCA. For (i) and (ii), the outline of this dissertation is as follows.

Regarding (i), we propose a highly efficient equivalence verification method for cryptographic hardware based on Galois-field arithmetic. The proposed method takes a gate-level netlist of cryptographic hardware and its design specification and returns

whether they are equal or not. Using the proposed method, we can check whether cryptographic hardware is designed according to the specification and thus detect all possible causes of vulnerabilities such as bugs. We demonstrate the effectiveness of the proposed methods through experimental verification of some practical circuits, including AES and ECC.

For (ii), this paper proposes a method to evaluate the security of cryptographic modules against DL-SCA. In (ii), we propose efficient attack methods using DL-SCA against some cryptographic modules. The reason for proposing the attack method is that it can be used to detect physical vulnerabilities in cryptographic modules. In this paper, we solve the problem of security evaluation by DL-SCA for both symmetric-key and public-key cryptographic modules. Specifically, for the symmetric-key cryptography, we propose a solution to the imbalanced data problem, which is an open problem in applying DL-SCA. The proposed method uses the probability of key value instead of the HW/HD because the occurrence probability of key value is uniform. The use of the probability of key value mitigates the imbalanced data problems derived by the binomial distribution. For the latter, DL-SCA for public-key cryptographic modules, we propose an attack method against quantum computer cryptography, for which the applicability of DL-SCA has not been studied. The proposed methods in (i) and (ii) will lead to the detection of vulnerabilities in cryptographic modules and consequently to the security evaluation.

数式の表記

本論文全体で使用する数式の表記について述べる．$\mathbb{R}, \mathbb{Q}, \mathbb{N}, \mathbb{Z}$ はそれぞれ，実数，有理数，自然数，整数の全体を表す．任意の実数 a, b について，開区間，閉区間，左半開区間，右半開区間をそれぞれ $(a,b) \coloneqq \{x \in \mathbb{R} \mid a < x < b\}$，$[a,b] \coloneqq \{x \in \mathbb{R} \mid a \leq x \leq b\}$，$(a,b] \coloneqq \{x \in \mathbb{R} \mid a < x \leq b\}$，$[a,b) \coloneqq \{x \in \mathbb{R} \mid a \leq x < b\}$ と定義する．$\mathbb{R}_+ = [0,\infty)$，$\mathbb{Z}_+ = \{0\} \cup \mathbb{N}$ とおく．任意の素数のベキ $p \in \mathbb{N}$ に対して，$\mathbb{F}_p$ を元の個数が p の有限体とする．ベクトルや行列は，$\boldsymbol{x}$ などの太字のイタリック体で表記する．本論文で出現するベクトルは原則として列ベクトルとする．

原則的に Calligraphic 書体の大文字は集合を表すこととする．また確率変数を定義する際は大文字のイタリック体，そのとり得る値（元）を小文字のイタリック体で表記する．例えば，集合 $\mathcal{X}$ に値を取る確率変数は，X と表記し，$\mathcal{X}$ の元は x とする．確率測度を Pr とする（例えば，離散確率変数 X が値 x を取る確率は $\Pr(X = x)$ となる）．確率変数 X が値として元 x を取る事象を $X = x$，もしくは $x \leftarrow X$ と表記する．確率変数 X の期待値を $\mathbb{E}X$ とする．本論文では任意の確率変数に対して，その確率密度関数もしくは確率質量関数が存在すると仮定する．例えば，確率変数 X, Y の同時確率は $q_{X,Y}(x,y)$ のように表記する．確率変数 X, Y に対し，X が与えられたときの Y の条件付き確率分布を，$q_X(x) \neq 0$ ならば $q_{Y|X}(y \mid x) = q_{Y,X}(y,x)/q_X(x)$ と定義する．

サンセリフ体の記号は原則としてアルゴリズムを表す．例えば，暗号プリミティブによる暗号化は Enc，復号化は Dec と表記する．

実数列 $\{a_n\}_{n=1,2,\ldots}$ と $\{b_n\}_{n=1,2,\ldots}$ に対して，$n \to \infty$ のとき $a_n = O(b_n)$ は，$\sup_n |a_n/b_n| < \infty$ を表す．

目次

第 1 章

緒言

情報化社会の深化に伴い，秘匿通信や認証，電子署名などの情報セキュリティ機能実現のための，暗号技術の利用が拡大している．具体的な応用例として，電子マネーやクレジットカードなどのスマートカード，暗号電子メールや電子商取引に用いられるSSL/TLS（Secure Socket Layer/Transport Layer Security）[1] などが挙げられる．この傾向は，モノのインターネット（IoT: Internet of Things）に代表される次世代情報通信システムにおいて益々強まると予想される．IoT はあらゆる機器がインターネットに接続され相互にやり取りを行う概念であり，膨大な数の IoT デバイスによってシステムが構成される．IoT システムでは，ノード間の通信を盗聴・改ざんすることでシステム全体へ攻撃が可能なことから，暗号技術を用いて通信の保護を行う必要がある．また，サーバやパーソナルコンピュータ（PC）に加えて，モバイル端末や車載システムなどのリソース制約の厳しいデバイスも構成要素となり得る．したがって，効率的な暗号処理に向けて，暗号計算のための専用モジュール（暗号モジュール）による実装が必要不可欠である．

組み込み機器における暗号モジュールは，IoT などの情報システムにおける信頼の起点であり，改ざん攻撃への対策は必須である．暗号モジュールが実現する暗号アルゴリズムは，簡単化された通信モデルにおける攻撃を想定して，数学的に安全性を定義している．言い換えれば，たとえ数学的に安全性が保証された暗号アルゴリズムを使用しても，前提となる通信モデルを破るような攻撃に対しては安全性を担保できない．そのような脅威として，暗号モジュールの実装の不備を突くような物理攻撃が挙げられる．人によって直接監視されていない無数のデバイスが占める IoT システムでは，このような物理攻撃は大きな問題となり得る．物理攻撃には様々な手段が存在するが，大まかに侵入型と非侵入型の２つに分けられる [2]．侵入型攻撃とは，暗号モジュールの実装された LSI チップのパッケージを開封し，回路内部に対する直接的な観察・改変を行う攻撃のことである．侵入型攻撃は，実装された回路に直接アクセスすることで，強力な攻撃が可能だが，LSI（Large Scale Integration）に関する専門知識と，顕微鏡やプローブステーションなどの高価な機

器を必要とする．また，LSI のプロセスの微細化に伴い，攻撃を行うこと自体が困難となりつつある [3]．非侵入型の攻撃では，LSI チップの開封などを行わずに，暗号モジュールの解析を行う．非侵入型の攻撃には，主にフォルト攻撃とサイドチャネル攻撃がある．フォルト攻撃では，暗号処理の実行中に不正なクロック信号や入力電圧の印加，チップへのレーザー照射を行い，一時的な誤作動を引き起こして計算誤りを誘発させる．そして，計算誤りのパターンから秘密鍵を解析する．一方，サイドチャネル攻撃は，暗号演算において副次的に発生する物理的変量（消費電力，漏洩電磁波，処理時間など）に，暗号演算に関する情報が漏洩することを利用する攻撃である．物理的変量によって漏洩する情報をサイドチャネル情報と呼び，サイドチャネル攻撃では，このサイドチャネル情報を用いることで暗号内で扱われている秘密情報の抽出を行う．これらの攻撃は，暗号モジュールへ非侵入で実行できるという点で，強力な攻撃である．特に，サイドチャネル攻撃は安価な PC とオシロスコープで実行することができ，攻撃の痕跡も残らないことから，現実的な脅威として認識されている．事実，情報セキュリティ関連の学会でもサイドチャネル攻撃は高い関心を集めており，その検知・対策手法に関する研究報告が相次いでなされている．過去の研究から，サイドチャネル情報からの情報の漏洩の多くは，暗号モジュールを構成するデータパスによって引き起こされることが知られており [4]，サイドチャネル攻撃の対策では，対策回路を暗号モジュールに適切に実装することが求められる．

一方で，暗号モジュールを安全に実装することは容易ではない．暗号モジュールは，1 つでも故障を引き起こす入力が存在すると，秘密情報の漏洩に直結する [5]．また，暗号モジュールのサイドチャネル対策では，実装の不備は物理攻撃に対する脆弱性となる．そのため，暗号モジュールの機能設計では，バグなどの脆弱性となりえる要素は完全に取り除く必要がある．現在広く用いられている暗号アルゴリズムである AES（Advanced Encryption Standard）や，楕円曲線暗号（ECC: Elliptic Curve Cryptography）では，ガロア体上の算術を用いて構築されている．しかし，ガロア体算術は，主に暗号や符号理論で使用される数体系であり，通常の設計・製造環境ではサポートされていない．一例として，暗号ハードウェアの実装に着目すると，現在の LSI の設計自動化（EDA:Electronic Design Automation）技術で，標準的に用いられるハードウェア記述言語（HDL: Hardware Description Language）には，ガロア体の演算を行うための高位合成や設計支援のための機能は用意されていない．そのため，これらの暗号アルゴリズムを回路として実装するには，ガロア体で表現される算術アルゴリズムを人手で論理式に変換する必要があるため，多大な労力がかかる．これは，暗号モジュールの設計および最適化を困難にしている．特に，サイドチャネル攻撃の対策を行うには，暗号モジュールのデータパスが，d-probing モデルと呼ばれる数学的な条件を満たす必要があり [6]，人手で誤りなく設計することは容易ではない．さらに深刻な問題として，設計した回路の機能検証手法の欠如がある．一般に，暗号回路の入出力長は少なくとも 64 ビットを有し，網羅的なテストパターンを

用いる論理シミュレーションには膨大な時間がかかる．論理シミュレーションを用いない，形式的検証手法も盛んに研究されているが，それらの殆どは整数や浮動小数点演算器向けのものであり，直接ガロア体演算器に適用することは難しい．例えば，従来の算術演算回路の形式的検証では，回路の出力関数を表す BDD（Binary Decision Diagram）や BMD（Binary Moment Diagram）などのグラフの同型判定が用いられる．しかし，BDD や BMD などの決定グラフを用いた方法は，たかだか 16 ビット程度のガロア体乗算器であっても現実的な時間では検証できないことが示されている [7].

また，設計した暗号モジュールのサイドチャネル攻撃に対する安全性評価にも課題がある．暗号モジュールの公的な認証制度である JCMVP（Japan Cryptographic Module Validation Program）では，暗号モジュールのサイドチャネル攻撃耐性評価として，t 検定に基づく相関ベースの漏洩評価が行われている [8]．これは，差分電力解析や相関電力解析，テンプレート攻撃などの従来の攻撃を前提として考案された安全性評価である．これらの攻撃では，暗号演算の計算途中で生じる中間値が，ハミング重み（HW: Hamming Weight）やハミング距離（HD: Hamming Distance）などの形で，サイドチャネル波形のある一点でのみ漏洩するという仮定をおいている．一方で，近年，新種の攻撃方法として，深層学習を利用したサイドチャネル攻撃（DL-SCA: Deep-Learning based Side-Channel Attack）が提案された [9]．DL-SCA はプロファイリング型のサイドチャネル攻撃の一種であり，プロファイリングフェーズと攻撃フェーズから構成される．プロファイリングフェーズでは，攻撃者は，自身が所有する攻撃対象と同じ型番のデバイスから予めサイドチャネル情報の取得を行い，深層学習を用いてデバイス固有の特徴をモデルとして抽出する．次に，攻撃フェーズではプロファイリングフェーズで抽出したモデルを利用して，攻撃対象デバイスのサイドチャネル情報から効率的に秘密鍵を推定する．DL-SCA は，従来のサイドチャネル攻撃と異なり，サイドチャネル波形全体から中間値の推定を行うため，波形に含まれるすべての（相関に限らない）情報を利用した攻撃が可能となる．加えて，DL-SCA は，ノイズに関する仮定や攻撃対象デバイスに関する知識をプロファイリングフェーズの学習により補うことができるため，暗号モジュールの環境に頑健な攻撃が可能である．事実，DL-SCA により，従来のサイドチャネル攻撃では攻撃が難しい，対策済みの暗号モジュールからも効率的な鍵回復攻撃が可能であることが知られている．ただし，DL-SCA は比較的新しい技術であり，その適用可能性について，未だ不明な点が多い．今後，IoT の普及や人工知能（AI: Artifical Inteligence）技術の発展により，攻撃者が容易にプロファインリングが可能となることを鑑みれば，暗号モジュールの安全性（脆弱性）評価において，DL-SCA の脅威を正確に把握することは必須である．

本論文では，上記に示した課題に対して，「サイドチャネル攻撃対策された暗号モジュールの安全性評価技術の確立」を目指して，以下の 2 つの項目について研究を遂行する．まず，(i) 暗号モジュールの論理的な安全性評価手法として，ガロア体算術に基づく暗号ハー

ドウェアの効率的な検証技術を開発する．次に，(ii) 暗号モジュールの物理的な安全性評価手法として，深層学習に基づくサイドチャネル攻撃耐性評価手法を提案する．(i) と (ii) について，本論文の概要を以下に述べる．

(i) に関して本論文では，ガロア体算術に基づく暗号ハードウェアの高効率な等価性検証手法を提案する．提案手法は，入力として暗号ハードウェアのゲートレベルネットリストと，その設計仕様を受け取り，出力として両者が等しいかどうかを返す．提案手法を用いることで，与えられた任意の暗号ハードウェアに対して，仕様どおりに設計されているかを調べることができるため，バグなどの脆弱性となりえる原因をすべて検出できる．提案する検証手法は，ガロア体を用いる暗号回路と親和性の高い計算機代数に基づく．具体的には，まず設計仕様から，暗号ハードウェアが満たすべき入出力関係をブール多項式として抽出する．次に，ゲートレベルネットリストを構成するすべてのゲートから，ゲート機能を表現するブール多項式を抽出し，グレブナー基底と呼ばれる多項式集合を生成する．最後に，回路のプライマリ出力変数をグレブナー基底を用いて簡約し，この簡約結果が仕様のブール多項式と一致するかを調べることで，等価性を判定する．提案手法の特徴は，検証過程において，ゼロサプレス型二分決定グラフ（ZDD: Zero-suprressed Binary Decision Diagram）[10] を多項式の表現として用いる点にある．ZDD は，多項式をコンパクトに表現できるため，検証時間を大幅に短縮できる．提案手法では，ZDD を効率的に活用可能な，新たな多項式簡約アルゴリズムを提案する．提案する簡約アルゴリズムを用いることで，サイドチャネル攻撃の対策が施された AES ハードウェアのような実用的な回路も約 11 秒で検証できることを実験的に示す．

(ii) に関して本論文では，暗号モジュールの DL-SCA による安全性評価手法を提案する．(ii) では，与えられた暗号モジュールのサイドチャネル攻撃に対する耐性の評価を行う．具体的に本論文では，共通鍵暗号と公開鍵暗号モジュールのそれぞれについて，DL-SCA による安全性評価における課題の解決を行う．まず共通鍵暗号モジュールの安全性評価について述べる．共通鍵暗号モジュールに DL-SCA を適用する際の未解決問題として，学習データが二項分布に従うために，学習及び推論が適切に行われない不均衡データ問題がある．多くの既存の DL-SCA では，プロファイリング用の波形を用いて暗号アルゴリズムの中間値（例えば AES 第一ラウンドの S-box の出力値）を予測するモデルを学習させる．ここで，学習のコストを削減するため，モデルのクラスラベル（出力）として，中間値の HW/HD がしばしば使用される．しかしながら，HW/HD は不均衡な二項分布に従うため，推論時に少数派クラスが軽視され，推論精度に深刻な問題をきたす．このようなデータのクラスラベルの分布が不均衡なことに由来して発生する問題は，不均衡データ問題と呼ばれる．本論文では，この不均衡データ問題への解決策として，鍵の尤度の使用を提案する．これは，従来の鍵推定に用いられてきた HW/HD の尤度を，鍵値に関する同時確率（尤度）へ変更することで，分布の偏りを除去し，攻撃性能を向上

させる．

次に，公開鍵暗号モジュールに対する DL-SCA について説明する．本稿では，公開鍵暗号の中でも PQC（Post Quantum Cryptography）に注目する．PQC とは，量子計算機が実用化した際にも，安全性が保証される暗号方式のことである．現在 PQC の標準化に向けて，NIST（アメリカ国立標準技術研究所）が PQC のコンペティションを実施しており，候補となる暗号スキームの数学的および実装面の安全性に関して精力的に研究が行われている．PQC と DL-SCA は，どちらも比較的新規の技術であり，DL-SCA の PQC への適用はほとんど検討されていない．そこで，本稿では，DL-SCA に対する PQC の安全性評価の一環として新たな攻撃手法を提案する．提案手法は，PQC を利用した鍵カプセル化（KEM: Key Emcapsulation Mechanism）による鍵配送プロトコルを対象とする．現在，有力候補となっているすべての KEM スキームにおいて，内部処理に FO 変換（Fujisaki–Okamoto 変換）が使用されている．本論文では，FO 変換への DL-SCA により，候補となっているほぼすべての KEM スキームへ攻撃が可能なことを実証する．

本論文は，以上の内容を取りまとめたものであり，以下に示す 6 章によって構成される．第 1 章は，本研究の背景，目的および本論文の概要をまとめた緒言である．第 2 章では，暗号モジュールの安全性評価に関する基礎的考察を行う．まず，暗号技術とその実装技術について述べる．次に，暗号モジュールに対する実装攻撃の脅威と，サイドチャネル攻撃およびその対策手法について説明する．第 3 章では，暗号ハードウェアの設計時における脆弱性検知を目的とした，形式的検証手法を提案する．ZDD を用いた効率的な多項式簡約アルゴリズムを提案し，ECC や AES ハードウェアの検証実験を通してその有効性を示す．第 4 章では，DL-SCA を用いた共通鍵暗号モジュールの脆弱性検知として，効率的な暗号解読手法を提案する．第 5 章では，DL-SCA を用いた公開鍵暗号モジュールに対する攻撃可能性を指摘する．第 6 章は結言である．

以上，本論文の企図するところを概説した．

第 2 章

暗号モジュールの安全性評価に関する基礎的考察

2.1　はじめに

本章では，暗号モジュールの脆弱性検知に関する基礎的考察を行う．まず，暗号技術全般に関して概要を述べ，代表的な暗号アルゴリズムの解説を行う．次に，暗号アルゴリズムを実装した暗号モジュールとそれに対する物理攻撃について述べる．さらに，本論文で主に取り扱うサイドチャネル攻撃について，原理，分類，対策法を述べる．

2.2　暗号とその実装に関する基礎的考察

本節では，暗号技術とその実装に関する概要を述べる．以下では，まず情報セキュリティにおける暗号技術の位置づけを述べ，特に重要な構成要素である暗号アルゴリズムについて説明する．その後，暗号アルゴリズムの実装形態として，暗号モジュールについて述べた上で，暗号モジュールに対する物理攻撃について概説する．また，暗号モジュールの安全性評価についても述べる．

2.2.1　暗号技術の概要

情報セキュリティ技術は，様々な脅威に対抗して，情報システムを安全に運用するための技術である．経済協力開発機構（OECD）の情報セキュリティガイドラインでは，情報セキュリティの構成要素として「機密性」,「完全性」,「可用性」を挙げている [11]．機密性とは，権限を持たない第三者に情報がもれないことである．完全性は，情報が正確であり改ざんされていないことである．そして，可用性は許可を与えられたものが，いつでも情報にアクセスできることを指す．安全性を保証するためには，扱う情報の特性に合わせて，これら 3 つの要素を適切なレベルで維持しなければならない．

暗号技術は，これら 3 つの要素のうち，主に機密性と完全性を保証するための技術であ

る．機密性のための「暗号」と，完全性のための「認証・署名」により，様々な脅威に対抗する．暗号技術の中でも基礎となるのは次の 6 つである．

■暗号 暗号とは，正当な送信者と受信者以外には，内容を秘匿する技術のことである．送信者は，メッセージ（平文）m に対して，暗号化鍵 k_E を用いて暗号文 c を生成する．一方，受信者は暗号文 c に対して，復号鍵 k_D を用いて平文 m を得る．以上の関係は，それぞれ

$$c = \mathsf{Enc}(k_E, m),$$
$$m = \mathsf{Dec}(k_D, c)$$

と与えられる．暗号アルゴリズムは，復号鍵 k_D なしで暗号文から平文を復元することが困難なように設計される．ここで，暗号化鍵 k_E と復号鍵 k_D が一致するものを共通鍵暗号方式，異なるものを公開鍵暗号方式とよぶ．

■鍵配送 鍵配送は，暗号化や復号で使用する鍵を安全に配送・共有するための技術である．例えば，共通鍵暗号方式では，送信者と受信者の間で共通の秘密鍵 k を共有する必要がある．鍵配送では，公開鍵暗号方式を利用することで，この問題を解決する．

■ハッシュ関数 ハッシュ関数とは，任意長の長さのメッセージから，固定長の長さの値（ハッシュ値）を返す関数である．ハッシュ関数は認証・署名に用いられ，特に (i) 一方向性（原像計算困難性），(ii) 第 2 原像計算困難性，(iii) 衝突困難性の 3 つの性質を満たすように設計される．(i) 一方向性とは，ハッシュ値が与えられたときに，元のメッセージを求めることが困難であることである．逆方向の計算が困難な関数のことを一般に一方向性関数とよぶ．(ii) 第 2 原像計算困難性とは，あるメッセージとそのハッシュ値が与えられたときに，同一のハッシュ値となる異なるメッセージを計算することが困難であることである．(iii) 衝突困難性とは，同じハッシュ値を持つ，2 つの異なるメッセージを求めることが困難であることである．これらの条件を満たすハッシュ関数のことを，暗号学的ハッシュ関数という．これらの性質から，あるハッシュ値を作り出せるのは，事実上元のメッセージを知っている場合に限られる．よって，通信の前後でハッシュ値を比較することで，完全性の検証が行える．

■メッセージ認証コード メッセージ認証コードとは，送られてきたデータが改ざんされていないことを検証できる技術のことである．加えて，期待された通信相手から送信されたことも確認する．メッセージ認証コードでは，送信者はまずメッセージと秘密鍵を認証子生成アルゴリズムに入力し，認証子（タグ）を計算する．次に，送信者はメッセージとタグを受信者に送信する．受信者は，受け取ったメッセージと秘密鍵からタグを生成し，受け取ったタグと一致するかを確認する．一致すれば，送信メッセージは正しい送信者か

ら送られており，改ざんもされていないことを確認できる．

■デジタル署名　デジタル署名とは，電子的に契約の際の捺印を実現する技術である．ユーザ認証とデータ認証を同時に実現する．送信者が署名を付けたメッセージを受信者に送信したいとする．まず送信者は，署名を行うための署名鍵と，署名の正当性を確認するための検証鍵を生成する．次に，送信者は，2 つの鍵とメッセージから署名を生成し，メッセージと署名を受信者に送信する．受信者は，検証鍵とメッセージ，署名から，署名の正しさを検証する．デジタル署名は，(i) 正当性と (ii) 偽造不可能性を満たさなければならない．(i) 正当性とは，正しい署名鍵から生成された署名であれば，その署名は正しく検証されるということである．(ii) 偽造不可能性とは，メッセージと署名のペアを偽造されたときに，検証時の検証式が成り立たないことである．

■擬似乱数生成器　擬似乱数生成器（PRNG: Pseudo Random Number Generator）は，統計的に偏りのない乱数系列を出力するアルゴリズムのことである．暗号技術では，秘密鍵の生成など至るところで用いられる．暗号における PRNG が満たすべき性質として，(i) 無作為性，(ii) 予測不可能性，(iii) 再現不可能性がある．(i) 無作為性とは，生成された乱数に統計的な偏りがないことである．(ii) 予測不可能性とは，過去の数列から次の数が予測できないことである．(iii) 再現不可能性とは，同じ数列を再現できないことである．

2.2.2　暗号アルゴリズム

暗号アルゴリズムは，暗号技術における最も重要な構成要素である．現在広く使われている暗号方式は，大きく分けて共通鍵暗号方式と公開鍵暗号方式の 2 つである．以下では各方式について述べる．

■共通鍵暗号方式　共通鍵暗号方式とは，暗号化と復号で同一の鍵（秘密鍵）を使用する暗号方式のことである．共通鍵暗号方式は，公開鍵暗号方式と比較して，一般に高速に暗号化・復号ができるという利点がある．一方で，共通鍵暗号方式を用いて安全に通信を行うためには，送信者と受信者の間で事前に秘密鍵を共有しておく必要がある．共通鍵暗号方式が満たすべき性質として，(i) 正当性と (ii) 秘匿性がある．(i) 正当性とは，暗号文を復号すると，元の平文が得られることである．(ii) 秘匿性とは，暗号文から元の平文の情報が全く得られないことである．共通鍵暗号方式の実現方法として，主にストリーム暗号とブロック暗号がある．ストリーム暗号は，秘密鍵と擬似乱数生成器を使用して得られる疑似乱数系列と．平文の間の排他的論理和により，暗号文を計算する暗号である．ブロック暗号は，平文を固定長のブロックに分割し，各ブロックに対して暗号化処理を行う暗号である．

■公開鍵暗号方式 公開鍵暗号方式とは，暗号化と復号で異なる鍵を使用する方式のことである．送信者は，相手の公開鍵で平文を暗号化し，暗号文を生成する．受信者は，受信した暗号文を自身の秘密鍵を使用して復号する．公開鍵暗号方式では，公開鍵と秘密鍵の間の鍵の非対称性により，事前に鍵の共有が必要ないという利点がある．一方で，共通鍵暗号方式と比べて，公開鍵暗号方式は単位データあたりの処理に時間がかかるという欠点がある．そのため，公開鍵暗号方式はメッセージの送信ではなく，共通鍵暗号方式で使用する共通鍵の配送手段として用いられる．公開鍵暗号方式も，共通鍵暗号方式と同様に，(i) 正当性と (ii) 秘匿性を満たさなければならない．特に公開鍵暗号方式における秘匿性は，攻撃モデルと解読モデルによって細かな分類分けが存在する．攻撃モデルは攻撃者の攻撃能力を表し，解読モデルは，暗号の解読難易度を意味する．したがって，より高い攻撃能力を有する攻撃者に対しても，高い解読難易度を有することが望ましい．

攻撃モデルには，選択平文攻撃 (CPA: Chosen Plaintext Attack)，選択暗号文攻撃 (CCA: Chosen Ciphertext Attack)，適応的選択暗号文攻撃 (CCA2: Adaptive Chosen Ciphertext Attack) がある [12]．選択平文攻撃は，攻撃者が任意の平文の暗号文を得られるとするモデルである．選択暗号文攻撃は，予め選択した暗号文と，それに対応する平文を利用する攻撃モデルである．適応的選択暗号文攻撃は，選択した暗号文と，それに対応する平文を利用する攻撃モデルである．適応的選択暗号文攻撃は，暗号文の復号結果を返す復号オラクルに何度でもアクセスできる状況を想定しており，選択暗号文攻撃よりも強力な攻撃である．解読モデルには，一方向性 (OW: Onewayness) と識別不可能性 (IND: Indistinguishability)，頑強性 (NM: Non-Malleability) がある．一方向性とは，秘密鍵を用いずに復号を行うことが困難なことである．識別不可能性とは，２つの平文のどちらかの暗号文が与えられたときに，その暗号文がどちらの平分に対応するものかを識別できないことである．頑強性は，ある暗号文が与えられたときに，攻撃者にとって意味のある別の暗号文が得られないことである．攻撃モデルでは，適応的選択暗号文攻撃が最も強力であり，解読モデルでは，頑強性が最も高い安全性レベルとなる．したがって，公開鍵暗号は，適応的選択暗号文攻撃に対して頑強性を有する（NM-CCA2 安全）ことが望ましい．実際には，IND-CCA2 安全であれば，NM-CCA2 安全であることが知られているため，公開鍵暗号を設計する際には IND-CCA2 安全を満たすことが目標となる．

2.2.3 暗号アルゴリズムの具体例

本節では，共通鍵暗号と公開鍵暗号の代表的なアルゴリズムである AES および ECC について述べる．

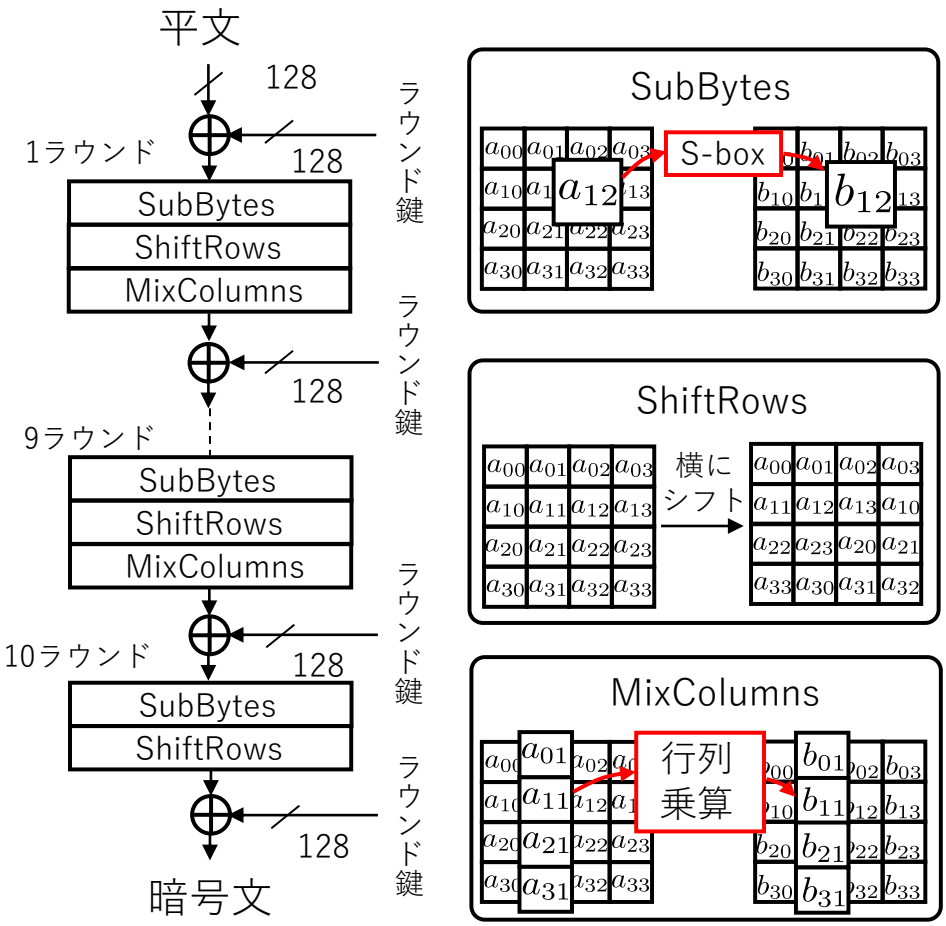


図 2.1: AES のアルゴリズム

■AES　AES は，ISO/IEC 18033-3 によって規定された共通鍵暗号であり，SPN (Substitution-Permutation Network) 型の 128 ビットブロック暗号アルゴリズムである．鍵長は 128, 196，256 ビットをサポートする．以下では鍵長 128 ビットの AES アルゴリズムについて述べる．図 2.1 に AES アルゴリズムを示す．AES アルゴリズムでは，128 ビットのデータを，1 つの要素が 1 バイトからなる 4×4 の行列と解釈する．AES の暗号化は，ラウンド関数と呼ばれるデータの攪拌処理を 10 回適用することで行う．AES のラウンド処理は，SubBytes, ShiftRows, MixColumns, AddRoundKey の 4 つからなる．これらの演算はすべて既約多項式を $\alpha^8 + \alpha^4 + \alpha^3 + \alpha + 1$ とする有限体 $\mathbb{F}_{2^8}$ 上で行われる．ここで，α は，不定元を表す．SubBytes は 8 ビット入出力の非線形変換テーブル S-box による変換（有限体上の逆元演算を含む），ShiftRows はバイト単位の循環シフト，MixColumns はデータの行列に対する有限体上の行列演算，AddRoundKey は XOR によるラウンド鍵の加算である．各ラウンドで使用されるラウンド鍵の生成処理は，鍵スケジュールと呼ばれ，XOR，循環シフト，定数 Rcon_i，および S-box から構成される．AES

の復号は，暗号化と逆の手順を踏むことで行われる．AddRoundKey は XOR であるため暗号化と共通であり，MixColumns は逆行列を乗算する invMixColumns，ShiftRows は逆方向の循環シフトをする invShiftRows，SubBytes は逆変換である invSubBytes が代わりに用いられる．

■ECC　楕円曲線暗号（ECC: Elliptic Curve Cryptography）は，Koblitz と Miller によってそれぞれ独立に発明された公開鍵暗号である [13]．ECC は，楕円曲線に基づく暗号の総称である．ECC として，楕円曲線上の ElGamal 暗号，楕円曲線上の Diffie-Hellman の鍵共有（ECDH: Elliptic Curve Diffie-Hellman key exchange）や ID ベース暗号などがある．2048 ビットの公開鍵である ElGamal 暗号と 256 ビットの公開鍵である楕円曲線上の ElGamal 暗号は同等の安全性を持つことが知られており，扱うビット数が小さく計算速度も早いことから，RSA 暗号から ECC の置き換えが進んでいる．以下では ECC として，ECDH について述べる．

ECDH は，楕円曲線を利用した鍵共有アルゴリズムの 1 つである．プロトコルについて説明するために，アリスとボブの 2 者間で鍵共有を行う状況を考える．p を素数とし，$\mathbb{F}_p$ を元の個数が p の有限体とする．アリスとボブは事前に使用する楕円曲線 E を決めておき，有限体 $\mathbb{F}_p$ 上の楕円曲線 $E(\mathbb{F}_p)$ の 1 点 $Q \in E(\mathbb{F}_p)$ を公開する．アリスとボブは，乱数 $k_a \in \mathbb{F}_p$ と $k_b \in \mathbb{F}_p$ をそれぞれ決め，公開鍵 k_aQ と k_bQ を求めておく．アリスはボブに k_aQ，ボブはアリスに k_bQ を送信する．そして，アリスは受信した公開鍵に自身の秘密鍵 k_a を乗算し，k_ak_bQ を求める．同様にボブも k_bk_aQ を計算する．楕円曲線上の点に対するスカラー倍は交換法則が成立するため，$k_ak_bQ = k_bk_aQ$ となり，両者の情報は一致する．アリスとボブ以外の第三者は，秘密鍵 k_a と k_b を知らないため共有鍵 k_ak_bQ を知ることはできない．これが ECDH のプロトコルである．ECC 上のディフィーヘルマン問題の困難性から，公開情報 Q と k_aQ，もしくは k_bQ から，秘密情報 k_a と k_b を求めることはできない [13]．これにより盗聴の危険性が排除される．

2.2.4　暗号アルゴリズムの実装形態

本論文では，暗号技術を実装したハードウェアやソフトウェアを暗号モジュールと呼ぶ．暗号アルゴリズムの実装形態には，ソフトウェア実装とハードウェア実装の 2 つがある．ソフトウェア実装は，汎用プロセッサ上で動作するプログラムとして暗号機能を実装する．ハードウェア実装は，ASIC（Application Specific Integrated Circuit）や FPGA（Field Programmable Gate Array）上の回路として暗号機能を実現したもののことである．

ソフトウェア実装は，プログラムの書き換えによって暗号機能の修正や拡張ができるた

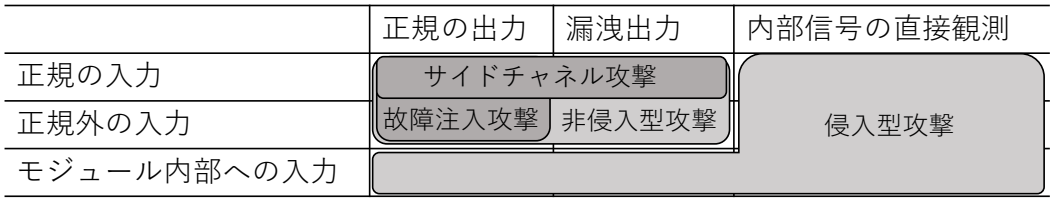


図 2.2: 攻撃の分類（文献 [2] を参考に作成）

め柔軟性がある．また，暗号機能を動作させる CPU やメモリには，既製品を用いることができるため，コスト面において優位性がある．ただしソフトウェア実装では，既存の命令セットを使用して暗号機能を実装しなければならないため，速度や消費電力の面では，ハードウェア実装に比べて不利である．一方，ハードウェア実装では，暗号機能に適したアーキテクチャを採用できるため，高速，小型かつ低消費電力で暗号機能を実現できる．また，ハードウェア実装は，物理攻撃への対策を回路のアーキテクチャのレベルで組み込むことができるため，汎用 CPU 上で動作するソフトウェア実装に比べて，高い安全性を実現できる．

2.2.5　暗号モジュールへの物理攻撃

2.2.2 節で述べたとおり，暗号アルゴリズムには安全性のために満たすべき性質が定められている．これらの性質は，送信者と受信者が盗聴の危険性がある通信路上で，通信を行うモデルに基づき定められている．すなわち，従来のモデルでは，攻撃者は暗号アルゴリズムの入出力にしかアクセスできないと仮定されている．しかし，物理攻撃では，攻撃者はチップの開封を伴う内部信号の観測や，消費電力や漏洩電磁波の測定により，暗号アルゴリズムの解読に必要な情報を入手する恐れがある．これは，攻撃者が通信路に流れる情報以外の情報（暗号計算の中間値など）へアクセスすることと等価であり，従来の通信路モデルでは想定されない状況である．したがって，暗号モジュールの実装では，暗号アルゴリズムの従来の満たすべき性質に加えて，物理攻撃への対策を別途実施する必要がある．

図 2.2 に暗号モジュールへの物理攻撃の分類を示す．暗号モジュールへの物理攻撃は，対象の暗号モジュールの変形を伴うか否かによって，侵入型と非侵入型の 2 つに大別される [2]．侵入型攻撃は，暗号モジュールへ狭義のタンパー手段（切る，削る，孔を開ける，溶かす，分解するなど）により，モジュールの変形を伴う直接的な手段を通して，モジュール内の秘密情報を窃取する攻撃のことである．代表的な侵入型攻撃として，回路内のバスやレコード，メモリの値の，マイクロプロービングによる読み出しが挙げられる．暗号アルゴリズムを実装した回路に直接アクセスするため高い攻撃能力を有するが，チッ

プの開封には専門知識や高価な機器が必要となる．また，侵入型攻撃に対する対策・検知は，古くから研究されており，暗号モジュールのワンチップ化や，プローブの接近を検知するための金属メッシュセンサなどを用いるなど，様々な対策手法が存在する．したがって，物理攻撃を意識した製品へ，侵入型攻撃を検知されずに行うことは容易ではない [3]．

一方，非侵入型攻撃は，暗号モジュールの変形を伴うことなく，正規もしくは非正規の入出力を利用して行う攻撃のことである．非侵入型攻撃は，侵入型攻撃と比べて痕跡が残らず検知が困難であり，攻撃方法によっては安価な機器で実行可能なため，現実的な脅威として注目されている．

非侵入型攻撃は，サイドチャネル攻撃と故障注入（フォルト）攻撃に大別できる．フォルト攻撃は，暗号処理の実行中に正規ではない入力を印加し，誤作動による計算誤りを誘発させ，その誤りパターンから秘密鍵を解析する攻撃である．正規ではない入力として，不正なクロック信号や入力電圧の印加，チップへのレーザー照射などが挙げられる．一方で，サイドチャネル攻撃は，暗号演算において副次的に発生する物理的変量（消費電力，漏洩電磁波，処理時間など）を計測し，統計処理を行うことで秘密情報の抽出を行う攻撃である．フォルト攻撃と比較して，サイドチャネル攻撃は実装による検知が困難であり，PC とオシロスコープのような比較的安価な機器のみで実行可能なため，高い注目を集めている．特に近年では，深層学習技術を利用した，新たなタイプのサイドチャネル攻撃（DL-SCA）が報告されるなど，攻撃自体の高度化・最適化に関する研究が盛んに行われている．これらの情勢から，本稿ではサイドチャネル攻撃を中心に扱う．

2.2.6 暗号モジュールの安全性評価

暗号モジュールは，様々な暗号プロトコルを実現するための基本プリミティブであるため，実装の脆弱性や攻撃の危険性を可能な限り排除されることが望まれる．ただし，暗号モジュールの脆弱性や安全性は，暗号モジュールの実装形態や運用形態に強く依存するため，その安全性評価にあたっては，暗号モジュールそのものへの実験的な安全性評価試験が必要となる．暗号モジュールの試験及び認証制度として，米国・カナダの CMVP（Cryptographic Module Validation Program）[14] や，IPA が運用する JCMVP(Japan CMVP)[8] がある．これらの制度では，暗号モジュールのセキュリティ要求に関する国際標準である ISO/IEC 19790[15] や ISO/IEC 24759[16] に従い，攻撃の知識及び実験環境を持つテストラボにて，暗号モジュールの試験が行われる．テストラボは，それらの試験の結果をもとに報告書を作成し，認証機関が報告書の内容を受けて認定証を発行する．

IPA が運用する JCMVP で行う試験は，暗号モジュールへの物理攻撃だけでなく，暗号が正確に実装されているか確認する暗号アルゴリズム実装試験や，文書・ソースコードレビューまで含まれる．これは，JCMVP の目的が，(1) 暗号の実装が正しく，(2) それが

ただしく実行され，(3) 重要情報が適切に保護されていることを担保することであるためである．JCMVP では，これら 3 つの要素を担保するために，物理攻撃に限らない様々な試験を実施することで，暗号モジュールの安全性を保証する．

しかし，JCMVP の行う暗号モジュール試験も必ずしも万全とは言えない．例えば，JCMVP の実施する暗号アルゴリズム実装試験では，国際標準 ISO/IEC 18367[17] に基づき，暗号モジュールに様々な入力を印加し，その出力の正しさの確認が行われる．ただし，暗号モジュールでは，非常に少数であっても，故障を引き起こす入力ベクトルが存在すると，秘密鍵漏洩の危険性が存在する [18]．また，近年では，ハードウェアトロイ (HT: Hardware Trojan) と呼ばれるバックドアを，設計製造時に挿入される危険性も指摘されている [19]．HT は，通常，極めて少数の入力ベクトルでのみトリガーされるため，テストベクトルを印加し，その応答を確認するような試験では，検知することは難しい．加えて，JCMVP の実施するサイドチャネル攻撃の耐性試験にも課題がある．同試験では，国際標準 ISO/IEC 17825[20] に従い，t 検定に基づく相関ベースの漏洩検査が実施される．ただし，すでに指摘したとおり，昨今では深層学習を用いることで，単純な相関関係にとどまらない漏洩情報を，攻撃に用いることが可能となっている．したがって，従来の t 検定による漏洩評価では十分とは言えない．事実近年では，深層学習を用いた新たな漏洩評価手法の検討も行われている [21]．

2.3　サイドチャネル攻撃に関する基礎的考察

本節では，サイドチャネル攻撃について述べる．以下では，まず，サイドチャネル攻撃の原理について述べる．続いて，サイドチャネル攻撃の分類ついて述べ，その対策手法について説明する．最後に，近年報告された新たな種類のサイドチャネル攻撃である，DL-SCA（Deep-Learning based Side-Channel-Attack）について述べる．

2.3.1　情報漏えいの機序

本節では，サイドチャネル攻撃の元となる原理について述べる．すでに述べたとおり，サイドチャネル攻撃は暗号モジュールの消費する電力や漏洩電磁波を計測し，解析することで実現される攻撃である．これらの消費電力および漏洩電磁波の発生源は，回路を構成する CMOS（Complementary Metal-Oxide-Semiconductor）セルである [22]．すなわち，CMOS セルの振る舞い（スイッチ）が暗号の演算内容と関係を持つために，サイドチャネル情報の解析によって秘密情報の窃取が可能となる．したがって，サイドチャネル攻撃の原理を理解するためには，CMOS セルの挙動を考える必要がある．

図 2.3 に，最も簡単な CMOS セルの例として，CMOS インバータを示す．CMOS イン

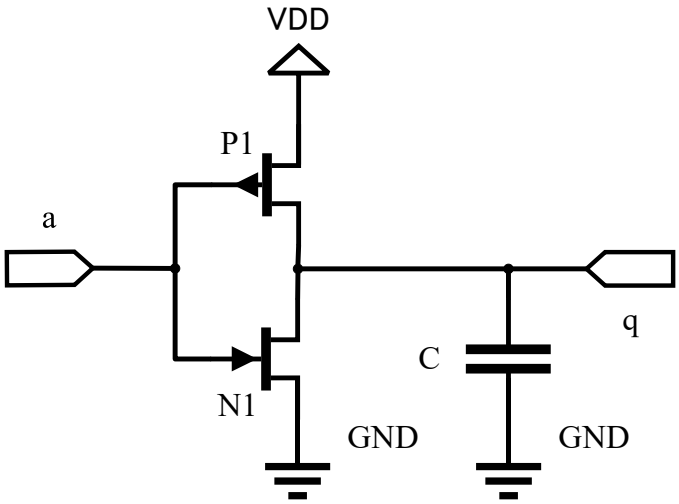


図 2.3: CMOS インバータ

バータが消費する電力は，大きく分けて静的電力と動的電力がある．静的電力は，CMOS インバータが常に使用している一定の電力のことであり，動的電力はインバータを構成するトランジスタがスイッチすることで生じる電力のことである．以下では，サイドチャネル攻撃に関係がある動的電力に注目する．

図中の，端子 a が入力値であり，端子 q が出力値である．端子 a の信号が $0 \rightarrow 1$ に変化したとき，V_{DD} からトランジスタ P1 を通ってコンデンサ C へ電流が流れる．そして，コンデンサ C が完全に充電されると，端子 q の電位が V_{DD} と一致し，出力値が 1 となる．同様に，端子 a の信号が立ち下がる（$1 \rightarrow 0$）と，コンデンサ C からトランジスタ N1 を通って，GND へ電流が流れる．そして，コンデンサが完全に放電すると端子 q の信号が 0 となる．また，端子 a の立ち上がりや立ち下がりの際には，P1 と N1 の両方のトランジスタのゲートの入力が中間電位になり，同時に駆動するタイミングがある．このとき，直接 V_{DD} から GND への電流が生じる．インバータで発生する動的電力は，上述したコンデンサ C の充放電と，V_{DD} から GND への貫通電流に由来する．したがって，インバータの入力値の変化に応じて，消費電力が変化する．

サイドチャネル攻撃では，これまでに述べたような，消費電力（電流）と計算機内の論理値の関係を利用して暗号解読を行う．例えば AES のハードウェア実装へのサイドチャネル攻撃を考える．AES は，入力平文に対してラウンド関数を繰り返し適用して，暗号文を計算する．AES のラウンド関数の適用回数は，セキュリティレベルによって決定される．ラウンド関数をより多く適用するほど，攻撃者が暗号文から暗号解読を行う際に必要な計算量が増大する．言い換えれば，ラウンド関数の適用回数が不十分であれば，AES は安全ではない．AES へのサイドチャネル攻撃では，消費電力から暗号計算の中間値の

解析を行う．上述の通り，CMOS セルの消費電力は，入力信号の変化に比例する．したがって，消費電力量から AES の中間値が間接的に推定できる．中間値が漏洩した場合，ラウンド関数を本来よりも少ない回数だけ適用したときの出力値を求めることと同じ効果が得られる．この結果，本来想定されていた鍵の探索空間が大幅に削減され，現実的な時間で全探索が可能となる．これがサイドチャネル攻撃の原理である．

サイドチャネル攻撃を行うには，暗号計算の中間値から消費電力を予測する必要がある．この消費電力を予測するためのモデルのことを，電力モデルと呼ぶ．通常，電力モデルは，暗号モジュールがハードウェア実装か，ソフトウェア実装かによって異なる．ハードウェア実装の場合は，予測した中間値の前のクロックと，次のクロックの間のハミング距離（HD: Hamming Distance）が用いられる．これは，先程の CMOS インバータの例と同様に，消費電力が中間値のビットフリップの量に比例すると考えられるためである．一方で，ソフトウェア実装の場合は，クロック間の影響は考えずに，単に現在のクロックの中間値の値のハミング重み（HW: Hamming Weight）を使用する．これは CPU で使われるプリチャージバスの影響のためである．

2.3.2　サイドチャネル攻撃の種類

本節では，サイドチャネル攻撃の種類および分類分けについて説明する．まず，攻撃に使用するサイドチャネル情報の種類として，処理時間と，消費電力，電磁波の 3 つがある．処理時間の情報を用いる攻撃をタイミング攻撃，消費電力を用いる攻撃を電力解析，電磁波を用いる攻撃を電磁波解析という [22]．ただし，電力・電磁波解析は計測法の違いのみであり，計測後のデータ解析法は同じである．解析に用いる波形の枚数に応じて，更に分類分けがなされ，一度の観測波形のみを使用するものを単純電力解析（SPA: Simple Power Analysis）や単純電磁波解析（SEMA: Simple Electro-Magnetic Analysis）とよぶ [22]．複数回の計測に基づくものには，様々な種類が存在し，プロファイリングの有無によってノンプロファイリング攻撃と，プロファイリング攻撃に分けられる．ここで，プロファイリングとは，攻撃対象と同一の型番のモジュールから，攻撃者が事前にサイドチャネル情報と秘密情報の間の関係性を抽出することである．プロファインリング攻撃では，攻撃者は事前に抽出したプロファイリング情報を利用して，攻撃対象デバイスから効率的に秘密情報の窃取を行う．これは IoT デバイスのように，同一のデバイスが大量に存在し，そのうちの 1 つを攻撃者が容易に入手できるようなシナリオに対応する．一方で，ノンプロファイリング攻撃では，そのようなプロファインリング情報の存在を仮定しない．したがって，ノンプロファイリングよりもプロファイリング攻撃のほうが，より強力な攻撃である．代表的なノンプロファイリング攻撃には差分電力解析（DPA: Differential Power Analysis），相関電力解析（CPA: Correlation Power Analysis），相互情報量解析

（MIA: Mutual Information Analysis）がある．DPA や CPA では，前節で述べた HW や HD モデルを利用して，中間値から消費電力を予測し，サイドチャネル情報の統計処理によって秘密鍵の推定を行う．対して，MIA は，電力モデルを仮定せずに，サイドチャネル情報と中間値の間の相互情報量が最大となるような秘密情報を推定することで攻撃を行う．一方，代表的なプロファイリング攻撃にはテンプレート攻撃と，DL-SCA がある．テンプレート攻撃は，サイドチャネル波形が中間値の HW/HD を平均としたガウス分布に従っていると仮定し，最尤推定により秘密情報の推定を行う．DL-SCA は，波形と中間値の間の関係を深層学習モデルによってモデル化して，サイドチャネル情報の解析を行う攻撃である．

以上に示した攻撃では，DL-SCA が最も強力な攻撃であることが知られており，解析に必要な波形数が，テンプレート攻撃と比較して数倍程度少なくて済むことが報告されている．本稿では，最も強力な攻撃である DL-SCA を中心に扱う．

2.3.3 サイドチャネル攻撃の対策

サイドチャネル攻撃の実装レベルの対策方法は，様々なものが提案されているが，大まかにハイディングとマスキングに分けられる [22]．どちらも基本アイデアは，中間値と消費電力の間の関係性を断つことである．ただし，その方向性に違いがある．

ハイディングは，どのような入力が与えられたときにも，消費電力が計算処理の内容によって変化しないように回路を設計することで対策を行う．例えば，代表的な対策手法に SABL（Sense Amplifier Based Logic）[23]，RSL（Random Switching Logic）[24]，DRSL（Dual-rail Random Switching Logic）[25]，TDPL（Three-phase Dual-rail Pre-charge Logic）[26] などがある．ただしこれらの対策では，回路設計に用いられるセルライブラリを根本から設計し直す必要があるため，ASIC では製造できない．代わりに，標準セルライブラリのみを用いて，ハイディングを実現する方法として，WDDL（Wave Dynamic Differential Logic）[23] や MDPL（Masked Dual-Rail Pre-charge Logic）[27] などがある．これらのハイディング手法は，暗号モジュールの消費電力特性を変えて，中間値と消費電力の間の依存関係を取り除くことで対策する．したがって，ハイディングによる対策では，暗号モジュールの実装方法によって，適用が困難な場合がある．また，ハイディングは証明可能安全な対策ではない．すなわち，ハイディングによる対策は，サイドチャネル攻撃耐性を理論的に保証できない．これら 2 つの問題を解決するための，別のアプローチとしてマスキングがある．

マスキングは，秘密分散法に基づく対策手法であり，暗号化処理の中間値をシェアと呼ばれる複数の独立な中間値に分解することで対策を行う．代表的なマスキング対策であるブーリアンマスキングでは，分解に XOR が用いられる．例えばある中間値 z を d

個のシェア $s_1, s_2, \ldots, s_d$ を使用して分解する場合，$z = s_1 \oplus s_2 \oplus \cdots \oplus s_d$ とする．d 個のシェアが，独立かつ一様乱数に従って生成される場合，攻撃者が中間値 z を知るためには，すべてのシェア $s_1, \ldots, s_d$ の値を正確にサイドチャネル波形から抽出する必要がある．サイドチャネル情報に含まれるノイズの量によって，シェア 1 つあたりにおける，正確な値を推定できる確率には限界が存在するため，シェアの数が増えることにより攻撃が成功する確率は指数的に減少する [28]．代表的なマスキングスキームに，TI（Threshold Implementation）や DOM（Domain Oriented Masking）などがある．特に，AES のマスキングハードウェア実装に関しては，非常に多くの先行研究がある [29, 30, 31, 32, 33, 34, 35, 4, 36]．

2.3.4　深層学習に基づくサイドチャネル攻撃

近年，暗号モジュールに対する深層学習 (DL: Deep Learning) を用いたサイドチャネル攻撃 (DL-SCA: DL-based Side-Channel Attack) が高い注目を集めている [37, 38, 39, 40]．DL-SCA はプロファイリング型のサイドチャネル攻撃の一種であり，プロファイリングフェーズと攻撃フェーズから構成される．プロファイリングフェーズでは，攻撃者は自身が所有する攻撃対象と同じ型番のデバイスから予めサイドチャネル情報の取得を行い，DL を用いてデバイス固有の特徴をモデルとして抽出する．次に，攻撃フェーズではプロファイリングフェーズで抽出したモデルを利用して，攻撃対象デバイスのサイドチャネル情報から効率的に秘密鍵を推定する．DL-SCA では，従来のプロファイリング攻撃（テンプレート攻撃など）と異なり，攻撃対象デバイスのサイドチャネル情報に関する特別な知識や仮定を必要とせずに学習および攻撃を行うことができるという利点がある．ここで，特別な知識や仮定とは，暗号アルゴリズムの実装方法（SCA 対策の有無を含む）や，サイドチャネル情報が含むノイズの分布などを指す．それらはしばしば攻撃者にとって利用不可もしくは非現実的となる．事実，先行研究では，DL-SCA を用いることにより従来のテンプレート攻撃などと比較して手軽かつ効率的に AES の秘密鍵を抽出できることが示されている [9]．IoT などの次世代情報通信ネットワークシステムにおいては，プロファイリング攻撃が可能となる暗号デバイスの増加が見込まれることから，今後 DL-SCA の脅威を正確に把握する必要性が高まると予想される．以下では，AES を例として，DL-SCA の具体的な手順について説明する．

DL-SCA では，まず，プロファイリング用のデバイスを用いてサイドチャネル情報から深層学習モデルを学習させる．DL-SCA におけるモデルの入力はサイドチャネル情報（電力や電磁波の時間波形）であり，出力は秘密情報を表すクラスの推定確率となる．特に，DL-SCA に用いる NN モデルとしては全結合層（MLP: Multilayer Perceptron）と，畳み込みニューラルネットワーク (CNN: Convolutional Neural Network) が学習効率お

よび攻撃成功確率の観点から有効なことが知られている [39]．以降の例では，AES の第一ラウンド S-box 出力の HW をサイドチャネル情報から予測する例を考える．DL-SCA における NN の学習と推論は，それぞれプロファイリングフェーズと攻撃フェーズに対応する．

プロファイリングフェーズはプロファイリング用デバイスのサイドチャネル情報を教師データとした教師あり学習であり，教師データ（訓練用データ）は次に示す集合として定義される．

$$\mathcal{D}_P = \{(K_i, M_i, \boldsymbol{X}_i) \mid i = 1, 2, \ldots, n_P\}. \tag{2.1}$$

ここで，K_i と M_i はそれぞれ i 番目の計算時に用いられた攻撃対象バイト[*1]の秘密鍵と平文（の確率変数）であり，$\boldsymbol{X}_i$ は i 番目の計算時に取得されたサンプル数 n_t 点からなるサイドチャネル情報（の確率変数）である．ここで，n_P はプロファイリングフェーズで使用した波形数を表す．ここで，K_i と M_i を用いて攻撃対象となる中間値（S-box の出力）を計算し，サイドチャネル情報から中間値（の HW）を予測する NN を学習させる．ここで，予測するラベルの確率変数を $Z_i = \psi(K_i, M_i)$ とする．ただし，ψ は鍵と平文から攻撃対象の中間値（の HW）を返す関数である．学習フェーズにおける目的は，教師データに対するモデルのフィッティングにより，波形とラベルの間の真の確率分布 $q_{Z|\boldsymbol{X}}$ をモデルによって模倣することである．プロファイリングフェーズでは，真の確率分布を模倣するために次式で定義される CE を最小化するモデルパラメータ θ を求める．

$$\begin{aligned} \mathrm{CE}(p) &= -\mathbb{E} \log p(Z \mid \boldsymbol{X}; \theta) \\ &= -\int \sum_z q_{Z,\boldsymbol{X}}(z, \boldsymbol{x}) \log p_{Z|\boldsymbol{X}}(z \mid \boldsymbol{x}; \theta) d\boldsymbol{x}. \end{aligned} \tag{2.2}$$

ここで，$p_{Z|\boldsymbol{X}}(Z \mid \boldsymbol{X}; \theta)$ はモデルパラメータ θ が与えられた際のモデルの確率分布を表す．式 (2.2) の $\mathrm{CE}(p)$ は $p = q$ となるモデルパラメータ θ のときに限り最小値をとる[*2]．しかし，式 (2.2) は未知である真の確率分布 q に関して平均を取る必要があるため，現実的には計算不可能である．そこで，有限のサンプルを用いて，CE（の最小化）を経験誤差（の最小化）で次のように近似する [41, 42]．

$$\mathrm{CE}(p) \approx \mathrm{NLL}(\theta) = \frac{1}{n_P} \sum_{i=1}^{n_P} -\log p_{Z|\boldsymbol{X}}(Z_i \mid \boldsymbol{X}_i; \theta). \tag{2.3}$$

ここで，NLL (Negative Log Likelihood) を負の対数尤度関数と呼ぶ．NLL は使用する学習データのサンプルが真の確率分布に従うときに限り，データ数を増加させることでク

[*1] 通常の（プロファイリング攻撃ではない）差分電力解析や相関電力解析と同様に，DL-SCA でも 1 バイトごとの解析を 16 バイト全てに適用することで秘密鍵全体を取得する．

[*2] そのようなモデルパラメータ θ は存在しない可能性があることに注意されたい．

ロスエントロピー関数に確率収束する．したがって，十分なデータ数があれば，式 (2.3) の最小化が正当化される．

次に攻撃フェーズでは，式 (2.3) の最小化によって決定されたモデルパラメータ $\hat{\theta}$ を用いて正解鍵の推定を行う．まず，式 (2.1) と同様に攻撃フェーズで使用するデータを $\mathcal{D}_A = \{(M_j, \boldsymbol{X}_j) \mid j = 1, \ldots, n_A\}$ とする．鍵の推測値 k におけるデータセット $\mathcal{D}_A$ に対する NLL を

$$\mathrm{NLL}_k(\hat{\theta}) = \frac{1}{n_A} \sum_{j=1}^{n_A} -\log p_{Z|\boldsymbol{X}}(\psi(k, M_j) \mid \boldsymbol{X}_j; \hat{\theta}) \tag{2.4}$$

とすると，$p \approx q$ が成り立つような適切な $\hat{\theta}$ が選択されていれば（すなわちモデルが適切に学習されていれば），正解鍵 k で式 (2.4) が最小値を取ることが期待される．そこで，式 (2.4) に基づいて全ての鍵候補に対して $\mathrm{NLL}_k(\hat{\theta})$ を計算し，負の対数尤度を最小とする鍵候補を正解の秘密鍵とする．

2.3.5　サイドチャネル攻撃の安全性評価

通常，暗号モジュールへのサイドチャネル攻撃の目的は，秘密鍵を推定することである．一般に，サイドチャネル攻撃では，取得されたサイドチャネル波形に含まれるノイズの影響により，秘密鍵の推定に複数の観測が必要となる．サイドチャネル攻撃における安全性は，秘密鍵の推定に必要な波形数で見積もられる．マスキングやハイディングは，攻撃が成功するために必要な波形数を増加させることを目的とした対策である．ここで，これらの対策は必ずしもサイドチャネル攻撃を根絶するものではないことに注意されたい．ハイディングやマスキングなどの対策は，消費電力と暗号モジュール内部の処理の関係性を弱めるものの，完全に無関係にすることはできない．演算内容と消費電力に少しでも関係性が存在する限り，サイドチャネル攻撃が成立する可能性はいつも残されている．

一方で，サイドチャネル攻撃に対する安全性評価を行う立場からすれば，安全性を担保できる処理数が算出できれば，プロトコル層での対策を実施することができる．例えば，ある暗号モジュールの秘密鍵の推定に少なくとも n 波形（暗号化処理）が必要であったとする．この場合，セキュリティシステムを構築する設計者は，n 回の暗号化処理を行う前に暗号モジュールを使用不可，もしくは鍵の更新（リキー）を実施すれば良い．ただし，暗号モジュールを使用不可，もしくはリキーを実施する場合，相応のコストが発生する．したがって，攻撃が成立するために必要な波形数は，極力大きいことが望ましい．また，プロトコル層での対策を行うためにも，安全性の下界となる波形数 n を正確に求めることが肝要である．

2.4 結び

本章では，暗号モジュールの脆弱性検知に関する基礎的考察を行った．まず，暗号技術全般について概要を述べ，代表的な暗号アルゴリズムである AES と ECC を概説した．次に，暗号アルゴリズムの実装形態について説明し，アルゴリズム自体の安全性に加えて物理攻撃に対する安全性も重要であることを述べた．その後，サイドチャネル攻撃について，概要，分類，対策法の順に述べた．さらに，本論文で取り扱う DL-SCA について，攻撃の手順を述べた．

第 3 章

暗号ハードウェアの等価性検証手法

3.1 はじめに

本章では，論理的な安全性評価として暗号ハードウェアの設計時における脆弱性検知手法について述べる．1 章で述べたとおり，暗号ハードウェアでは設計時に混入されたバグなどの脆弱性によって，秘密鍵が漏洩することが知られている [43, 44, 45]．また近年では，攻撃者が秘密鍵の漏洩を目的として，秘密裏にハードウェアトロイ（HT: Hardware Trojan）と呼ばれるバックドアを設計時に挿入する可能性も指摘されている．暗号ハードウェアはシステムの安全性や信頼性の要であると考えられることから，バグや HT などの脆弱性を検知することは極めて重要である．そこで本章では，これらのバグや HT の根絶を目的として，等価性検証に基づく形式的脆弱性検知手法を提案する．提案手法では，暗号ハードウェアのネットリストと設計仕様の等価性検証を行い，これらの間に差がある場合に設計データに脆弱性が含まれていると判定する．本章では，まず従来の等価性検証手法について述べ，次に本章全体で使用する記号を説明する．続いて暗号ハードウェアのデータパス（組合せ回路部）を対象として，提案する等価性検証手法について述べる．また，提案手法の応用として，HT が暗号回路に設計時に挿入された場合を想定した，検知手法の開発も行う．同検知手法では，提案したデータパス部への等価性検証を利用して，レジスタなどを含む暗号ハードウェア全体の等価性検証を実現する．

3.2 関連研究

本章では，代表的な共通鍵暗号である AES や公開鍵暗号である ECC などの主要な構成要素であるガロア体上の算術演算に着目する．組合せ回路に対する代表的な等価性検証手法として，二分決定グラフ（BDD: Binary Decision Diagram）や二分モーメントグラフ（Bnary Moment Diagram）などの決定グラフを用いたものがある [46, 47, 48]．これらの手法では，与えられたゲートレベルネットリストとその設計仕様を，論理的に等価な

決定グラフに変換し，これらの間の同型判定を行うことで等価性検証を行う．文献 [47] では，BMD を用いることで整数演算器の効率的な検証が可能なことが示されている．一方で，暗号で主に使用されるガロア体の算術演算回路は，BDD や BMD などの決定グラフを用いた手法では，入力オペランド長が，たかだか 16 ビット程度で検証が困難になることが指摘されている [7]．したがって，AES などの暗号回路では，入力ビット長が最低でも 128 ビット以上になることから，BDD や BMD では実用的な暗号回路の等価性検証を行うことは難しいと予想される．また，ガロア体を明示的に扱えるように決定グラフを拡張したものとして，MODD (Multiple Output Deicision Diagram) なども提案されているが，これらを用いても根本的な解決にはならないことが知られている [49]．決定グラフを用いる以外の方法として，等価性判定問題を充足可能性問題（SAT:Satisfiability）へ帰着させ，SAT ソルバを用いて解く方法が考えられるが，同手法でもたかだか 16 ビット程度のガロア体算術演算回路ですら検証が困難なことが知られている [7]．

比較的大規模なガロア体算術演算回路を検証するための代表的な手法が 2 つ存在する．1 つは，GF-ACG (Galois-Field Arithmetic Circuit Graph) と呼ばれる，数学的かつ形式的なガロア体算術演算回路のためのグラフ表現を用いる方法である [50, 51, 52]．同手法では，ガロア体算術の回路構造を階層的なグラフとして表現し，グラフの表す機能をガロア体上の方程式として表す．そして，回路の GF-ACG の最下層から順番に，各層の等価性判定を実施する．先行研究において，GF-ACG を用いることで 128 ビット AES データパスや，リードソロモン符号などの等価性検証が可能になることが示されている．一方で，GF-ACG の致命的な欠点として，検証対象となる回路が GF-ACG に基づくグラフ表現になっていなければ検証できないことが挙げられる．言い換えれば，階層構造を持たないフラットなゲートレベルネットリストを GF-ACG では検証できない．論理合成ツールの最適化中に意図せずバグが挿入される可能性や，HT がゲートレベルネットリストに挿入される危険性を考慮すると，GF-ACG は本論文で目的とする脆弱性検知方法として十分であるとは言えない．

もう一つの方法として，ゲートレベルネットリストに適用可能かつリファレンス回路を必要としないガロア体算術演算回路の形式的検証手法がある [53, 54, 55]．同手法では，回路機能検証をイデアル所属問題に帰着することで代数的に回路機能検証を行う．すなわち，設計仕様を表すガロア体方程式をネットリスト上の論理ゲートが表現する $\mathbb{F}_2$ 上の連立ガロア体方程式から導出できるかを判定することで回路機能を検証する．イデアル所属問題の評価にはしばしば非現実的な計算時間が必要となることが一般に知られているが，回路の機能検証におけるイデアル所属問題を効率的に解くための手法がいくつか報告されている．既存手法では，イデアル所属問題の評価を，(1) 検証対象回路を表現する連立代数方程式に対応するグレブナー基底の算出と (2) 設計仕様を表す方程式のグレブナー基底による簡約の 2 ステップにより実現する．これに対して，文献 [7] では，回路のトポロジ

カルな特徴を用いる RTTO (Reverse Topological Term Order) をもちいることで，(1) において最も計算時間が必要となるグレブナー基底の算出をほぼコスト無しで実現可能なことを示した．一方，(2) では，中間結果として与えられる多項式の項数が大きく増大し，簡約処理を行うための計算量およびメモリ使用量が大幅に増大する問題が生じていた．そこで，文献 [53] では，$\mathbb{F}_2$ 上のガロア体方程式を ZDD（Zero-suppressed binary Decision Diagram) を用いて表現することで簡約処理を通常の多項式表現に比べて高速に実行した例が示されている．本論文では，ゲートレベルネットリストにも適用可能な 2 つ目の手法を元に，脆弱性検知手法を開発する．

3.3　準備

本節では，本章全体で使用する記号の準備を行う．また，提案手法の基本的な構成要素である ZDD も説明する．

3.3.1　グレブナー基底に基づく形式検証

係数を $\mathbb{F}_2$ にもち，変数 $w_1, w_2, \ldots, w_l$ からなる多項式のなす環を $\mathbb{F}_2[w_1, w_2, \ldots, w_l]$ とする．変数 $w_1, w_2, \ldots, w_l$ は l 本の配線からなる回路のすべての信号線を意味する．$\mathbb{F}_2[w_1, w_2, \ldots, w_l]$ のすべての多項式 f は，係数 $c_1, c_2, \ldots, c_t \in \mathbb{F}_2$ と単項式 $X_1, X_2, \ldots, X_t$ を用いて，$f = c_1X_1 + c_2X_2 + \cdots + c_tX_t$ と表せる．ここで，各単項式は $w_1^{e_1}w_2^{e_2}\cdots w_l^{e_l}; (e_i \in \mathbb{Z}_+)$ とかけることに注意されたい．本章を通して，多項式の項順序は，全順序かつ整列順序関係である RTTO によって決定される．形式的には，RTTO は 2 つの変数（配線）w_k と w_i について，w_k が w_i よりもプライマリ出力に近い場合に $w_k > w_i$ とする．もし，2 つの配線の順序が回路のトポロジーから決定できない場合には，これらの順序はランダムとする．この変数の順序を課した上で，辞書式順序を適用して多項式を表現する．以降簡単のため，$k > i$ を満たす場合に，$w_k > w_i$ が成立するようにすべての変数（配線）の添字を定める．n と m をそれぞれプライマリ入力と出力の変数の数とする．添字の定め方から，$w_1, w_2, \ldots, w_n$ はプライマリ入力変数，$w_{l-m+1}, \ldots, w_l$ はプライマリ出力に対応する．RTTO の規則から，$\mathbb{F}_2[w_1, w_2, \ldots, w_i \ldots, w_l]$ のすべての多項式 f は，$X_1 > X_2 > \cdots > X_t$ を満たす単項式を用いて，$f = c_1X_1 + c_2X_2 + \cdots + c_tX_t$ と表せる．ここで t は多項式 f を構成する単項式の数である．ここで，f の先頭項，先頭単項式，先頭係数をそれぞれ $\mathrm{lt}(f) = c_1X_1$, $\mathrm{lm}(f) = X_1$, $\mathrm{lc}(f) = c_1$ とする．もし $f \in \mathbb{F}_2[w_1, w_2, \ldots, w_i \ldots, w_l]$ が 0 でないとき，$\mathrm{lc}(f) = 1$ のため $\mathrm{lt}(f) = \mathrm{lm}(f)$ が成立する．s 個の多項式 $f_1, f_2, \ldots, f_s$ が生成するイデアルを $\langle f_1, f_2, \ldots, f_s\rangle = \{f_1h_1 + \cdots + f_sh_s \mid h_1, \ldots, h_s \in \mathbb{F}_2[w_1, w_2, \ldots, w_i \ldots, w_l]\}$

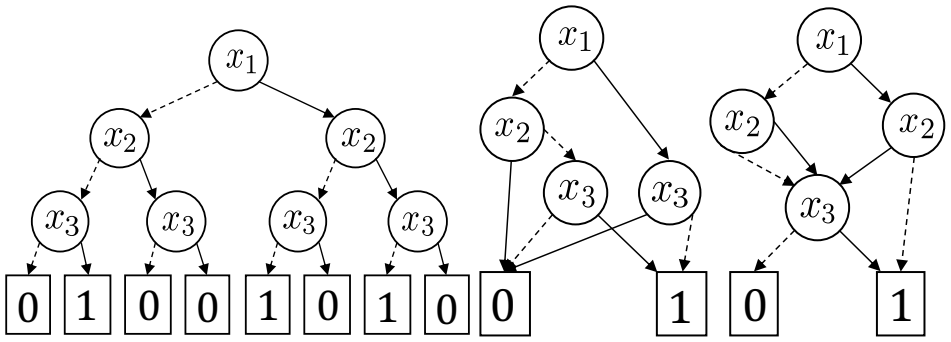
とする．各配線の信号線の値は二値であることから，すべての変数においてべき等律 $w_i^2 = w_i$ が成立することが望ましい．そこで，ブール多項式環 $\mathbb{B}(w_1, w_2, \ldots, w_i, \ldots, w_l)$ を，商環 $\mathbb{F}_2[w_1, w_2, \ldots, w_i, \ldots, w_l]/\langle w_1^2 + w_1, w_2^2 + w_2, \ldots, w_i^2 + w_i, \ldots, w_l^2 + w_l \rangle$ として定義する．すべての論理式は次式を用いてブール多項式環へ変換できる．

$$\begin{aligned} \neg w_i &\mapsto w_i + 1, \\ w_i \wedge w_j &\mapsto w_i w_j, \\ w_i \vee w_j &\mapsto w_i + w_j + w_i w_j, \\ w_i \oplus w_j &\mapsto w_i + w_j. \end{aligned}$$

f と g をブール多項式環 $\mathbb{B}(w_1, \ldots, w_l)$ の多項式とする．もし f の先頭単項式 $\mathrm{lm}(f)$ が $\mathrm{lm}(g)$ によって割り切れて，その剰余が r のとき，f は g で r へ簡約できるといい，$f \xrightarrow{g} r$ と表す．ここで，$r = f - [\mathrm{lt}(f)/\mathrm{lt}(g)]g$ である．加えて，$g \mid f$ は，f が g で割り切れることを表す．同様に，f を多項式集合 $\mathcal{G} = \{g_1, \ldots, g_s\}$ で割ることで，剰余 r を得ることができる．この多項式集合による剰余 r は，$\mathcal{G}$ がグレブナー基底のとき，一意になる．グレブナー基底の定義から，$\mathcal{G}$ の剰余 r が 0 のときかつそのときに限り f は，グレブナー基底 $\mathcal{G}$ が生成するイデアル $\mathcal{I}$ に含まれる．言い換えれば，剰余 r が 0 ではないとき，$f \notin \mathcal{I}$ である．グレブナー基底 $\mathcal{G}$ による多項式 f の簡約を $f \xrightarrow{\mathcal{G}} r$ と表記する．剰余 r の一意性は，回路の等価性判定に用いることができる．

命題 1. *(*組合せ回路の等価性検証 *[53])* n 入力，m 出力の回路 $\mathcal{C}$ が与えられたとき，すべてのゲートの入出力間の関係式から得られる多項式集合を $\mathcal{F} = \{f_{n+1}, \ldots, f_l\} \subset \mathbb{B}(w_1, \ldots, w_l)$ とする．ここで，$w_1, \ldots, w_l$ は $\mathcal{C}$ のすべての配線である．$\mathcal{F}$ の生成するイデアルを $\mathcal{I} = \langle \mathcal{F} \rangle$ とし，z_i $(1 \leq i \leq m)$ を回路 $\mathcal{C}$ の出力の 1 ビットとする．イデアル $\mathcal{I}$ のグレブナー基底 $\mathcal{G}$ による，z_i の簡約結果を r_i とする．すべての r_i と，設計仕様の出力が一致することと，回路 $\mathcal{C}$ と設計仕様の機能が等価（論理的に等価）であることは同値である．

定理 1 から，設計仕様とゲートレベルネットリストの間の等価性を検証できる．上記の定理を使用するにあたって，グレブナー基底 $\mathcal{G}$ を導出する必要がある．一般にグレブナー基底の導出には，ブッフバーガーのアルゴリズムが用いられるが，同アルゴリズムは最悪計算量が二重指数時間であることが知られており，回路検証に用いることは事実上不可能である．この問題に対し，文献 [7] は，RTTO を項順序として用いた場合，多項式集合 $\mathcal{F} = \{f_{n+1}, \ldots, f_l\}$ がそのままグレブナー基底となることを示した．これによりグレブナー基底の導出にブッフバーガーのアルゴリズムを用いる必要はない．よって以降，$\mathcal{G} = \mathcal{F}$ として議論をすすめる．



(a) 完全二分木　　(b) BDD　　(c) ZDD

図 3.1: $x_1x_2x_3 + x_1 + x_2x_3 + x_3$ のグラフ表現

3.3.2　ブール多項式環の ZDD 表現

ブール多項式環 $\mathbb{B}(x_1, \ldots, x_l)$ は，商環 $\mathbb{F}_2[x_1, \ldots, x_l]/\langle x_1^2 - x_1, x_2^2 - x_2, \ldots, x_l^2 - x_l\rangle$ と定義される．ここで，$x_1, \ldots, x_l$ はブール多項式環をなす変数を表す．ブール多項式環の元（多項式）をブール多項式と呼ぶ．従来，ブール多項式は単項式のリストとして表現されてきた．リストによる多項式の表現は非常にシンプルな一方で，多項式に出現しうる変数の数に対して指数的にリストサイズが大きくなる可能性がある．例えば，3 変数からなるブール多項式環 $\mathbb{B}(x_1, x_2, x_3)$ における最も大きい多項式は $x_1x_2x_3 + x_1x_2 + x_2x_3 + x_1x_3 + x_1 + x_2 + x_3 + 1$ であり，$2^3 = 8$ 項からなる．l 変数からなる多項式は最大で 2^l 項を含み，このような多項式は後の実験結果で示すとおり，HT を挿入された回路の等価性検証では出現する可能性がある．したがって，単純なリスト表現による多項式の取り扱いでは，形式検証には不十分である．リスト表現の代替として，多項式を $(x_1 + 1)(x_2 + 1)(x_3 + 1)$ のように因数分解した形で保持する方法がある．因数分解し各因子を保持する方法であれば，先程の最悪のケースであってもたかだか 3 つの因子を保持すればよいことになる．このような因数分解した形で保持するためのより一般的な表現方法として，BDD や ZDD のような決定グラフがある．

図 3.1(c) にブール多項式 $x_1x_2x_3 + x_1 + x_2x_3 + x_3$ の ZDD 表現を示す．また比較のために，同図 (a) に完全二分木，(b) に BDD も示した．各決定グラフは変数に対応する節点と，因子を表す二種類の枝，そして 0/1-終端節点から成る．図中の点線と実線はそれぞれ 0-枝と 1-枝を表す．決定グラフの根にあたるノードから終端節点までの各パスは単項式に対応し，それらの和が決定グラフ全体が表す多項式になる．

まず最初に最もシンプルな完全二分木について説明する．完全二分木ではあるノード x_i

から伸びる 1-枝が x_i の乗算，0-枝が (x_i+1) の乗算を表す．同様に，1 と 0-終端節点は，それぞれ 1 と 0 の乗算を表す．したがって，図 3.1(a) の完全二分木は，根から葉までのすべてのパスの表す多項式の和である $x_1x_2(x_3+1)+x_1(x_2+1)(x_3+1)+(x_1+1)(x_2+1)x_3$ に対応する．例えば，項 $x_1(x_2+1)(x_3+1)$ は，節点 x_1 から 1-枝，x_2 の 0-枝，x_3 の 0-枝を通って最後に 1-終端節点にたどり着くパスに対応する．完全二分木はわかりやすく，シンプルな構造を持つ一方で，木のサイズが出現する変数の数に対して指数的に増加するという問題がある．

BDD と ZDD は完全二分木から冗長な節点を削除することで得られ，ブール多項式をコンパクトに表現できる．ZDD の削除規則では，すべての同型なサブグラフを一つにまとめ，0-終端節点を指す 1-枝を持つ節点を削除する．また，ZDD は完全二分木と異なり 0-枝が定数 1 を乗算することを表す．同規則を完全二分木に適用することで，任意のブール多項式の ZDD を得ることができる．加えて，ZDD は BDD と同様に Apply アルゴリズムを用いることで，2 つの ZDD 同士の論理演算後の ZDD を得ることができる [56]．これにより完全二分木を経ずに ZDD を構築できる．ZDD は充足可能性条件を陽に表さないが，ブール多項式に含まれる単項式のみを表現することで，BDD よりも小さくブール多項式を保持できる．特にこの傾向は，XOR を多く含むような回路において顕著になることが知られている [49]．

3.4 提案手法

従来のガロア体算術演算回路の検証手法の問題点として，(1) 検証時に計算量が爆発する可能性があることと (2) 標数が 2 よりも大きいガロア体算術演算回路への適用可能性が不明という問題がある．(1) については，既存の検証手法では，前節で導入した命題 1 をもとに回路のプライマリ出力変数に対する簡約を繰り返すことで，等価性を調べる．また，この簡約手続きにおいて ZDD を多項式の表現に用いることで，計算時間の大幅な削減を行っている．この簡約処理は，RTTO を用いてグレブナー基底を導出した場合，各論理ゲートの出力変数を入力側変数に置き換える処理をプライマリ出力からプライマリ入力まで繰り返すことと等価となる．しかし，このような簡約操作では，中間処理で出現する多項式の ZDD による表現効率が低下するため，ZDD の利点を十分に活かすことが難しい． 特に，等価性の証明に必要な簡約操作は，プライマリ出力ビットを表現する変数全てに適用する必要があるが，各簡約における中間結果を共有することができないため，出力 k ビットの回路に関して完全に独立な k 回の簡約を実行する必要がある．例えば，i ビット目の出力を表現するガロア体方程式を簡約する場合，その簡約の中間結果は全て i ビット目の出力を中間変数（プライマリ入出力を除く論理ゲートの出力を表す変数）で表現するガロア体方程式となる．このとき，簡約一回あたりの計算コストは無視できないほ

ど大きいことに注意されたい．以上の理由から，ZDD の節点数の増加による簡約時の代入と，簡約回数の増加による計算時間の増加が発生する．

(2) については，標数が 2 以外のガロア体算術演算回路に，従来手法が適用可能かは不明という問題がある．特に，同手法では，ブール多項式環 $\mathbb{B}$ を用いて検証を実施することを想定しているため，$\mathbb{F}_{p^m}(p \geq 3)$ の拡大体を用いた回路の検証には適していないと考えられる．ペアリング暗号などの次世代暗号では，標数が 2 以上のガロア体表現を用いることで，より効率的に実装が可能であることが知られている．例えば，ペアリング暗号では標数 3 のガロア体 [57, 58, 59, 60, 61] や，超楕円曲線を用いたものでは標数 5，もしくは 7 がより高効率であることが知られている [62, 63]．このように，標数が 2 以上である多標数ガロア体算術演算回路は，ペアリングや超楕円曲線暗号では重要な役割を担っている．よって，多標数ガロア体算術演算回路の脆弱性検知は，ペアリングなどの次世代の暗号方式を活用する上で重要である．

以上の理由から，本節では (1) と (2) のそれぞれを解決するための，新たな検証手法を提案する．

3.4.1　高速な等価性検証手法の提案

本節では，(1) を解決するための新たな簡約アルゴリズムを提案する．

■提案アルゴリズム　$x_1, x_2, \ldots, x_n$ を検証対象回路のビットレベルの入力変数，$z_1, z_2, \ldots, z_m$ を出力変数とする．ここで，n と m は入力および出力ビット長である．

提案手法は次の 4 つのステップで実現される．

Step 1: 設計仕様のガロア体方程式をブール多項式へ変換する．

Step 2: すべてのゲートに対して，入出力間の関係式を抽出する．

Step 3: 検証対象回路のゲートレベルネットリストの全てのノード（論理ゲートの出力）の正規形（剰余）をプライマリ入力側から順次求める．

Step 4: Step 1 と Step 2 で得られた多項式の比較を行う．

既存手法と大きく異なるのは Step 2 である．同ステップは既存手法におけるプライマリ出力変数の正規化に対応する処理である．言い換えれば，提案手法では正規化処理の改良によって高速な回路機能検証を実現する．ここで，提案手法は多項式表現に既存手法と同様に ZDD を用いる．以下では，それぞれのステップについて説明する．

Step 1 では，検証対象の回路の仕様となるガロア体方程式からブール多項式を構築する．ここでは例として，$Z = A \times B$ で与えられるガロア体乗算器の場合を扱う．ただし，$A, B \in \mathbb{F}_{2^m}$ は入力変数であり，$Z \in \mathbb{F}_{2^m}$ は出力変数である．これらのワードレベル変数

は，次に示す関係式を通してビットレベル変数へ変換できる．

$$A = x_0 + x_1\alpha + \cdots + x_{s-1}\alpha^{s-1}, \tag{3.1}$$
$$B = x_s + x_{s+1}\alpha + \cdots + x_{2s-1}\alpha^{s-1}, \tag{3.2}$$
$$Z = z_0 + z_1\alpha + \cdots + z_{s-1}\alpha^{s-1}. \tag{3.3}$$

ここで，α は既約多項式の解である．Step 1 では，仕様となる方程式 $Z = AB$ に，式 (3.1) と (3.2), (3.3) を代入し，不定元を整理することでブール多項式の関係式を得る．

Step 2 では，検証対象の回路のすべてのゲートから，入出力間の関係式をブール多項式として抽出する．例えば，NOT と AND, OR, XOR ゲートであれば，以下の方程式として抽出される．

$$f_k = w_k + w_i + 1,$$
$$f_k = w_k + w_j w_i,$$
$$f_k = w_k + w_j w_i + w_j + w_i,$$
$$f_k = w_k + w_j + w_i.$$

ここで，w_i と w_j は入力変数であり，w_k は出力変数である．また f_k は配線 w_k を出力変数とするゲートに対応する多項式を表す．RTTO に従い，多項式 f_k の先頭項は w_k である．すべてのゲートから抽出された多項式集合 $\mathcal{G} = \{f_i \mid n+1 \leq i \leq l\}$ はグレブナー基底をなす．

Step 3 では，Algorithm 1 に従い，回路のすべての配線の正規形を求める．ここで，回路出力の正規形 $f'_{l-m+1}, f'_{l-m+2}, \ldots, f'_l$ は検証で求めるべきプライマリ出力のグレブナー基底による剰余であることに注意されたい．また，このプライマリ出力の正規形は，プライマリ入力変数だけからなる多項式となる．Algorithm 1 では，回路の入力側の配線から順番に正規形を計算し，最終的にプライマリ出力変数の正規形を求める．各正規形 f'_k は，$f'_k = \overline{w_k}^{\mathcal{G}}$ で与えられる．ここで $\overline{w_k}^{\mathcal{G}}$ は，グレブナー基底 $\mathcal{G} = \{f_{n+1}, \ldots, f_l\}$ による w_k の簡約結果（正規形）を表す．本アルゴリズムにおける正規系の計算はすべて ZDD を用いた多項式表現上で行われることに注意されたい．

Algorithm 1 の 3–5 行目では，プライマリ入力変数の正規形 $f'_1, f'_2, \ldots, f'_n$ を求める．ここでプライマリ入力変数の正規形は，入力変数の単項式（すなわち，$f'_1 = w_1, \ldots, f'_n = w_n$）で与えられることに注意されたい．6–18 行目では，入力側から順に各配線の正規形を計算する．関数 “GetInputIndex/Indices” は，配線 w_k を出力としてもつゲートの，入力配線のインデックスを返す関数である．Algorithm 1 の多項式簡約にかかる繰り返し回数は，回路内のゲートの個数と等しい．

例 3.4.1. 図 *3.2* に $\mathbb{F}_{2^2}$ の乗算器の例を示す．$w_1, w_2, \ldots, w_{11}$ は配線の変数である．また，$\mathbb{F}_{2^2}$ は，既約多項式 $\alpha^2 + \alpha + 1$ で定義される．ここで，α は既約多項式の根である．

Algorithm 1 提案する多項式簡約アルゴリズム

Require: $w_1, \ldots, w_l$: wires in the n-input circuit.
Ensure: $f'_1, \ldots, f'_l$: canonical representations of the wires.

```
1: k ← 1
2: while k ≤ l do
3:     if k ≤ n then
4:         f'_k ← w_k
5:     else
6:         if w_k is an output wire of the NOT gate then
7:             j ← GetInputIndex(w_k)
8:             f'_k ← f'_j + 1
9:         else
10:            {j, i} ← GetInputIndices(w_k)
11:            if w_k is an output wire of the AND gate then
12:                f'_k ← f'_j × f'_i
13:            else if w_k is an output wire of the OR gate then
14:                f'_k ← f'_j × f'_i + f'_j + f'_i
15:            else if w_k is an output wire of the XOR gate then
16:                f'_k ← f'_j + f'_i
17:            end if
18:        end if
19:    end if
20:    k ← k + 1
21: end while
```

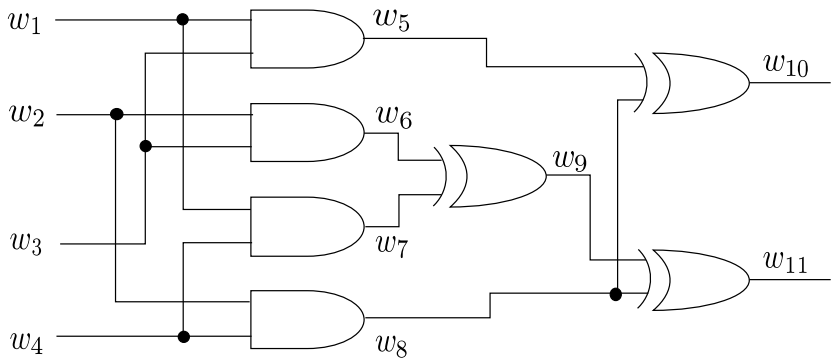


図 3.2: $\mathbb{F}_{2^2}$ のガロア体乗算器の例

Step 1: A と B は入力のワードレベル変数，Z を出力変数とする．入出力変数のワード

とビットレベル変数の間に次の関係が成り立つ．

$$A = \alpha w_2 + w_1, \tag{3.4}$$
$$B = \alpha w_4 + w_3, \tag{3.5}$$
$$Z = \alpha w_{11} + w_{10}. \tag{3.6}$$

ワードレベルの仕様 $Z = AB$ に式 (3.4) と (3.5)，(3.6) を代入すると，

$$\alpha w_{11} + w_{10} = \alpha(w_4 w_2 + w_4 w_1 + w_3 w_2) + w_4 w_2 + w_3 w_1$$

が得られる．両辺の不定元を比較することで，次のブール多項式が得られる．

$$w_{10} = w_4 w_2 + w_3 w_1,$$
$$w_{11} = w_4 w_2 + w_4 w_1 + w_3 w_2.$$

Step 2: 回路のすべてのゲートの入出力間の関係式として，

$$f_5 = w_5 + w_3 w_1, \quad f_6 = w_6 + w_3 w_2, \quad f_7 = w_7 + w_4 w_1,$$
$$f_8 = w_8 + w_4 w_2, \quad f_9 = w_9 + w_7 + w_6,$$
$$f_{10} = w_{10} + w_8 + w_5, \quad f_{11} = w_{11} + w_9 + w_8,$$

が得られる．

Step 3: *Algorithm 1* から，配線 $w_1, w_2, \ldots, w_{11}$ の正規形 $f'_1, f'_2, \ldots, f'_{11}$ が次の通り得られる．

$$f'_1 = w_1, \quad f'_2 = w_2, \quad f'_3 = w_3, \quad f'_4 = w_4,$$
$$f'_5 = f'_3 f'_1 = w_3 w_1, \quad f'_6 = f'_3 f'_2 = w_3 w_2,$$
$$f'_7 = f'_4 f'_1 = w_4 w_1, \quad f'_8 = f'_4 f'_2 = w_4 w_2,$$
$$f'_9 = f'_7 + f'_6 = w_4 w_1 + w_3 w_2,$$
$$f'_{10} = f'_8 + f'_5 = w_4 w_2 + w_3 w_1,$$
$$f'_{11} = f'_9 + f'_8 = w_4 w_2 + w_4 w_1 + w_3 w_2.$$

Step 4: *Step 1* と *3* で得られたブール多項式を比較する．これらが一致することから，図 *3.3* の正しさが証明された．

■提案アルゴリズムの正当性の証明　提案手法の正当性を証明する．具体的には，Algorithm 1 から得られる $f'_1, \ldots, f'_i, \ldots, f'_l$ がすべて正規形であることを示す．もしすべての多項式 f'_i が，グレブナー基底 $\mathcal{G} = \mathcal{F} = \{f_{n+1}, \ldots, f_l\}$ による各配線 w_i の正規形ならば，提案手法の正当性が示される．

これを示すために，まず補題 1 を示す．

補題 1. 入力ビット長 n をもつ任意の組合せ回路を C とする．C のすべての配線集合を $\{w_1, \ldots, w_i, \ldots, w_l\}$ とし，すべてのゲートの入出力の関係式からなる多項式集合を $\mathcal{G} = \{f_{n+1}, \ldots, f_l\}$ とする．ここで，$\mathcal{G}$ は項順序として *RTTO* を用いることで，グレブナー基底となる．任意の多項式 $f \in \mathbb{B}(w_1, \ldots, w_l)$ が正規形であることと，f が $\mathbb{B}(w_1, \ldots, w_n) = \mathbb{F}_2[w_1, \ldots, w_n]/\langle w_1^2 + w_1, w_2^2 + w_2, \ldots, w_n^2 + w_n\rangle$ の元であることは同値である．

Proof. ($\Rightarrow$) f は正規形であるため，多項式 f の先頭項は，すべてのゲートの多項式 $f_{n+1}, f_{n+2}, \ldots, f_l$ で割り切ることができない．項順序が RTTO のため，$f_i \in \mathcal{G}$ $(n+1 \leq i \leq l)$ の先頭項の集合 $\{\mathrm{lt}(f_i) \mid i = n+1, \ldots, l\}$ は，すべてのゲートの出力配線の集合と等しい．したがって，f が $\mathcal{G}$ によって割り切れないということは，f がプライマリ入力変数のみを含むことと等しい．よって，f は $\mathbb{B}(w_1, \ldots, w_n)$ の元である．

($\Leftarrow$) f が $\mathbb{B}(w_1, \ldots, w_n)$ の元のとき，f の先頭項はどの多項式の先頭項 $\mathrm{lm}(f_i)$ でも割り切ることはできない．したがって，f は $\mathcal{G}$ の正規形である．　□

次に，２つの正規形の和や乗算もまた，正規形であることを証明する．

補題 2. $\mathbb{B}(w_1, \ldots, w_l)$ をブール多項式環とし，f_1 と f_2 を $\mathbb{B}(w_1, \ldots, w_l)$ の多項式とする．このとき，以下が成り立つ．

$$\overline{f_1 + f_2}^{\mathcal{G}} = \overline{f_1}^{\mathcal{G}} + \overline{f_2}^{\mathcal{G}}, \tag{3.7}$$

$$\overline{f_1 \times f_2}^{\mathcal{G}} = \overline{f_1}^{\mathcal{G}} \times \overline{f_2}^{\mathcal{G}}. \tag{3.8}$$

ここで，$\overline{f}^{\mathcal{G}}$ は，多項式 $f \in \mathbb{B}(w_1, \ldots, w_l)$ の $\mathcal{G}$ による正規形を表す．

Proof. f_1 と f_2 と，その正規形 $\overline{f_1}^{\mathcal{G}}$ と $\overline{f_2}^{\mathcal{G}}$ に対して，$f_1 = u_1 + \overline{f_1}^{\mathcal{G}}$ と $f_2 = u_2 + \overline{f_2}^{\mathcal{G}}$ を満たすような多項式 $u_1, u_2 \in \mathcal{I}$ が存在する．したがって，f_1 と f_2 の和は，$f_1 + f_2 = u_1 + u_2 + \overline{f_1}^{\mathcal{G}} + \overline{f_2}^{\mathcal{G}}$ である．u_1 と u_2 はイデアル $\mathcal{I}$ の元なので，その和 $u_1 + u_2$ も $\mathcal{I}$ の元である．すなわち，$u_1 + u_2 = a_l f_l + \cdots + a_{n+1} f_{n+1}$ を満たすような多項式 $a_l, \ldots, a_{n+1} \in \mathbb{B}(w_1, \ldots, w_l)$ が存在する．よって，$f_1 + f_2 = a_l f_l + \cdots + a_{n+1} f_{n+1} + \overline{f_1}^{\mathcal{G}} + \overline{f_2}^{\mathcal{G}}$ である．ここで，$a_i f_i, i \in \{n+1, \ldots, l\}$ の先頭項は，RTTO において $\overline{f_1}^{\mathcal{G}}$ と $\overline{f_2}^{\mathcal{G}}$ よりも順序が上である．したがって，$f_1 + f_2$ の G による割り算の過程で，$\overline{f_1}^{\mathcal{G}} + \overline{f_2}^{\mathcal{G}}$ を得る．補題 1 から，$\overline{f_1}^{\mathcal{G}} + \overline{f_2}^{\mathcal{G}}$ は正規形である．よって，$\overline{f_1 + f_2}^{G} = \overline{f_1}^{\mathcal{G}} + \overline{f_2}^{\mathcal{G}}$ である．

次に式 (3.8) に着目する．$f_1 = u_1 + \overline{f_1}^{\mathcal{G}}$ かつ $f_2 = u_2 + \overline{f_2}^{\mathcal{G}}$ より，$f_1 \times f_2 = u_1 u_2 + u_1 \overline{f_2}^{\mathcal{G}} + \overline{f_1}^{\mathcal{G}} u_2 + \overline{f_1}^{\mathcal{G}} \overline{f_2}^{\mathcal{G}}$ である．ここで，式 (3.7) が成立することから，$\overline{f_1 \times f_2} = \overline{f_1}^{\mathcal{G}} \overline{f_2}^{\mathcal{G}}$ である．　□

上記の補題を用いることで，提案手法の正当性を証明できる．

定理 1. *(提案手法の正当性) Algorithm 1* で得られる多項式 $f'_1, \ldots, f'_l$ は，すべての配線の正規形 $\overline{w_1}^{\mathcal{G}}, \ldots, \overline{w_l}^{\mathcal{G}}$ と等しい．

Proof. i 番目の多項式に関する数学的帰納法によって証明する．

1. $i \leq n$ のとき，f'_i は配線 w_i に等しい．補題 1 から，w_i は正規形である．したがって，$w_i \xrightarrow{\mathcal{G}} f'_i$ である．
2. $i > n$ とし，$w_{i_1}, \ldots, w_{i_v}$ を配線 w_i を出力としてもつゲートの入力変数とする．帰納法の仮定から，入力配線 $w_{i_1}, \ldots, w_{i_v}$ の正規形 $f'_{i_1}, \ldots, f'_{i_v}$ はすでに得られていることに注意されたい．ここで，$\mathrm{lt}(f_i) = w_i$ を満たす多項式 $f_i \in \mathcal{G}$ が存在する．f_i は多項式の抽出方法から，必ず $w_i + \xi_i(w_{i_1}, \ldots, w_{i_v})$ とかける必要がある．ここで，ξ_i は i 番目のゲートの論理機能を表す関数である．w_i の $\mathcal{G}$ による簡約では，必ず f_i による簡約が行われ，その結果 $w_i \xrightarrow{f_i} \xi_i(w_{i_1}, \ldots, w_{i_v})$ が得られる．補題 2 から，多項式の加算や乗算結果の正規形は，正規形の加算と乗算結果と等しい．よって，w_i の $\mathcal{G}$ による簡約は，$w_i \xrightarrow{f_i} \xi_i(w_{i_1}, \ldots, w_{i_v}) \xrightarrow{\mathcal{G}} \xi_i(\overline{w_{i_1}}^{\mathcal{G}}, \ldots, \overline{w_{i_v}}^{\mathcal{G}})$ となる．帰納法の仮定から，$w_{i_v} \xrightarrow{\mathcal{G}} f'_{i_v}$ である．よって，$w_i \xrightarrow{\mathcal{G}} \xi_i(f'_{i_1}, \ldots, f'_{i_v})$ となる．Algorithm 1 から，$f'_i = \xi_i(f'_{i_1}, \ldots, f'_{i_v})$ であり，以上より $w_i \xrightarrow{\mathcal{G}} f'_i$ である．

□

3.4.2 多標数ガロア体算術演算回路の検証手法の提案

従来のガロア体算術演算回路の検証手法の問題は，標数が 2 の演算器のみが検証対象だったことである．そこで本節では，従来の検証手法を拡張し，標数が 2 よりも大きいガロア体算術演算回路にも適用可能な等価性検証手法を提案する．以降，標数が 2 ではないガロア体を多標数ガロア体とよぶことにする．まず，本節では提案する多標数ガロア体算術演算回路のための新たな簡約手法を提案する．次に，標数が 2 の場合の検証に用いた ZDD と対応する，多標数向けの新たな決定グラフとして，GFBMD (Galois-Field Binary Moment Diagram) を提案する．最後に，提案手法を用いて多標数ガロア体算術演算回路の検証を実施し，その有効性を確認する．

■提案する簡約手法 提案手法の検証アルゴリズムについて説明する．提案手法は，従来の非階層的な検証手法と同様の手続きを，多標数へ拡張することによって検証を実施する．提案手法と従来手法の相違点は，ガロア体 $\mathbb{F}_{p^m}$ のガロア体算術演算回路の検証に，その素体 $\mathbb{F}_p$ を多項式表現に用いることである．提案手法の適用では，与えられるネットリストは素体 $\mathbb{F}_p$ 上の演算を最小の構成要素と仮定する（従来の組み合わせ回路は AND や XOR などの $\mathbb{F}_2$ 上の演算を構成要素とするとみなせる）．

提案手法における検証の手続きは次の 4 つのステップによって行われる．

Step 1: 回路の入出力が満たすべき $\mathbb{F}_{p^m}$ 上の方程式を，素体上の連立方程式へ変換する．

Step 2: 与えられたネットリスト（Verilog HDL など）から，素体上の演算モジュールの接続関係を抽出し，連立代数方程式を立式する．また同時にグレブナー基底の導出も行う．

Step 3: ネットリストのプライマリ出力変数を，Step 2 で得られたグレブナー基底を用いて簡約する．

Step 4: Step 1 と Step 3 で得られた方程式の比較を行う．

Step 4 で 2 つの方程式の間の等価性が認められた時点で，検証対象の回路の正しさが証明される．

Step 1 では，まず，回路の入出力が満たすべき仕様を，素体上 $\mathbb{F}_p$ の連立方程式へと変換する．ここでは具体例として，拡大体 $\mathbb{F}_{p^m}$ 上のガロア体乗算器の例を考える．ガロア体乗算器の入力変数を A, B，出力変数を Z とする．ガロア体乗算器の仕様は，入出力変数を用いて $Z = AB$ とかける．ここで，$a_1, a_2, \ldots, a_m$，$b_1, b_2, \ldots, b_m, z_1, z_2, \ldots, z_m \in \mathbb{F}_p$ を入出力 A, B, Z の係数とすると，

$$A = a_1 + a_2\alpha + \ldots, a_s\alpha^{s-1} = \sum_{i=1}^{m} a_i\alpha^{i-1}, \tag{3.9}$$

$$B = b_1 + b_2\alpha + \ldots, b_s\alpha^{s-1} = \sum_{i=1}^{m} b_i\alpha^{i-1},$$

$$Z = z_1 + z_2\alpha + \ldots, z_s\alpha^{s-1} = \sum_{i=1}^{m} z_i\alpha^{i-1} \tag{3.10}$$

が成り立つ．ここで α は不定元であり，既約多項式 $P(\alpha) = 0$ が成り立つ．式 (3.9)–(3.10) を $Z = A \times B$ に代入することで，

$$\begin{aligned}\sum_{i=1}^{m} z_i\alpha^{i-1} &= \left(\sum_{i=1}^{m} a_i\alpha^{i-1}\right)\left(\sum_{i=1}^{m} b_i\alpha^{i-1}\right) \\ &= \sum_{i=1}^{m}\sum_{j=1}^{m} a_i b_j \alpha^{i+j-2} \\ &= \sum_{i=1}^{m} f_i(a_1, a_2, \ldots, a_m, b_1, b_2, \ldots, b_m)\alpha^{i-1}\end{aligned}$$

を得る．ここで，$f_i(a_1, a_2, \ldots, a_m, b_1, b_2, \ldots, b_m)$ は $\mathbb{F}_p$ の元であり，変数 $a_1, a_2, \ldots, a_m, b_1, b_2, \ldots, b_m$ からなる多項式である．多項式 $f_i(a_1, a_2, \ldots, a_m, b_1, b_2, \ldots, b_m)$ を具体的に求めることでガロア体乗算器の仕様を得ることができる．

Step 2 では，ネットリストから素体 $\mathbb{F}_p$ 上の方程式を抽出する．上述の仮定から，ネッ

Algorithm 2 多標数ガロア体データパスのための多項式簡約アルゴリズム

Require: $w_1, \ldots, w_l$: All edges, $f_{u+1}, \ldots, f_l$: All the polynomials of input–output relations
Ensure: $f'_{l-v+1}, \ldots, f'_l$: Canonical representations of all edges connected to the POs

```
1: for k ← 1 to l do
2:     z ← w_{l−v+k}
3:     while z is not a canonical representation do
4:         f_i ← GetPoly(z)                    ▷ Get f_i ∈ G such that lt(f_i) | lt(z).
5:         z ← Reduce(z, f_i)                  ▷ Reduce z by f_i.
6:     end while
7:     f'_{l−v+k} ← z
8: end for
```

トリストの最小構成要素は素体上の演算（加算や乗算）である．もし標数 p が 2 であれば，すべての論理演算 AND, XOR, OR などが $\mathbb{F}_2$ 上の演算に対応する．

ここで，以降の説明のために，必要な用語の定義を行う．まず，与えられた回路記述に現れる最小の構成要素（すなわち素体上の演算）をモジュールと呼び，それらをつなぐものをエッジと呼ぶことにする．もし標数 2 の場合，エッジは単に一配線となる．検証対象の回路記述に現れるエッジの総数を l とし，すべてのエッジに対して RTTO に従って出力側から順番に $w_l, w_{l-1}, \ldots, w_1$ と名付ける．エッジに対する命名規則から，添字 i, j について $j > i$ が成り立つとき，エッジ w_j は w_i よりもより出力側に位置する．加えてプライマリ出力と入力につながっているエッジの数を，それぞれ n と m とする．したがって，$w_{l-m+1}, w_{l-m+2}, \ldots, w_l$ と，$w_1, w_2, \ldots, w_n$ は，それぞれプライマリ出力と入力につながっているエッジを指す．

また，プライマリ入力ではないすべてのエッジは，必ず回路を構成するモジュールの出力につながる．言い換えれば，添字 i が $n+1$ 以上であるような，あるエッジ w_i には，その信号値を決定する素体上の方程式 f_i が存在する．この方程式は，形式的に $f_i = w_i - \mathrm{tail}(f_i)$ と表せる．ここで $\mathrm{tail}(f_i)$ は，エッジ w_i を出力とするようなモジュールの入力エッジからなる多項式である．そして，この方程式の集合 $\mathcal{J} = \{f_i \mid n+1 \leq i \leq l\}$ と，消失イデアル $\mathcal{J}_0 = \{w_i^p - w_i \mid 1 \leq i \leq l\}$ の和集合 $\mathcal{G} = \mathcal{J} \cup \mathcal{J}_0$ は，RTTO を導入することでグレブナー基底なる[*1]．Step 2 の目的は，このグレブナー基底 $\mathcal{G}$ をネットリストから抽出することである．

Step 3 では，プライマリ出力に接続されている各エッジ $w_{l-m+1}, \ldots, w_l$ を，グレブナー基底 $\mathcal{G}$ で簡約する．提案手法の簡約アルゴリズムを Algorithm 2 に示す．Algorithm 2

[*1] 各モジュールの出力に接続するエッジは唯一（2 つ以上のモジュールの出力に接続されたエッジが存在しない）であることから，各多項式 f_i の先頭項 $\mathrm{lt}(f_i)$ も唯一（$i \neq j$ であるような添字 i と j について，$\mathrm{lt}(f_i)$ と $\mathrm{lt}(f_j)$ は互いに素）に決定できる．この条件は，多項式集合 $\mathcal{G} = \{f_i \mid v+1 \leq i \leq l\}$ が，そのイデアル $\langle \mathcal{G} \rangle$ のグレブナー基底である十分条件であることから，$\mathcal{G}$ はグレブナー基底である．

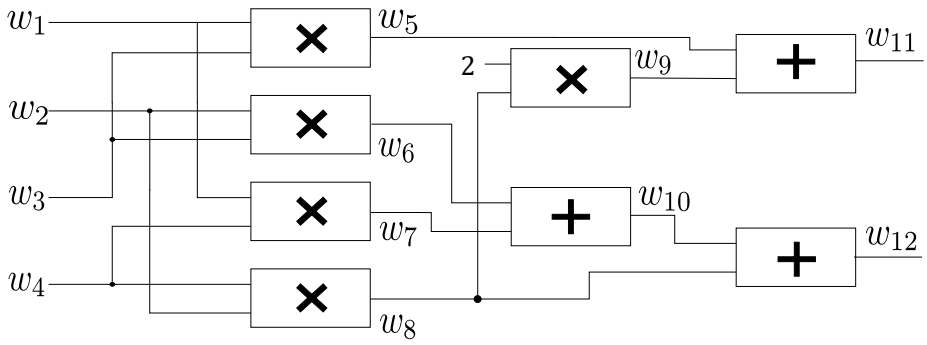


図 3.3: $\mathbb{F}_{3^2}$ のガロア体乗算器

は，従来手法 [54] を多標数へ拡張したものである．まず，2 行目でプライマリ出力へ接続されているエッジの一つを変数 z へ格納する．次に，3 行目から 6 行目で z の G による簡約を行う．具体的には，まず 4 行目の “GetPoly” で，z の先頭項 $\mathrm{lt}(z)$ を割り切るような，先頭項をもつ多項式 f_i をグレブナー基底から取り出す．次に 5 行目で，多項式 f_i を用いて z の簡約を実施し，それを z に格納する．この簡約操作は，多項式 z に含まれる f_i の先頭項の変数を，$\mathrm{tail}(f_i)$ で置き換えることで行われる．この操作を z が簡約できなくなるまで繰り返す．そして，最後に 7 行目で f'_{l-u+k} へ z を格納する．Algorithm 1 の停止性は，グレブナー基底の定義から明らかである．加えて，計算量は簡約回数に依存するためネットリストに出現するモジュールの数に比例すると考えられる．

Step 4 では，Step 3 で得られた $f'_{l-u+1}, \ldots, f'_l$ を，Step 1 で得られた仕様と比較する．もしこれらが等しければ回路記述が正しいといえる．

例 3.4.2. 図 *3.3* に示す拡大体 $\mathbb{F}_{3^2}$ 上の乗算回路を例として，提案手法の計算の流れを示す．ここで $+$ と $\times$ と記載されたモジュールはそれぞれ素体上の加算と乗算を表す．

Step 1: まず，仕様を素体上の方程式へ変換する．拡大体上の入力変数を A, B，出力変数 Z とすれば，次が成り立つ．

$$\begin{aligned} A &= \alpha w_2 + w_1, \\ B &= \alpha w_4 + w_3, \\ Z &= \alpha w_{12} + w_{11}. \end{aligned}$$

ここで，α は，既約多項式 $t^2 + 2t + 1$ の解である．したがって，$Z = AB$ は

$$\alpha w_{12} + w_{11} = \alpha(w_4 w_2 + w_4 w_1 + w_3 w_2) + 2 w_4 w_2 + w_3 w_1$$

となる．ここから仕様として，次の素体上の方程式が得られる．

$$\begin{aligned} w_{12} &= w_4 w_2 + w_4 w_1 + w_3 w_2, \\ w_{11} &= 2 w_4 w_2 + w_3 w_1. \end{aligned}$$

Step 2: 回路構成に従って素体上の方程式を求める．具体的には次のようになる．

$$f_5 = w_5 - w_3w_1, f_6 = w_6 - w_3w_2, f_7 = w_7 - w_4w_1,$$
$$f_8 = w_8 - w_4w_2, f_9 = w_9 - 2w_8,$$
$$f_{10} = w_{10} - w_7w_6, f_{11} = w_{11} - (w_9 + w_5),$$
$$f_{12} = w_{12} - (w_{10} + w_8).$$

Step 3: *Algorithm 1* に従って，出力変数 w_{11}, w_{12} の簡約を行う．w_{11} は次のようになる．

$$w_{11} \xrightarrow{f_{11}} w_9 + w_5 \xrightarrow{f_9} 2w_8 + w_5$$
$$\xrightarrow{f_8} w_5 + 2w_4w_2 \xrightarrow{f_5} 2w_4w_2 + w_3w_1.$$

同様に，w_{12} についても次のようになる．

$$w_{12} \xrightarrow{f_{12}} w_{10} + w_8 \xrightarrow{f_{10}} w_8 + w_6 + w_7$$
$$\xrightarrow{f_8} w_7 + w_6 + w_4w_2 \xrightarrow{f_7} w_6 + w_4w_2 + w_4w_1$$
$$\xrightarrow{f_6} w_4w_2 + w_4w_1 + w_3w_2.$$

Step 4: *Step 3* で得られた正規形を *Step 1* の仕様の方程式と比較する．今回の例では，これらが同一であることから回路の正しさが証明された．

■Galois-field binary moment diagrams　ZDD を多標数ガロア体算術演算回路を検証するために拡張した GFBMD について説明する．

GFBMD は，整数演算回路の検証に用いられる*BMD（binary moment diagram）とほぼ同じ構成をもつ．*BMD は次の正極性ダビオ展開に類似の式がベースとなっている．具体的には，変数 x に関して一次の任意の多項式 $f(x)$ は，必ず次の形で展開できる．

$$f(x) = c_1f_1 + xc_2f_2. \tag{3.11}$$

ここで，c_1 と c_2 は整数係数，f_1 と f_2 は変数 x を含まない多項式である．式 (3.11) を繰り返し適用することで，どのような多変数の一次多項式も二分決定木の形に分解できる．このような展開を用いた二分決定木を作成し，3.3.2 節で述べた ZDD と同様の削除規則を適用することで得られる決定グラフを*BMD という．一方，本節で導入する GFBMD は，式 (3.11) の整数係数 c_1 と c_2 をガロア体 $\mathbb{F}_p$ の元として，$\mathbb{F}_p$ 上の方程式 $f(x)$ を展開することで得られる．GFBMD は次数が 2 以上の多項式を表現することができないものの，様々な応用に用いることができると考えられる．

*BMD のように正規化を行わなければ，多項式に対して，GFBMD は一意にならない．一般に，*BMD では一意性を確保するため，式 (3.11) の係数を最大公約数でくくりだす

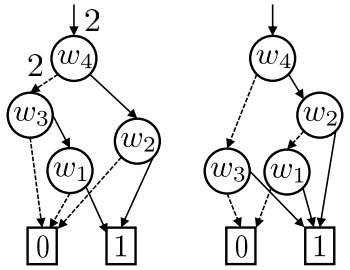


図 3.4: $2w_4w_2 + w_3w_1$ （左）と $w_4w_2 + w_4w_1 + w_3w_2$ （右）を表す GFBMD

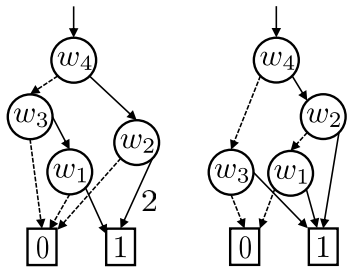


図 3.5: $2w_4w_2 + w_3w_1$（左）と $w_4w_2 + w_4w_1 + w_3w_2$（右）を表す*BMD

[64]．一方，GFBMD は，係数が $\mathbb{F}_p$ の元であるため，最大公約数でくくりだしても表現が一意にならない．そこで，提案する GFBMD では，最大公約数ではなく c_2 でくくりだすようにする．すなわち，正規化された GFBMD では，c_2 はつねに値が 1 になる．この制約を設けることで，GFBMD と多項式は一対一対応になる．また，GFBMD を用いて $\mathbb{F}_p$ 上の方程式を表現する場合，GFBMD 上で多項式同士の加算や乗算を行えることが望ましい．これは，*BMD の APPLY 演算を GFBMD に拡張することで容易に実現される．

*BMD と GFBMD の差は主に正規化する際の因子に起因している．したがって，GFBMD の構築アルゴリズムは*BMD とほぼ同じである．より正確には，GFBMD は文献 [64] の “NormWeight” 関数を変えるだけで実装可能である．

図 3.4 に例 3.4.2 に示した回路出力の多項式に対応する GFBMD を示している．GFBMD は変数を表す節点と，実線と点線で示された二種類の枝，そして 0, 1-終端節点で構成される．w_i の節点から伸びる実線と点線の枝は，それぞれ w_i と “1” を乗算することを表す．加えて，各枝の横に表記されている数は乗数を表す．これらの枝と数字は式 (3.11) の展開時の係数に対応している．例えば，図 (3.4) の左のグラフにおいて，根から w_4 の点線，w_3 の実線，w_1 の実線を通って，1-終端節点にたどり着くパスは多項式 $2 \times 2 \times w_3 \times w_1 = w_3w_1$ を表す．このように，根から葉までのすべてのパスに対応する多項式の和が，グラフ全体が表す多項式となる．

比較のために，図 3.5 に GFBMD に対応する*BMD を示す．$2w_4w_2 + w_3w_1$ の*BMD

は GFBMD と正規化の際の因子のために違いがある．例えば，同図の*BMD の w_2 の節点の実線の乗数は 2 だが，これは GFBMD では用いることはできない．一方で，多項式 $w_4w_2 + w_4w_1 + w_3w_2$ の GFBMD と*BMD は，正規化のための因子が一致するため同じグラフ構造を有する．

3.5 暗号ハードウェアの検証実験

本節では，前節で提案した 2 つの等価性検証手法の有効性を，実際の暗号データパスへ適用することで確認する．まず，標数 2 のガロア体に基づく暗号ハードウェアとして ECC と AES に対して検証を行う．次に，標数が 3 以上のガロア体算術演算回路として，ペアリング暗号で使用されるガロア体乗算器に対して等価性検証を行った結果を示す．

3.5.1 ECC と AES データパスの検証実験

提案手法の有効性を楕円曲線暗号（ECC: Elliptic Curve Cryptography）向けの Mastrovito 乗算器と，AES ハードウェアへ適用し確認する．まず，Mastrovito 乗算器の検証について述べる．検証対象は，文献 [65, 66] で提案された，ガロア体モジュール生成器によって生成された Mastrovito 乗算器のゲートレベルネットリストである．Mastrovito 乗算器は多項式基底表現に基づくガロア体の典型的な並列乗算器である．例として，図 3.6 に，$\alpha^4 + \alpha + 1$ を既約多項式とする $\mathbb{F}_{2^4}$ の乗算器を示す．Mastrovito 乗算器は，最初に多項式基底における乗数の剰余による余りから決定される行列を生成し，次に生成した行列と被乗数との間の乗算を行う．Mastrovito 乗算器は既約多項式の次数が増えるに従って，入出力のビット長も増える．ここで，検証のためのツールは，C++ を用いて作成し，高効率な決定グラフの操作が可能な PolyBori ライブラリを使用した．また，実験に使用した計算機は，メモリ容量が 384GB，CPU が 3.50-GHz Intel Xeon Gold 6144 の Linux サーバである．

図 3.7 に，Mastrovito 乗算器の提案手法による検証にかかった時間を示す．ここで，横軸は拡大次数（入力オペランド長），縦軸は検証時間を示す．比較のために，提案手法に加えて従来手法で最も高速な検証手法として，Gupta らの結果も合わせて示す [53]．本実験では，楕円曲線暗号（ECC: Elliptic Curve Cryptography）向けに推奨されたパラメータを含む，64 から 571 ビットの範囲でオペランド長を変えた Mastrovito 乗算器を使用した．従来手法の検証時間は，文献 [53] に示されたアルゴリズムに従い，再実装したプログラムを実行して得たものである．再現した値は，元の文献 [53] で示された検証時間とほぼ一致することを確認している．従来手法では，仕様となる方程式からビットレベル（ブール多項式）を得るのにかかる時間を検証時間に含んでいなかったため，本稿でもこ

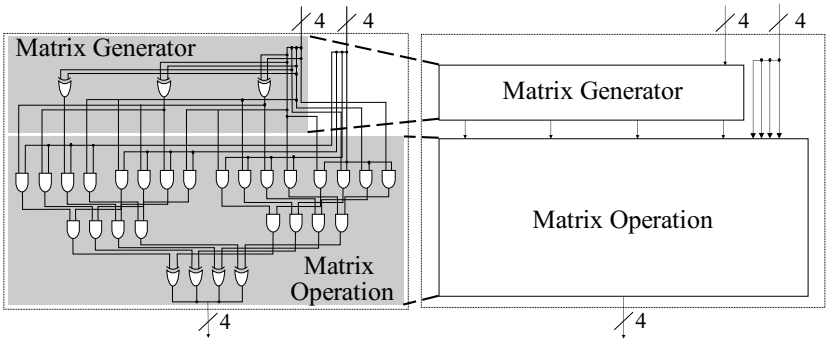


図 3.6: $\mathbb{F}_{2^4}$ の Mastrovito 乗算器の例

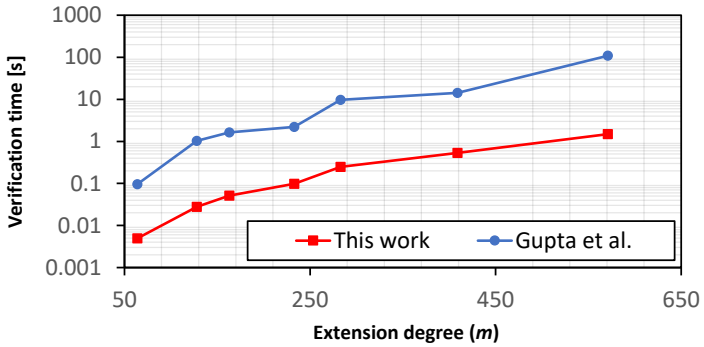


図 3.7: 様々な拡大次数における Mastrovito 乗算器の検証時間

れに従うこととした．図 3.7 から，提案手法は従来手法と比べて圧倒的に短い時間で検証が完了していることが確認できる．

表 3.1 に，検証時間とゲート数を示す．同表から提案手法は Mastrovito 乗算器の規模がたとえ大きい場合でも，高速に検証できることが確認できる．結果的に，提案手法は平均して約 20 倍，最大で 70 倍の高速化を実現した．

次に，バグを含む Mastrovito 乗算器の検証実験を行った．提案手法が，バグを含む回路に対しても効率的に検証が可能なことを確認するために，233 ビット Mastrovito 乗算器に対して，様々なバグを挿入し，その検証時間を計測した．挿入したバグは，乗算器内のいくつかのゲートを異なる種類の論理ゲートに置き換えるものと，特定のゲートの入力配線を誤ったゲートの出力に接続するものである．それぞれのバグを 0 から 10 個まで挿入した 100 種類の回路を，各 10 回生成し，その検証時間を測定した．図 3.8 に，提案手法によりバグを含む Mastrovito 乗算器の検証を行った結果を示す．ここで，検証時間は各種類の乗算器の 10 回の検証時間の平均を示している．図中の “wrong gates” が誤った種類のゲートを使用した場合を示し，“wrong wires” が誤った配線を指す．比較のために，バグが含まれない乗算器の検証時間も座標 (0, 0) の点に示した．同図から，提案手法はバグの有無に関わらず検証時間がほぼ変化せず，回路の故障に対して頑健なことが読み

表 3.1: Mastrovito 乗算器のゲート数と，検証時間

m	Number of gates	Verification times [s]	
		This work	Gupta at el.
64	8.40×10^3	4.97×10^{-3}	9.64×10^{-2}
128	3.32×10^4	2.78×10^{-2}	1.03
163	5.37×10^4	5.13×10^{-2}	1.63
233	1.09×10^5	9.86×10^{-2}	2.21
283	1.61×10^5	2.47×10^{-1}	9.74
409	3.35×10^5	5.35×10^{-1}	1.42×10
571	6.53×10^5	1.49	1.09×10^2

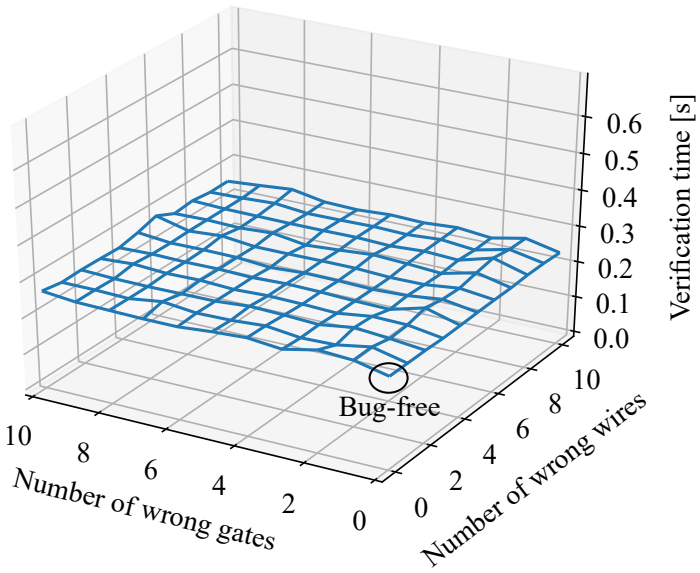


図 3.8: 故障の数を変化させたときの 233 ビット Mastrovito 乗算器の検証時間

取れる．

最後に，AES ハードウェアの検証実験の結果を示す．本実験では，合成体による逆元演算を含む，AES の一ラウンドデータパスの検証を行った．また，より実践的な例として，サイドチャネル攻撃に対する対策が施された AES データパスの検証も行った．図 3.10 に，検証対象となるマスキング対策された AES ハードウェアのブロック図を示す．同データパスの入力ビット長は 512，出力ビット長は 128 である．同図の左側に示すとおり，MaskedSBox は $\mathbb{F}_{2^8}$ から，$\mathbb{F}_{((2^2)^2)^2}$ への同型写像 δ と，その逆写像 δ^{-1}，アフィン変換，そしてマスクの調整部から構成される．

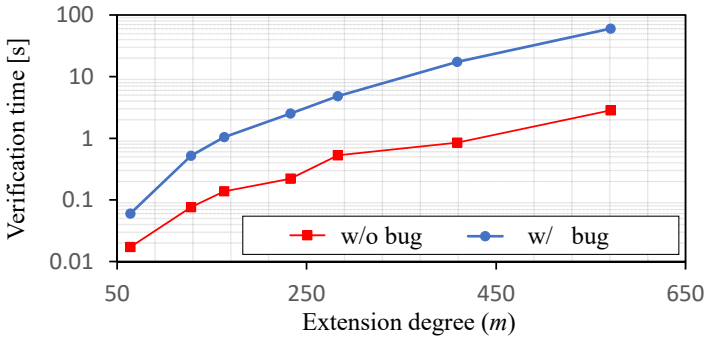


図 3.9: 特定の入力で故障を引き起こす脆弱性が含まれた Mastrovito 乗算器の検証時間

表 3.2: AES のラウンドデータパスの検証時間

	検証時間 [s]	項数	節点数	比率
w/o masking	3.41×10^{-1}	6.82×10^{2}	3.59×10^{2}	5.26×10^{-1}
w/ masking	1.11×10	1.71×10^{4}	3.61×10^{3}	2.11×10^{-1}

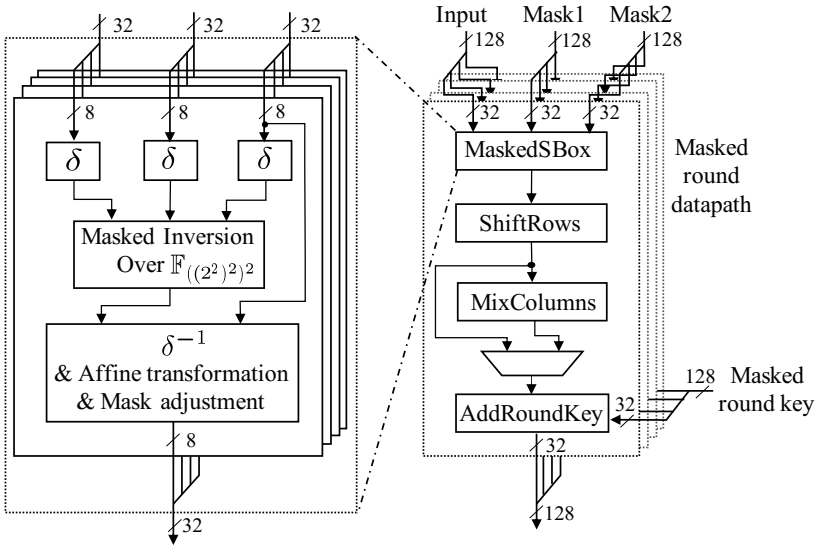


図 3.10: Masked AES round datapath[66].

表 3.2 に 128 ビット AES ラウンドデータパスの提案手法による検証時間を示す．また，ZDD 表現による効率化の度合いを調べるために，同表には検証中に出現した項数最大の多項式の項数と，ZDD の節点数，圧縮の比率も示した．同表に示すとおり，提案手法を用いることで，対策なしが 3.41×10^{-1} 秒，対策ありが 1.11×10 秒で検証できることがわかる．ここで，階層的なグラフ表現を用いる GF-ACG では，同ネットリストの検証は不可能であることに注意されたい．

表 3.3: ガロア体乗算器の検証時間

	$m=8$				$m=16$				$m=32$			
	$p=2$	$p=3$	$p=5$	$p=7$	$p=2$	$p=3$	$p=5$	$p=7$	$p=2$	$p=3$	$p=5$	$p=7$
Num. of additions	143	85	56	84	627	487	240	360	2,545	2,488	992	1,488
Num. of multiplications	64	92	92	120	256	376	376	496	1,024	1,520	1,520	2,016
Num. of terms of IP	5	3	2	3	5	4	2	3	5	5	2	3
Verification time [ms]	6.39	4.83	4.23	5.46	18.9	18.3	9.89	15.2	88.5	140	39.3	69.8
Time / Num. of terms	1.28	1.61	2.12	1.82	3.78	4.58	4.95	5.07	17.7	28.0	19.7	23.3

	m=64				m=128				m=256			
	$p=2$	$p=3$	$p=5$	$p=7$	$p=2$	$p=3$	$p=5$	$p=7$	$p=2$	$p=3$	$p=5$	$p=7$
Num. of additions	10,095	6,051	4,032	6,048	40,704	40,673	16,256	24,385	163,324	195,875	65,280	163,209
Num. of multiplications	4,096	6,112	6,112	8,128	16,394	24,512	24,512	32,640	65,536	98,176	98,176	130,819
Num. of terms of IP	5	3	2	3	5	5	2	3	5	6	2	5
Verification time [ms]	1030	599	424	706	21,100	26,700	6,290	15,300	315,000	554,000	105,000	480,000
Time / Num. of terms	206	200	212	235	4,220	5,340	3,150	5,100	63,000	92,300	52,500	96,000

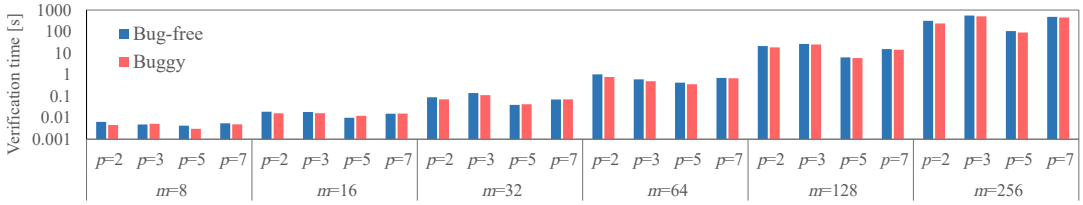


図 3.11: バグが含まれたガロア体乗算器の検証時間

3.5.2　多標数ガロア体乗算器の検証実験

■多標数ガロア体乗算器の検証　まず，GFBMD を用いずに多標数ガロア体算術演算回路の検証を行った場合の結果を示す．ガロア体乗算器に対する提案手法の実験結果を示す．2 入力の拡大体 $\mathbb{F}_{p^m}$ 上の乗算器を対象として，標数 p は 2, 3, 5, 7，拡大次数 m は 8 から 256 まで変化させて，それぞれについて検証を行う．

表 3.3 に，提案手法による検証時間を示す．実験結果から，検証時間は拡大次数に強く依存しており，標数の違いによってほとんど依らないことがわかる．一般に，各素体上の

表 3.4: 故障が含まれたガロア体乗算器の検証時間

	$m=8$				$m=16$				$m=32$			
	$p=2$	$p=3$	$p=5$	$p=7$	$p=2$	$p=3$	$p=5$	$p=7$	$p=2$	$p=3$	$p=5$	$p=7$
Verification time of bug-free one [ms]	6.39	4.83	4.23	5.46	18.9	18.3	9.89	15.2	88.5	140	39.3	69.8
Verification time of buggy one [ms]	4.54	5.17	3.03	4.91	15.9	16.2	12.2	15.5	70.5	111	42.3	70.5
Ratio	0.710	1.07	0.716	0.899	0.841	0.885	1.23	1.02	0.797	0.793	1.08	1.01

	$m=64$				$m=128$				$m=256$			
	$p=2$	$p=3$	$p=5$	$p=7$	$p=2$	$p=3$	$p=5$	$p=7$	$p=2$	$p=3$	$p=5$	$p=7$
Verification time of bug-free one [ms]	1,030	599	424	706	21,100	26,700	6,290	15,300	315,000	554,000	105,000	480,000
Verification time of buggy one [ms]	776	492	357	677	18,500	25,200	5,870	14,300	239,000	506,000	90,800	449,000
Ratio	0.753	0.821	0.842	0.959	0.877	0.944	0.933	0.935	0.759	0.913	0.865	0.935

演算モジュール（$\mathbb{F}_p$ の加算や乗算）の回路規模は標数に，ガロア体乗算器に含まれる素体上の演算モジュールの数は拡大次数によって決定される．Algorithm 1 の計算量は出現するモジュールの数に依存する．ここで，乗算器に出現するモジュールの数は，使用した既約多項式の数にも依存することに注意されたい．

表 3.3 には，各乗算器に使用した既約多項式の項数も示している．表から，検証時間は標数よりも既約多項式の項数に依存して変化することがわかる．例えば，実験中最も既約多項式の項数が少なかった標数 5 の場合は検証時間が少なく，最も既約多項式の項数が大きかった標数 2 の場合は検証時間が長かった．この項数の影響を検証時間から除去するために，検証時間を項数で割った結果を表 3.3 の 5 行目に示す．同一の拡大次数の場合は，正規化を行った検証時間の差がほぼ同等となる事がわかる．

次に，ガロア体乗算器にバグが挿入されていた場合についても実験を行った．仮定したバグは，モジュールの入力エッジが誤ったモジュールの出力エッジに接続されている場合とした．階層的な検証手法である GF-ACG では，グレブナー基底の導出にブッフバーガーのアルゴリズムを用いているため，このような簡単なバグが含まれている場合であっても検証を行うことはできない．表 3.4 にバグが含まれた乗算器の検証時間と，バグありとなしのときの検証時間の比を示す．加えて，図 3.11 には同結果の棒グラフを示す．結果から，バグが含まれている場合のほうが，バグが含まれていない場合よりも少ない時間で検証が終了していることがわかる．これは，提案手法では，各ビットのプライマリ出力変数の正規形を計算するたびに仕様の方程式と比較し，バグが発見された時点で検証を終

表 3.5: GFBMD を用いた場合の検証時間

	$m=64$				$m=128$				$m=256$			
	$p=2$	$p=3$	$p=5$	$p=7$	$p=2$	$p=3$	$p=5$	$p=7$	$p=2$	$p=3$	$p=5$	$p=7$
W/ GFBMDs [s]	2.05	9.77×10^{-1}	6.16×10^{-1}	1.01	1.54×10	1.28×10	4.53	6.70	1.20×10^{2}	1.35×10^{2}	3.42×10	9.58×10
W/o GFBMDs [s]	1.03	5.99×10^{-1}	4.24×10^{-1}	7.06×10^{-1}	2.11×10^{1}	2.67×10^{1}	6.29	1.53×10^{1}	3.15×10^{2}	5.54×10^{2}	1.05×10^{2}	4.80×10^{2}
Ratio	5.03×10^{-1}	6.13×10^{-1}	6.88×10^{-1}	6.99×10^{-1}	1.37	2.09	1.40	2.28	2.63	4.10	3.07	5.01

了するためである．

以上の結果から，提案手法を用いることで拡大次数が 256 のような実用的な乗算器であっても，おおよそ 8 分ほどで機能検証ができることがわかる．従来の GF-ACG においても，多標数のガロア体乗算器の検証手法は提案されているが [67]，同手法では細かく階層的に回路記述を行う必要がある．加えて，GF-ACG で回路記述を行わなければ，検証を実施することができない．これに対して，本手法は，素体上の演算モジュールを最小構成要素とする非階層的な回路記述も検証可能であり，より広範囲の回路記述に適用可能だと考えられる．

■GFBMD を用いた検証実験　次に，GFBMD を用いて検証を行い，その有効性を確認する．検証対象とするのは先程と同様にガロア体乗算器とする．ガロア体乗算器の検証では基本的に，検証時に次数が 2 以上の方程式が出現しないため GFBMD を用いることができる．表 3.5 に GFBMD を用いた場合と，用いなかった場合のそれぞれの検証時間を示す．表から，乗算器の拡大次数が増加するに従って，GFBMD を用いた場合のほうが，用いなかった場合に比べて検証時間が短い傾向にあることがわかる．特に，標数 7，拡大次数 256 の場合では，GFBMD を用いることで，検証時間が約 5 倍高速になっている．この高速化の影響は，GFBMD 上における加算と乗算の計算量に起因している．一般に，多項式の表現にリストによる表現を用いた場合，項数が a と b の多項式の乗算には $O(ab)$ の計算量が必要となる．一方，*BMD では，加算と乗算の計算量が，多項式に出現する変数の数に対して，最悪の場合でそれぞれ線形と指数時間かかることが知られている．ここから，GFBMD も同じ計算量をもつと予想される．しかし，Bryant らは多くの実用的な回路では，そのような指数時間のケースは発生せず，より小さいことを実験的に示している．よって，GFBMD の加算と乗算についても，リスト表現上で計算する場合と比べてより小さい計算量で抑えられている可能性がある．実際，表 3.5 は，その予想を支持している．

最後に，従来の非階層的な検証手法 [53] の適用可能性について考察する．従来手法では，ネットリストや回路仕様の表現にガロア体 $\mathbb{F}_2$ 上の多項式を使用するため，多標数の多項式を $\mathbb{F}_2$ 上の多項式として表現する必要がある．しかし，その項数は標数と拡大次数の増加に伴い爆発的に増加する．例として，表 3.6 に多標数のガロア体乗算器の仕様に出

表 3.6: 各乗算器の仕様を表す $\mathbb{F}_2$ 上の多項式の最大項数

標数 $[p]$	$m=2$	$m=3$	$m=4$
2	3	5	7
3	56	1290	17848
5	1130	94662	2028364

現する，最大の多項式の項数を示す．表から，標数が 2 より大きい場合は，拡大次数の増加に伴って項数が指数的に増加することがわかる．このことから，従来の非階層的な検証手法を多標数ガロア体乗算器へ適用することは困難であると予想される．

3.6　ハードウェアトロイ検知への応用

本節では，前節で提案した等価性検証手法の応用として，暗号ハードウェアに挿入されたハードウェアトロイ（HT: Hardware Trojan）の検知手法を提案する．本節で対象とする暗号ハードウェアは標数が 2 のガロア体算術演算回路に基づくものとし，検知が最も困難な特定の入力が与えられた場合にのみ故障を引き起こす HT が挿入された場合を想定する．本節で提案する手法は，前節までのデータパスを対象とするものと異なり，制御部を含む回路全体に対して適用可能なものである．提案手法は，(i) ZDD を用いた等価性検証による HT 検知と，(ii) 検知された HT の作動条件特定，(iii) HT の挿入箇所特定によって構成される．ここで，(i) はデータパスの入出力が満たすべき多項式が与えられれば，リファレンスとなる回路記述が必ずしも必要ではないことに注意されたい [53]．以降の各節では，これらのステップについて述べる．

3.6.1　等価性検証による HT 検知

本節では，HT 検知のための等価性検証手法について説明する．等価性検証では，組み合わせ回路を対象とし，基本的な流れは 3.4.1 節に従う．組み合わせ回路を対象とした理由の一つは，暗号ハードウェアの大部分は組み合わせ回路からなるガロア体算術演算回路によって構成されるため，HT も組合せ回路部に挿入される可能性が高いためである．事実，AES および ECC ともに組み合わせ回路に故障を入れることで，秘密情報を漏洩させる HT が報告されている [43, 68]．もう一つの理由は，リファレンスとなる回路記述が与えられた場合，暗号ハードウェア全体の検証に容易に本手法を拡張可能なためである．すなわち，リファレンスとなる回路記述とネットリストのレジスタの間に存在するすべての組合せ回路部の等価性を調べることで，ハードウェア全体の等価性検証を示すことができる．このようなシナリオは現実的であり，例えば論理合成ツールが出力したネットリスト

Algorithm 3 等価性検証

Require: Output variables of netlist: $z_1, z_2, \ldots, z_m$, Output variables of specification: $z'_1, z'_2, \ldots, z'_m$

Ensure: Boolean value representing whether they are identical

1: **for** $i = 1$ to m **do**
2: $g_i \leftarrow \text{ComputeZDD}(z_i)$
3: $g'_i \leftarrow \text{ComputeZDD}(z'_i)$
4: **if** $g_i \neq g'_i$ **then**
5: **return** false
6: **end if**
7: **end for**
8: **return** true

と，論理合成前の HDL の間の等価性を調べることで，HT を検知する場合が挙げられる．実際に，第 3.6.4 節では AES ハードウェア全体への適用例を示す．

Algorithm 3 に等価性検証のアルゴリズムを示す．このアルゴリズムは，ネットリストと仕様であるガロア体方程式もしくは回路記述を受け取り，両者の等価性判定結果を出力として返す．ここで，変数 m は出力変数のビット数を表す．提案アルゴリズムでは，1 行目から 7 行目までの for 文で各出力ビットの等価性を順次調べる．2–3 行目の ComputeZDD は，対応する出力ビットの ZDD を構築する関数である．ここで，ZDD のノードを構成する変数は，ネットリスト及びガロア体方程式の入力変数である．そして，4 行目でネットリストと仕様の ZDD の比較を行い，等価でない場合には false を返す．すべての出力ビットの等価性が確認できた場合，8 行目で true を返す．

3.6.2 HT 作動条件特定

本節では，前節の等価性判定で検知された HT の作動条件特定手法について述べる．g_i と g'_i をそれぞれネットリストと仕様の出力を表す ZDD とする．ここで，挿入された HT の影響により $g_i \neq g'_i$ とする．このとき，これらの間の XOR の結果の多項式 $g_i + g'_i$ は，HT の機能を表すことに注意されたい．したがって，HT の作動条件は，$g_i + g'_i$ の表す多項式の充足条件を列挙することで求めることができる．一方で，一般に充足条件の計算は NP 完全であり，加えて HT を表現するブール多項式は極めて巨大になり得るため汎用的な SAT solver により充足条件を求めることは容易ではない．そこで，本稿では FDD (Functional Decision Diagram) の充足条件数え上げアルゴリズムを適用することを提案する [69].

FDD の充足条件数え上げアルゴリズム [69] の基本アイデアは，ZDD の変数順序の低いものから順に論理値を割り当てていき，最終的にブール多項式の値が 1 になった場合に，その割当を充足条件としてみなすというものである．当然，このような充足条件の割当は，変数の数を l とすると，2^l 個の割当を試すことになるため計算量が爆発する恐れがある．そこで，従来手法 [69] では，部分的に割当を行ったブール多項式が 0 とならないことを確認しながら論理値の割当を行う．ZDD の性質を使うことで，部分的な論理値の割当が完全な充足条件へと拡張できる必要十分条件が，その部分割当の結果として得られるブール多項式の値が 0 とならないことであることが示せる．したがって，この枝刈りにより極めて高速に充足条件を求めることが可能となる．実際，文献 [69] では，HT を表す ZDD を g_{diff}，そのグラフサイズを $|g_{\mathrm{diff}}|$，充足条件の数を $\#g_{\mathrm{diff}}$ とすると，$O((l+|g_{\mathrm{diff}}|)\#g_{\mathrm{diff}})$ の計算量ですべての充足条件を求められることを示している．HT の作動条件を求める場合，充足条件の数は極めて少ないと考えられることから，極めて短時間で作動条件を求めることが可能である．

3.6.3　HT 挿入位置特定

本節では，HT の挿入箇所特定手法について述べる．

まず HT の条件として，以下を仮定する．

1. HT の作動確率は極めて低い（例えば，2^{-40} 以下）．
2. HT は一つの XOR ゲートを介して，故障を注入する．

一つ目の条件は，HT が可能な限り見つからないように挿入されることによる．二つ目の条件は最小のペイロード構成からくる．

まず，提案手法を説明するために記号の定義を行う．入力配線を $x_1, x_2, \ldots, x_n$ とし，これらの入力配線によって定義されるブール多項式環を $\mathbb{B}(x_1, x_2, \ldots, x_n)$ とする．ブール多項式環の順序として次数付き辞書式順序 (grlex) を採用する．また，ブール多項式環上の多項式 f について，その次数 $\deg(f)$ を grlex で最大の項の指数の和として定義する．もし $f = 0$ のときは，$\deg(f) = 0$ とする．すべての論理回路は，AND，OR，XOR，NOT などの基本論理素子によって構成され，これらはブール多項式の加算と乗算の組み合わせで表現できる．したがって，ネットリスト上のある配線のブール多項式は，それよりも入力側に存在する配線のブール多項式の加算や乗算の繰り返しの結果として与えられる．これらの加算や乗算は，殆どの場合で多項式の次数を上昇させると考えられる．そこで，本節では入力側に位置する配線の多項式は，より出力側に位置する配線の多項式よりも次数が等しいかより小さいと仮定する．

3.6.2 節と同様に，ネットリストの出力の ZDD を g_i，仕様の出力の ZDD を g_i' とする．

g_i と g'_i は ZDD だが，3.3.2 節で述べたとおりブール多項式とみなすことができる．そこで，これらの ZDD が与える多項式も同じ記号を用いて $g_i(\boldsymbol{x})$，もしくは簡単に g_i のように記述することとする．ここで，$\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ とした．

提案する HT の挿入位置特定手法では，従来手法と同様に，ネットリストから活性化確率が極めて低いゲートを見つけることで HT の挿入箇所を特定する．しかし，ZDD の表現するブール多項式から，そのような充足条件の組み合わせを求めることは#P 完全であることが知られており，事実上不可能である．そこで，提案手法ではブール多項式の次数に着目する．HT のトリガー部の出力の多項式を t とすると，その次数に関して次の定理が成り立つ．

定理 2. g_i と g'_i をそれぞれネットリストと仕様の出力の多項式とする．またトリガー部の出力の多項式を t とする．$\deg(t) > \deg(g'_i)$ のとき，これらの次数の間には次の関係が成り立つ．

$$\max\{\deg(g_i) - \deg(g'_i), \deg(g'_i)\} \leq \deg(t). \tag{3.12}$$

Proof. $\mathcal{C}$ と $\mathcal{C}'$ をそれぞれ，HT が挿入された回路とリファレンス回路とする．定義から，$\mathcal{C}$ と $\mathcal{C}'$ の i 番目の出力の多項式は，それぞれ g_i と g'_i で与えられる．プライマリ出力 g'_i からプライマリ入力までに存在するある配線を w とする．出力側に存在する配線の多項式は，入力側の配線の多項式よりも次数が大きいという仮定から，$\deg(g'_i) \geq \deg(w)$ が成り立つ．ブール多項式環はユークリッド環なので，

$$g'_i = qw + r \tag{3.13}$$

を満たす多項式 q と r が存在する．ここで，$\deg(t) \geq \deg(g'_i) \geq \deg(q)$ かつ $\deg(t) \geq \deg(g'_i) \geq \deg(r)$ である．式 (3.13) は回路 C' のすべての配線で成立するため，HT の挿入箇所を w とできる．この場合，式 (3.13) は

$$g_i = q(w + t) + r \tag{3.14}$$

となる．式 (3.14) の両辺の次数を比較することで，

$$\deg(g_i) \leq \deg(g'_i) + \deg(t)$$

を得る．この不等式と，仮定 $\deg(t) > \deg(g'_i)$ から定理は示される． □

この定理から，多項式の次数を用いることで，ネットリストからトリガー部を絞り込むことができる．一方で，$\deg(t)$ が $\deg(g'_i)$ よりも小さい場合，この定理は意味をなさなくなる．例えば，二入力ガロア体乗算器であれば $\deg(g'_i)$ は高々 2 次，AES の一ラウンドデータパスで次数は最大で 15 次であり，$\deg(t)$ はそれよりも大きい必要がある．このとき，次の定理からトリガー部の次数は極めて大きいことが保証できる．

定理 3. ブール多項式 $f(\boldsymbol{x})$ に対して，関数 $C_v(f) = |\{\boldsymbol{x} \mid f(\boldsymbol{x}) = v, \boldsymbol{x} \in \mathcal{X}\}|/|\mathcal{X}|$ を定義する．$\mathcal{X}$ は関数 f が取りうる入力のブール値の集合である．ここで，次が成り立つ．

$$\min\{C_0(f), C_1(f)\} \geq 2^{-\deg(f)}.$$

Proof. 多項式 f_1 を $f_1 = 1 + f$ とする．$C_0(f) = C_1(f_1)$ と $\deg(f) = \deg(f_1)$ が成り立つ．よって $C_1(f_1) \geq 2^{-\deg(f_1)}$ だけを示せば良い．

変数の数 n に関する数学的帰納法で示す．$n = 1$ のときは，$C_1(f_1) \geq 2^{-\deg(f_1)}$ は明らかに成り立つ．次に $n > 1$ のときを考える．f_1 の x_n に関するシャノン展開は

$$\begin{aligned} f_1(x_1, \ldots, x_n) = {} & x_n g(x_1, \ldots, x_{n-1}) \\ & + (x_n + 1) h(x_1, \ldots, x_{n-1}) \end{aligned}$$

である．ここで，$g(x_1, \ldots, x_{n-1})$ と $h(x_1, \ldots, x_{n-1})$ は $n-1$ 変数の多項式である． 帰納法の仮定から，$C_1(g) \geq 2^{-\deg(g)}$ かつ $C_1(h) \geq 2^{-\deg(h)}$ が成り立つ．よって

$$\begin{aligned} C_1(f_1) &= \frac{1}{2}(C_1(g) + C_1(h)) \\ &\geq \frac{1}{2}(2^{-\deg(g)} + 2^{-\deg(h)}) \\ &\geq \frac{1}{2}(2^{-\deg(f_1)} + 2^{-\deg(f_1)}) \\ &= 2^{-\deg(f_1)} \end{aligned}$$

である． □

この定理から，HT の検出率を下げるためには多項式の次数を絶対にあげなければならないことがわかる．例えば，モンテカルロテストに対する検出率を 2^{-40} 以下にするためには，トリガー部の出力のブール多項式の次数は 40 以上でなければならない．これは，ガロア体乗算器や AES のラウンドデータパスよりも大きく，よって式 (3.12) により絞り込むことができる．

アルゴリズム 4 に提案する HT 挿入位置特定手法の擬似コードを示す．同アルゴリズムでは，ネットリストと仕様の ZDD である g_i と g'_i に加えて，すべてのゲート出力の ZDD $w_1, w_2, \ldots, w_m$ を受け取る．ここで，m はゲートの数を表し，各インデックスはより入力に近い配線ほどインデックスの値が小さいものとする．このアルゴリズムは式 (3.12) を満たすゲートのリスト L を出力する．トリガー出力の次数がネットリストのプライマリ出力に至るまで減少しないと考えられることから，リスト L にはその間に含まれるすべてのゲートが加えられる．一方で，実際に HT のトリガーの出力を構成するであろうゲートは，このリストの中でインデックスの値が小さいものであると考えられる．よって，そのようなゲートを確認することで HT のトリガー部を特定することができる．

Algorithm 4 HT 挿入箇所特定アルゴリズム

Require: ZDDs of netlist and specification: g_i, g'_i, ZDDs of all gate outputs: $w_1, w_2, \ldots, w_m$

Ensure: Suspicious gate list: L

1: $L \leftarrow \emptyset$
2: **for** $j = 1$ to m **do**
3: **if** $\max\{\deg(g_i) - \deg(g'_i), \deg(g'_i)\} \leq \deg(w_j)$ **then**
4: $L \leftarrow L \cup \{j\}$
5: **end if**
6: **end for**
7: **return** L

3.6.4 実験

本章では，ECC と AES ハードウェアのネットリストの HT 検知およびその特定を行うことで，提案手法の有効性を確認する．具体的には，ECC 向けのガロア体乗算器の検証及び，AES ハードウェア全体の検証に提案手法を適用する．

■ガロア体乗算器の HT 検知 本節では，ECC 向けのガロア体乗算器に対する提案手法の適用結果を示す．検証対象の乗算器は，ガロア体乗算器ジェネレータ [65] によって生成された Mastrovito 乗算器とし，Synopsys 社の Design Compiler による論理合成結果として得られるネットリストを用いた．パラメータ等は，基本的に NIST 推奨のものに準じた[*2]．また，実験に用いた環境は CPU が Intel Xeon Gold6144 3.5GHz で，メモリは 756GB である．本節では，提案手法の有効性を確認するために，HT が挿入された乗算器を対象とする．ここで対象とする HT は，乗算器に特定の入力が与えられた場合にのみ故障を引き起こすものである．この故障を引き起こす入力ペアを攻撃者が知っている場合，Bug Attack と呼ばれる攻撃により ECC から秘密鍵が漏洩することが知られている[*3]．本節の実験では，比較対象となる ZDD はガロア体乗算の方程式からビットレベルの方程式を導出することで得られるため，従来の回路検証で必要なリファレンス回路記述は不要であることに注意されたい．

図 3.12 に HT 検知および作動条件と位置特定にかかった時間を示す．提案手法では，

[*2] 64 ビットと 128 ビットについては，パラメータが与えられていないため既存手法に準じた [53].

[*3] Bug Attack を示した文献 [45] では，主に素体上の RSA もしくは ECC に対する適用例のみが示されているが，これは拡大体上の ECC へ適用可能であると考えられる．

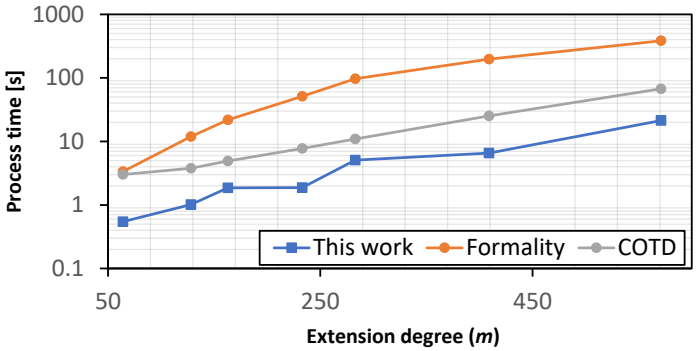


図 3.12: HT が挿入された乗算器の検知時間

表 3.7: それぞれの手法の実行時間ならびに特定されたゲートの数

	#Gates	提案手法		Formality		COTD	
		Time [s]	#SusGates	Time [s]	#SusGates	Time [s]	#SusGates
64	7,172	0.545	3	3.35	6	3.01	3509
128	27,234	1.01	1	11.9	11	3.77	14033
163	43,788	1.86	3	21.9	5	4.93	42191
233	87,761	1.88	11	51.1	6	7.76	6202
283	136,492	5.08	5	96.7	7	10.9	9019
409	271,753	6.53	9	198	6	25.2	16972
571	535,643	21.3	8	384	6	67.4	39031

与えられたネットリストとリファレンスとなる多項式の間の等価性検証により HT の検出を行った．比較のために，Synopsys 社の等価性検証ツールである Formality と，ヒューリスティックな HT 検知手法の一つである COTD の結果も示した [70]．Formality は，等価性検証時に “report_error_candidates” コマンドを用いることで故障を引き起こしていると推定されるゲートの候補も出力することができる．COTD は，SCOAP の可制御性と可観測性を利用したヒューリスティックな HT 検知手法である．本実験では，従来手法と同様にネットリストの可制御性および可観測性を Synopsys 社の Tetramax によって抽出し，Python のオープンソースライブラリである “Scikit learn” を用いて K-means クラスタリングを用いて，COTD を実装した．図 3.12 から，提案手法は他の 2 つの手法と比べて，より高速に HT の検知が行えることが読みれ取れる．特に，ヒューリスティックな手法である COTD よりも高速に等価性検証が行えることから，提案手法は暗号ハードウェアを構成するガロア体算術演算回路に対して特に適していると思われる．

表 3.7 は，HT の挿入位置特定によって挙げられたゲートの候補の数を示している．同表において，#Gates は回路を構成するすべてのゲートの数を表し，#SusGates はそれぞれの手法によって疑わしいと推定されたゲートの数が示されている．結果から，COTD によって推定されたゲートの数は極めて大きく，回路全体を構成するゲートに近いオー

表 3.8: AES に対する HT 検知にかかった時間

等価性検証	作動条件特定	挿入位置特定
2.91 s	105 μs	21.8 ms

ダーとなっていることがわかる．ここから，COTD によって推定されたゲートの殆どは，実際には HT とは無関係なものであると考えられる．一方で，Formality によって出力されたゲートの候補数は，提案手法のものに極めて近いものとなっている．しかし，実際に両者の間のゲートを比較すると全く一致しておらず，一方の手法では不適切なゲートが出力されていると予想される．ここで，提案手法によって推定されたゲートのリストには，定理 1 から必ずトリガーの出力が含まれていなければならないことを考えると，Formality の出力結果は HT の検知において適切ではないものであると思われる．

以上の結果から，暗号ハードウェアを構成するデータパスに対する HT 検知において，提案手法は従来手法と比べより適していると確認できる．

■AES ハードウェアに対する HT 検知　次に，より実用的な例として図 3.13 に示す AES ハードウェアに対して提案手法を適用した結果を示す．本実験で用いた AES は S-box に合成体を利用したラウンド実装のものとし，文献 [43] で示された故障注入攻撃を利用した HT が挿入された場合を対象とした．簡単のため，同図では制御部のロジックを省略している．図中の赤で示された部分が挿入された HT である．本実験で挿入された HT は，特定の平文が与えられると 8 ラウンド目の MixColumns の入力にビットフリップ誤りを挿入するものである．提案手法による HT 検知にあたって，本実験では検証対象となる AES のネットリストに加えて，リファレンスとなる AES の設計データが与えられると仮定した．ここで，これらのデータの間のレジスタの対応関係は既知であるとする[*4]．したがって，それぞれのデータのレジスタの間に存在する組み合わせ回路の等価性を示すことで，HT の有無を判定することができる．このようなシナリオは，例えば論理合成ツールが HT を挿入するような場合や，ソフトウェア IP に詳細な仕様が与えられている場合に相当する．

表 3.8 に等価性検証と作動条件の特定にかかる時間を示す．提案手法を用いることで，HT の検知及び，その作動条件の特定を約 3 秒で行えることがわかる．特に，作動条件の特定は等価性検証などに比べると，無視できるほどの時間で実行可能なことが確認できる．また，提案手法によって HT の一部として推定されたゲート数は 86 個であった．この内，最も入力側に位置するゲートは実際に故障を注入している XOR ゲートであった．

[*4] この仮定から Trust-Hub に公開されているような，レジスタを追加するタイプの HT は本実験の対象外であることに注意されたい．

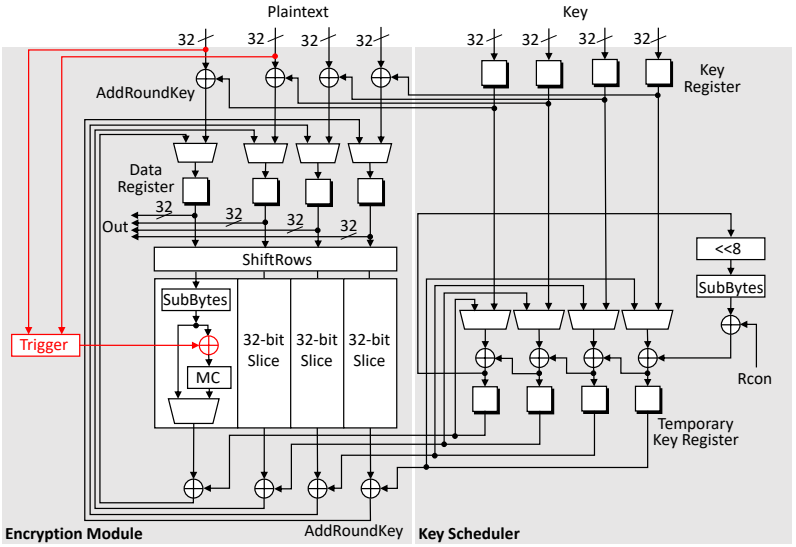


図 3.13: HT が挿入された AES の回路図

この結果から，提案手法は暗号ハードウェアに実際に挿入されうる HT の検知が可能であることを確認できる．

3.7　結び

本章では，設計時の論理的な安全性評価方法として，与えられた暗号ハードウェアのゲートレベルネットリストと設計仕様の間の等価性検証手法を提案した．提案手法は，(1)ZDD に適した簡約アルゴリズムの提案と，(2) 従来の等価性検証手法の多標数ガロア体算術演算回路への拡張の 2 つからなる．また，提案手法の応用として，暗号ハードウェアに挿入された HT の検知を行い，実際に HT が挿入された AES の完全検証を実施した．提案手法を用いることで，暗号ハードウェアに含まれた脆弱性を検知することが可能である．

第 4 章

共通鍵暗号モジュールの物理攻撃に対する安全性評価手法

4.1 はじめに

本章では，サイドチャネル攻撃に焦点を当て，深層学習を用いた物理的安全性評価について述べる．第 1 章で述べたとおり，DL-SCA による暗号モジュールの安全性評価を実施するにあたっては，DL-SCA の適用可能限界を正確に把握する必要がある．そのためには，DL-SCA の攻撃能力を最大限活かした上で，攻撃評価を実施しなければならない．一方で，DL-SCA による共通鍵暗号モジュールへの攻撃では，不均衡データ問題による性能劣化の問題が指摘されていた [71]．これは，DL-SCA の脅威の適切な評価を困難にしている．そこで，本章では，この不均衡データ問題の原因の解明と，その解消法について述べる．まず，DL-SCA における不均衡データ問題の関連研究について説明する．次に，不均衡データ問題の原因と，その定量的評価方法として，カルバックライブラー（KL: Kullback Leibler）ダイバージェンスを利用した指標を提案する．また，近年提案された不均衡データ向けのロス関数である CER (Cross Entropy Ratio) ロスについて，提案指標の観点から，なぜ有効に働くのかを述べる．更に本章では，不均衡データ問題の解消法として，鍵の尤度を使用した解決策を提案する．最後に，実際のオープンデータセットを利用した実験を通して，提案手法の有効性を実証する．

4.2 関連研究

深層学習を用いたサイドチャネル攻撃（DL-SCA）は，プロファイリング型サイドチャネル攻撃の一種である．典型的な同攻撃のプロファイリングフェーズでは，入力がプロファイリングデバイスの漏洩サイドチャネル情報，出力が暗号計算の中間値（例えば 1 ラウンド目の S-box の出力）の出現確率となるディープニューラルネットワーク (DNN) を

学習させる．そして，攻撃フェーズでは学習された DNN の出力確率の対数尤度を用いて秘密情報を推定する．しばしば，学習時の複雑度を軽減させる目的で，DNN の出力は中間値そのものではなく，そのハミングウェイト (HW) やハミングディスタンス (HD) をラベルとして利用する [71]．DNN を用いたプロファイリング攻撃では，上述のテンプレートアタックと異なり使用する波形や中間値に特別な仮定を置く必要がないという利点があり，その有効性が実験的に示されている [72]．

一方で，DNN を用いたプロファイリングサイドチャネル攻撃では，学習に用いられる HW や HD が二項分布に従う不均衡であることによる学習の困難性が指摘されている [71]．文献 [71] では，従来の機械学習で使われてきた精度や適合度などの指標がクラスの不均衡によって DL-SCA では有効ではないことが示されている．この問題の解決を目的として，同文献では汎用的なデータ拡張手法の一つである SMOTE を用いた，各クラスの出現頻度の均一化が有効であることを実験的に示した．しかしながら，データの真の分布は通常不明であるため，SMOTE のようなデータ拡張では，低品質なサンプルが追加され，むしろ精度を悪化させる可能性がある．また，データ拡張によって少数派クラスのデータ数を人工的に増やすことで，学習時と攻撃時のデータの分布に差が生じるが，これが SCA の攻撃成功確率（SR: Success Rate）に与える解析的な影響は不明である．これらの理由から，SCA のプロファイリングフェーズにおけるデータ拡張が攻撃時に与える影響を調べることは必須である．

不均衡データ問題を解決するための別のアプローチとして，文献 [73] では，DL-SCA のための新たな評価指標である CER が報告された．CER は，正解鍵の NLL (Negative Log Likelihood) をそれ以外の鍵候補の NLL の平均で割ったものである．同文献では，CER が 1 未満であるときに十分な波形数があればプロファイリング攻撃が必ず成功することを証明した．また，CER により，不均衡データによる問題が解決されることも実験的に示している．しかし，同文献で示された CER の有効性を示すための証明に用いられた仮定には，一般的には成立しない誤りが含まれている．ここで述べた未解決問題は，DL-SCA を用いたサイドチャネル攻撃に対する脆弱性の正確な把握や対策の開発を致命的に困難にしており，そのメカニズムを調べることは極めて重要である．

4.3　不均衡データによる悪影響の解析と定量的評価方法

本節では，サイドチャネル攻撃における不均衡データによる悪影響の原因について述べ，その定量的評価方法について説明する．また，評価方法として，機械学習において一般的なカルバックライブラーダイバージェンスを導入し，それによる不均衡度合いの評価について述べる．

DL-SCA ではデータセットに以下の二つの特徴がある．

1. サイドチャネル情報には秘密情報に関する十分な情報が含まれている（SNR（Signal Noise Ratio） が高い）とは限らない．
2. HW/HD が二項分布に従った不均衡な分布である．

これらの特徴は DL-SCA というタスク特有の特徴であり，二つ目の特徴に関しては既に先行研究で指摘されている．一方，一つ目の特徴に関してはサイドチャネル攻撃では常識ではあるものの，DL-SCA という文脈で二つ目の特徴と関連付けてこれまで議論されていない．

まず，従来の機械学習における画像認識や音声認識などの分類タスクでは与えられた入力に分類に十分な情報が含まれていることが多い．これらのタスクでは，人間によるアノテーションによってラベルが付与され，その意味で分類に十分な判断基準が存在することが期待される．一方，SCA では，アルゴリズミックノイズやサイドチャネル攻撃に対する対策，測定時の SNR などの影響により，サイドチャネル波形から暗号演算の中間値が一意に特定することは容易ではない．そのため，単一のサイドチャネル波形には HW/HD の値を推定するための十分な情報が含まれていないと考えられる．もしサイドチャネル情報と対応する中間値が統計的にほとんど独立に近い場合，モデルによって模倣される条件付き確率 $q(z \mid \boldsymbol{x})$ が近似的にラベルの正規確率 $q(z)$ となる．そして，このラベルの正規確率は HW/HD の影響で二項分布に従うため，推定すべき真の条件付き確率分布 $q(z \mid \boldsymbol{x})$ は近似的に二項分布 $\mathrm{Bin}(z)$ に等しくなる．すなわち，ニューラルネットワークの出力分布は，二項分布に強くバイアスされ，DL-SCA をより難しくしている．

二項分布に従うバイアスが DL-SCA をどのように難しくしているかを説明するために，AES の 1 ラウンド目の S-box の出力を中間値として使用した場合を例として述べる．この場合，ニューラルネットワークの出力分布 $p(\psi(k, m_i) \mid \boldsymbol{x}_i; \theta)$ は，サイドチャネル波形 $\boldsymbol{x}_i$ が与えられたときの中間値の HW の条件付き確率を表す．ここで，HW は 0 から 8 までの値を取り，m_i と $\boldsymbol{x}_i$ はそれぞれ i 番目の観測次の平文とサイドチャネル波形である．プロファイリング攻撃において正解鍵 k^* を推定するためには，モデルの出力分布の正解鍵のラベル $\psi(k^*, m_i)$ の確率値が大きくなければならない．しかし，中間値とサイドチャネル波形の関係性が弱く，モデルの出力分布が二項分布に近い場合，常に最頻値に当たる $z = 4$ の確率が最大で，$z = 0, 8$ の確率が最小になる．これは，正解ラベルが 0 や 8 などの場合でも，二項分布の最頻値である $z = 4$ に対応する不正解鍵の確率（尤度）を不当に高く見積もることにつながる．このような影響は，攻撃に使用できる波形数が無限にあるような理想的な場合では，すべての鍵候補が同じだけ二項分布のバイアスの影響を受けるので問題とならない．ただし，現実的には無限の波形数を用いることは不可能であり，結果的に攻撃に必要な波形数の増加につながる．

4.3.1　定量的評価方法

本節では，不均衡データによる悪影響の度合いを計るための指標を提案する．前節で述べたとおり，サイドチャネル攻撃への DL の適用が難しい根本的理由は，サイドチャネル情報とラベルの間の関係性が弱いこと，そしてラベルの生起確率が二項分布に従うことであることを述べた．この二つの影響を定量的に調べるには，条件付き確率 $q(z \mid \boldsymbol{x})$ とラベルの生起確率 $\mathrm{Bin}(z)$ の間の近さ（距離）を測ることが効果的である．そこで，本稿では，確率分布間の距離を計るのに汎用的に使われる指標として Kullback-Leibler (KL) ダイバージェンスを用いた次の評価指標を提案する．[*1]

$$D_{\mathrm{KL}}(\mathrm{Bin}(Z) \mid\mid q(Z \mid \boldsymbol{X})) = \mathbb{E}_{\boldsymbol{X}} \mathbb{E}_{Z} \log \left(\frac{\mathrm{Bin}(Z)}{q(Z \mid \boldsymbol{X})} \right). \tag{4.1}$$

式 (4.1) における確率分布 $q(z \mid \boldsymbol{x})$ は未知であり直接評価することはできない．そこで，確率 $q(z \mid \boldsymbol{x})$ を NN による推定確率 $p(z \mid \boldsymbol{x}; \theta)$ に置き換える．加えて，サイドチャネル情報 $\boldsymbol{X}$ に対する期待値は，プロファイリングフェーズで入手した有限のサンプル点で近似する．これらの近似により，式 (4.1) は，

$$\hat{D}_{\mathrm{KL}}(\mathrm{Bin}(Z) \mid\mid p(Z \mid \boldsymbol{X}; \theta)) = \frac{1}{n_P} \sum_{i=1}^{n_P} \sum_{z} \mathrm{Bin}(z) \log \left(\frac{\mathrm{Bin}(z)}{p(z \mid \boldsymbol{x}_i; \theta)} \right) \tag{4.2}$$

となる．ここで，二項分布は推定する中間値のビット長で変化し，中間値が r ビットのとき

$$\mathrm{Bin}(z) = \frac{\binom{r}{z}}{2^r}$$

である．ここで，分子は二項係数を表す．

KL ダイバージェンスは 2 つの確率分布を受け取り 0 以上の値を返す関数であり，2 つの確率分布が同一のときに限り 0 を返す．したがって，式 (4.1) の KL ダイバージェンスが小さいとき，条件付き確率 $q(z \mid \boldsymbol{x})$ がラベルの生起確率に近いことを表す．実際，Z と $\boldsymbol{X}$ をそれぞれラベルと波形の確率変数とすると，KL ダイバージェンスが 0 のとき，$q(z \mid \boldsymbol{x}) = \mathrm{Bin}(z) \Leftrightarrow Z \perp\!\!\!\perp \boldsymbol{X}$ となり，原理的に DL-SCA は絶対に成功しない．一方，モデル出力がワンホットベクトルのような分布のとき，KL ダイバージェンスは非常に大きい値を持つ．このように，この KL ダイバージェンスに基づく提案評価指標を用いることで，サイドチャネル攻撃の容易さ・困難さを見積もることができる．

[*1] 通常は，$D_{\mathrm{KL}}(\mathrm{Bin}(Z) \mid\mid q(Z \mid \boldsymbol{X})) = \mathbb{E}\left[\mathrm{Bin}(Z)/q(Z \mid \boldsymbol{X}) \mid \boldsymbol{X}\right]$ と定義されるが，ここでは二項分布との距離を測る必要があるため，異なる定義を採用する．

4.4 KL ダイバージェンスを用いた CER ロスの解析

本節では，CER ロスが不均衡データ問題を緩和する理由を，KL ダイバージェンスの観点から説明する．文献 [73] は，CER ロスが不均衡データに対して有効である理由を，CER < 1 のとき SCA が必ず成功することにあると述べた．しかし，この証明に用いられた仮定の一つである，正解鍵と不正解鍵の中間値が独立であるという仮定は一般には成り立たない [74, 75]．言い換えれば，CER ロスが不均衡データに有効である他の理由が存在しなければならない．これを説明するために，まずモデル出力と二項分布間の KL ダイバージェンスを増加させるような学習が，不均衡データによる悪影響を軽減すること述べる．次に，CER ロスが KL ダイバージェンスを増加させるようなロスとなっていることを説明する．

4.4.1 KL ダイバージェンスの増加の影響

4.3 節で，DL-SCA における不均衡データ問題は，ラベルの正規確率が二項分布に従って歪んでおり，その結果不正解ラベルの確率を不当に高く見積もることが原因であると述べた．言い換えれば，この悪影響は二項分布によるモデル出力の偏りを取り除くように学習させることで解決できる可能性を示唆している．そのような学習は，従来の NLL ロスの最小化に加えて，KL ダイバージェンスを増加させるように学習することで達成できる．

これを詳しく見るために，式 (4.2) を変形した次の式に着目する．

$$\hat{D}_{\mathrm{KL}}(\mathrm{Bin}(Z) \mid\mid p(Z \mid \boldsymbol{X}; \theta)) = \sum_{z \in \mathcal{Z}} \mathrm{Bin}(z) \log \mathrm{Bin}(z) - \frac{1}{n_P} \sum_{i=1}^{n_P} \sum_{z \in \mathcal{Z}} \mathrm{Bin}(z) \log p(z \mid \boldsymbol{x}_i; \theta). \quad (4.3)$$

式 (4.3) の第一項はモデルに依存しない定数項であるため，第二項に注目する．対数関数が単調増加関数であり引数の大小関係がそのまま出力に反映されることを考慮すると，第二項の値はモデル出力 $p(z \mid \boldsymbol{x}_i; \theta)$ の値が小さいほど大きくなることがわかる．特に二項分布 $\mathrm{Bin}(z)$ との内積により，最頻値に当たる $\mathrm{HW} = 4$ のラベルほど強くこの影響を受ける．これは，学習時に KL ダイバージェンスが大きくなるように学習させることが，クラスの不均衡の影響を軽減することを意味する．ここで，すべてのラベルに関する確率の和が 1 になる制約から，ラベルに関する出力確率が同時にすべて 0 になることはないことに注意されたい．

4.4.2　CER ロスと KL ダイバージェンスの関係

本節では，CER ロスと KL ダイバージェンスの関係について述べる．そのためにまず，CER ロスの定義を行う．まず波形の確率変数を $\boldsymbol{X}$，秘密鍵の確率変数を K，平文の確率変数を M とする．また中間値の HW/HD の確率変数を $Z = \psi(K, M)$ とし，鍵として k を使用したときの中間値の HW/HD を $Z^{(k)} = \psi(k, M)$ とする．波形が与えられたときのラベルの条件付き確率を $q(z \mid \boldsymbol{x})$ とし，ニューラルネットワークの表現する条件付き確率分布を $p(z \mid \boldsymbol{x}; \theta)$ とする．さらに k^* を秘密鍵とする．

DL-SCA では秘密鍵として k^* を使用している暗号モジュールに対して，各鍵候補 k を正解と仮定して尤度（クロスエントロピー）を計算し，その大小比較で秘密鍵の推定を行う．鍵候補 k を正解と仮定して計算されるクロスエントロピーを

$$\mathrm{CE}_k(p) = -\mathbb{E} \log p(Z^{(k)} \mid \boldsymbol{X}; \theta) \tag{4.4}$$

と定義する．$\mathrm{NLL}_k(p)$ の期待値は式 (4.4) に等しいため，波形数が無限のときに $\mathrm{NLL}_k(p)$ は式 (4.4) に確率収束することに注意されたい．各鍵候補に関するクロスエントロピー $\mathrm{CE}_k(q, p)$ を用いることで，CER は

$$\mathrm{CER}(p) = \frac{\mathrm{CE}_{k^*}(p)}{\mathop{\mathbb{E}}_{k \neq k^*} \mathrm{CE}_k(p)} \tag{4.5}$$

と定義できる．

この定義をもとに，KL ダイバージェンスの観点から CER ロスの最小化に関して解析する．式 (4.5) の分子は従来のクロスエントロピーロスに等しいため，ここでは分母に着目する．分母の期待値は鍵に関する平均に等しいので，

$$\begin{aligned} \mathop{\mathbb{E}}_{k \neq k^*} \mathrm{CE}_k(p) &= \frac{1}{|\mathcal{K}| - 1} \sum_{k \neq k^*} \mathrm{CE}_k(p) \\ &= \frac{|\mathcal{K}|}{|\mathcal{K}| - 1} \mathop{\mathbb{E}}_{k} [\mathrm{CE}_k(p)] - \frac{1}{|\mathcal{K}| - 1} \mathrm{CE}_{k^*}(p). \end{aligned} \tag{4.6}$$

ここで，$\mathcal{K}$ は鍵候補の集合であり，$|\mathcal{K}|$ は鍵候補の数を表す．式 (4.6) の第一項の $|\mathcal{K}|/(|\mathcal{K}| - 1)$ はほぼ 1 に一致する一方で，第二項の $1/(|\mathcal{K}| - 1)$ は非常に小さくほぼ無視できる．したがって，式 (4.6) は

$$\mathop{\mathbb{E}}_{k \neq k^*} \mathrm{CE}_k(p) \approx \mathbb{E} \mathrm{CE}_K(p) \tag{4.7}$$

と近似できる．式 (4.7) の右辺に注目すると，

$$\mathrm{CE}_k(p) = -\mathbb{E} \log p(Z^{(k)} \mid \boldsymbol{X}; \theta) = -\mathbb{E} \log p(\psi(k, M) \mid \boldsymbol{X}, \theta)$$

である．ここから，各鍵候補に関するクロスエントロピーの平均は

$$\begin{aligned}\mathop{\mathbb{E}}_{k}\mathrm{CE}_k(p) &= -\mathbb{E}\left[\mathbb{E}\left[\log p(\psi(K,M) \mid \boldsymbol{X};\theta) \mid K\right]\right] \\ &= -\mathbb{E}\frac{1}{|\mathcal{K}|}\sum_{k\in\mathcal{K}}\log p(\psi(k,M) \mid \boldsymbol{X};\theta) \end{aligned} \tag{4.8}$$

となる．ここで，鍵候補の集合の分割を

$$\mathcal{K} = \bigcup_{z\in\mathcal{Z}}\mathcal{K}_{z,m} = \bigcup_{z\in\mathcal{Z}}\{k \mid z = \psi(k,m)\} \tag{4.9}$$

のようにおく．鍵候補の集合の分割 $\mathcal{K}_{z,m}$ は，任意の平文 m に対して，式 (4.9) が成立する．また，$|\mathcal{K}_{z,m}| = \binom{r}{z}$ であり，r は中間値のビット長を表す．この分割を用いて式 (4.8) を

$$\begin{aligned}\mathbb{E}\mathrm{CE}_K(q,p) &= -\mathbb{E}\frac{1}{|\mathcal{K}|}\sum_{z}\sum_{k\in\mathcal{K}_{z,M}}\log p(\psi(k,M) \mid \boldsymbol{X};\theta) \\ &= -\mathbb{E}\sum_{z}\frac{|\mathcal{K}_{z,M}|}{|\mathcal{K}|}\log p(z \mid \boldsymbol{X};\theta) \\ &= D_{\mathrm{KL}}(\mathrm{Bin}(Z) \mid\mid p(Z \mid \boldsymbol{X};\theta)) + H(\mathrm{Bin}(Z))\end{aligned}$$

と変形する．$H(\mathrm{Bin}(Z))$ は二項分布のエントロピーである．以上より，CER は

$$\mathrm{CER}(q) \approx \frac{\mathrm{CE}(p)}{D_{\mathrm{KL}}(\mathrm{Bin}(Z) \mid\mid p(Z \mid \boldsymbol{X};\theta)) + H(\mathrm{Bin}(Z))}$$

と近似される．以上より，CER ロスの最小化は，二項分布とモデル間の KL ダイバージェンスの最大化に等しく，CER ロスが不均衡データに対して有効な理由となる．

4.5 推論時の不均衡データ問題の解消

本節では，不均衡データの悪影響が大きい場合（すなわち推論時の NN の出力分布が二項分布に近い場合）の攻撃フェーズでの解決策を提案する．その基本アイデアは，従来使われてきた HW/HD の尤度関数ではなく鍵の尤度関数を鍵推定に用いることである．前節で述べた KL ダイバージェンスを強制的に増加させるような不均衡データの解決策は，モデルの出力分布を真の分布から遠ざけてしまう可能性があるため，機械学習の観点からは必ずしも最善であるとは限らない．一方で，本章で提案する鍵の尤度の使用は，機械学習の観点においても理論的に妥当な不均衡データ問題の解決策と言える．この鍵値の尤度関数は，HW/HD に関して学習を行った NN から求めた鍵値ごとの確率から導出できる．ここで，提案手法では鍵の尤度関数 (本稿では KNLL と表記) を用いて NN の学習および推論を行うが，モデルは依然として中間値の HW/HD を推論する（AES の場合は 9 ク

ラス推論を行う）ことに注意されたい．本章では，鍵値の尤度関数を用いる提案手法が，ラベルの出現確率を均等にする学習フェーズにおけるデータ拡張と近い働きを持つことも示す．

4.5.1　鍵の尤度関数による推定

本節では，従来の HW/HD の尤度関数に基づく鍵推定の問題について述べた上で，鍵の尤度関数に基づく鍵推定の提案を行う．第 4.3.1 節で示した KL ダイバージェンスは，推定された確率分布に含まれる鍵の推定に必要な情報を判定するための指標であるとともに，その定義・性質から出力分布と二項分布との距離を表現する．言い換えれば，KL ダイバージェンスが 0 に近く NN がラベルを正確に予測できていない場合は，NN の出力分布は二項分布とほぼ等しいと考えられる．しかし，式 (2.4) による鍵の推定では，この二項分布の影響が全く加味されず，しばしば推論に多大な悪影響を与える．

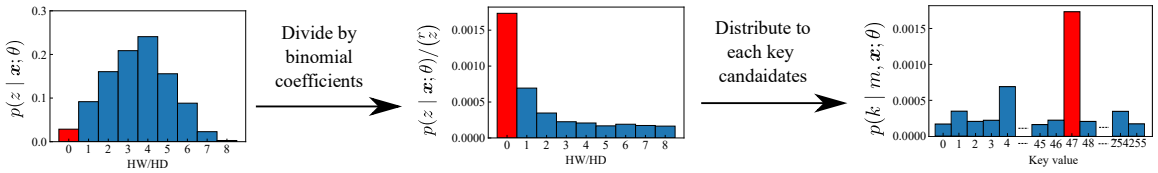
提案手法では，上記の悪影響を解消するため，鍵の推定時に次式で定義される鍵の NLL を使用する．

$$\mathrm{KNLL}_k(q) = -\frac{1}{n_A}\sum_{j=1}^{n_A}\log q(k \mid m_j, \boldsymbol{x}_j). \tag{4.10}$$

ここで，k は鍵候補を表す．鍵は一様分布であるため，鍵の NLL であれば波形に十分な鍵に関する情報が含まれていない場合でも不均衡データによって歪むことがないため，効率的に鍵の推定ができると考えられる．しかし，式 (4.10) の計算には，波形が与えられたときの鍵値の事後確率が必要であり，このままでは NN の推論結果から直接推定できない．そこで式 (4.10) を条件付き確率を用いて次のように変形する．

$$\begin{aligned}
\mathrm{KNLL}_k(q) &= -\frac{1}{n_A}\sum_{j=1}^{n_A}\log q(k, z_j \mid m_j, \boldsymbol{x}_j) \\
&= -\frac{1}{n_A}\sum_{j=1}^{n_A}\log q(k \mid m_j, z_j, \boldsymbol{x}_j) q(z_j \mid \boldsymbol{x}_j) \\
&= -\frac{1}{n_A}\sum_{j=1}^{n_A}\log q(k \mid m_j, z_j) q(z_j \mid \boldsymbol{x}_j).
\end{aligned}$$

ここで，鍵値と波形のラベルに対する条件付き独立性を仮定する．すなわち，ラベルが与えられたときに，鍵値と波形は独立とみなす．条件付き確率 $q(k \mid m, z)$ は，ある HW/HD が与えられたときに鍵値が k である確率を表し，二項分布の逆数に比例した分布となる．



(a) NN output　(b) Divided NN output　(c) Key value probability

図 4.1: 提案手法の概要： (a) NN の分布および， (b) NN の分布を二項係数で割った値，(c) 鍵値の確率

したがって，

$$q(k \mid m, z) = \begin{cases} \frac{1}{\binom{r}{\psi(k,m)}} & (z = \psi(k,m)) \\ 0 & (\text{otherwise}) \end{cases}.$$

波形が与えられたときのクラスラベルに関する条件付き確率 $q(z_j \mid \boldsymbol{x}_j)$ を，モデルの出力の分布 $p(z_j \mid \boldsymbol{x}_j; \hat{\theta})$ によって近似することで，次式が得られる．

$$\mathrm{KNLL}_k(p) \approx \mathrm{KNLL}_k(\hat{\theta}) = -\frac{1}{n_A} \sum_{j=1}^{n_A} \log \frac{p(z = \psi(k, m_j) \mid \boldsymbol{x}_j; \hat{\theta})}{\binom{r}{\psi(k,m_j)}}. \tag{4.11}$$

従来の HW/HD の NLL と比較して，式 (4.11) は二項係数による除算を含むことで，モデルの出力の二項分布へのバイアスを取り除くことができる．したがって，(4.11) は攻撃が難しいようなデータセットほど，効率的に鍵を取得できることが期待される．

図 4.1 に，正解ラベルが 0 で，その中間値が 47 のときの提案手法の流れを示す．理想的には，ニューラルネットワークの出力は正解ラベルの確率が他のラベルと比べて高いことが望ましい．しかし，実際には二項分布に従うバイアスによって，図 4.1(a) に示すように正解ラベル（赤）が必ずしも最大とはならない．これに対処するため，提案手法ではモデルの出力した確率分布を二項係数によって割る（図 4.1(b)）．次に，この除した確率値を，各鍵候補の確率値として分配する（図 4.1(c)）．ここで，分配された確率値の和が 1 になることに注意されたい．言い換えれば，この分配された確率は，どの鍵候補が正解であるかを表す確率であるとみなせる．最後に，この確率をもとに鍵の尤度 KNLL_k を計算し，正解鍵の推定を行う．

Algorithm 5 に提案手法の擬似コードを示す．同アルゴリズムは，入力としてプロファイリングと攻撃フェーズ用のデータセットを受け取り，正解鍵の推定値 k' を出力する．まず，1 行目で従来の HW/HD の尤度に従いモデルの学習を行う．次に，2–5 行目では，2 つの変数 KNLL_k と，invcoef の初期化を行う．ここで，$\mathcal{K}$ は鍵候補の集合，KNLL_k は各鍵候補の鍵の尤度を格納するための変数，invcoef は二項係数の逆数を格納するための

Algorithm 5 鍵の尤度による秘密鍵の推定方法

Require: Data for profiling: $\mathcal{D}_P$, Data for attack: $\mathcal{D}_A$, Set of key candidates: $\mathcal{K}$, Bit-length of intermediate value: r
Ensure: Estimated secret key: k'
1: $\theta \leftarrow \mathrm{Train}(\mathcal{D}_P)$
2: **for** $k \in \mathcal{K}$ **do**
3: $\quad \mathrm{KNLL}_k \leftarrow 0$
4: **end for**
5: $\mathrm{invcoef} = [1/\binom{r}{0}, 1/\binom{r}{1}, \ldots, 1/\binom{r}{r}]$
6: **for** $k \in \mathcal{K}$ **do**
7: $\quad$ **for** $(m, \boldsymbol{x}) \in \mathcal{D}_A$ **do**
8: $\quad\quad z \leftarrow \psi(k, m)$
9: $\quad\quad \mathrm{KNLL}_k \leftarrow \mathrm{KNLL}_k - (\log(p(z \mid \boldsymbol{x}; \theta)) + \log(\mathrm{invcoef}[z]))$
10: $\quad$ **end for**
11: $\quad \mathrm{KNLL}_k \leftarrow \mathrm{KNLL}_k / |\mathcal{D}_A|$
12: **end for**
13: $k' \leftarrow \arg\min_k \mathrm{KNLL}_k$
14: **return** k'

配列である．8–9 行目では，各鍵候補の鍵の尤度を学習済みモデルを使用して計算する．このとき，尤度の計算は二項係数による割り算を含むこと以外は，従来の NLL の計算と全く同一であることに注意されたい．最後に，13–14 行目で，最も NLL の小さかった鍵候補 k' を正解鍵として返す．

4.6　データ拡張との関係

本節では，SMOTE に基づくデータ拡張手法について述べたあと，データ拡張と提案する鍵の尤度に基づく鍵推定の関連性について述べる．

TCHES 2019 で，HW/HD による不均衡データの問題を解決するために，SMOTE と呼ばれるデータ拡張手法を用いてラベルの出現確率を均等する手法が有効となりうることが実験的に示された [76]．SMOTE は，少数データを増やすための最も有名なアルゴリズムの一つであり，これを拡張した多様なデータ拡張アルゴリズムが報告されている [77]．SMOTE では，まず少数派クラスラベルデータをランダムに一つ選択し，そのデータ近傍の同ラベルのサンプルを事前に定められたパラメータ a 数分だけ選択する．そして，この a 個の近傍点からランダムに選択した点と，最初に選択したサンプル点の間の点を，新たな点としてデータに追加する．

SMOTE のようなオーバーサンプリング手法が DL-SCA において有効である理由とし

て，上述した鍵値の NLL の計算に対応していることが理由として挙げられる．これを以下に示すベイズの定理を用いて説明する．

$$q(z \mid \boldsymbol{x}) = \frac{q(\boldsymbol{x} \mid z)}{q(\boldsymbol{x})} q(z). \tag{4.12}$$

上述の通り，もし波形が HW/HD に関する十分な情報を持っていない場合には，出力分布が二項分布に近づくことで鍵の推定が困難となる．これは，式 (4.12) において $q(\boldsymbol{x} \mid z)/q(\boldsymbol{x}) \approx 1$ であることを意味する．このとき，右辺の $q(z)$ が推定確率に大きな影響を及ぼす．一方で，HW/HD から鍵値の確率を導出するとは，式 (4.12) の両辺をラベルの出現確率で除していると見ることができる．これにより提案手法では，波形に十分な情報が含まれていない場合でも秘密情報の推定が可能となる．SMOTE はラベルの生起確率を一様分布にすることで，式 (4.12) の $q(z)$ を擬似的に一様にすることから，鍵の尤度を使用した場合と同様にラベルの不均衡による悪影響を緩和する効果が期待できる．

一方で，提案手法と SMOTE では，その効果に次の違いがある．すでに述べたとおり SMOTE は実際の漏洩電磁波や消費電力などのサイドチャネル情報を考慮・表現した拡張法ではないため，プロファイリング攻撃に必ずしも有効であるとは考えにくい．例えば，サイドチャネル攻撃ではジッターなどの影響で波形が時間方向にシフトすることが知られているが，SMOTE ではそのような影響を考慮することはできない．また，サイドチャネル攻撃では，波形にノイズを付与するようなデータ拡張が有効であることが知られているが [38]，SMOTE ではそのようなデータを追加することも難しい．このような不自然なサンプルの追加は訓練データとテストデータの間の確率分布を遠ざけてしまう可能性がある [78, 79]．

4.7 実験評価

本節では，実験を通して，本章の主張と提案手法の有効性を確認する．まず最初に実験条件について述べ，次に実験結果を示す．

4.7.1 実験条件

本稿の実験で使用したデータセットを以下に示す．

■AES_RD dataset　8-bit AVR マイクロコントローラに実装された 128-bit AES のソフトウェア実装における消費電力波形のデータセットである．同実装にはランダム遅延による対策が施されている [80]．本実験では，同データセットに含まれた 5 万波形のうち半分の 2 万 5 千波形を学習用データとし，残りの半分をテスト用データとした．

■ASCAD dataset　ATmega8515 に実装されたブーリアンマスキングによる対策が施された AES ソフトウェア実装から取得された消費電力波形のデータセットである [9]．多くの先行研究で，同データセットが標準的に使用されている [73, 81]．本実験では，同データセットの推奨に従い固定鍵サブセットの 5 万波形を学習用，残りの 1 万波形をテスト用データとした．ASCAD データセットは近年の DL-SCA に関する文献のほとんどで用いられている公開データセットである．

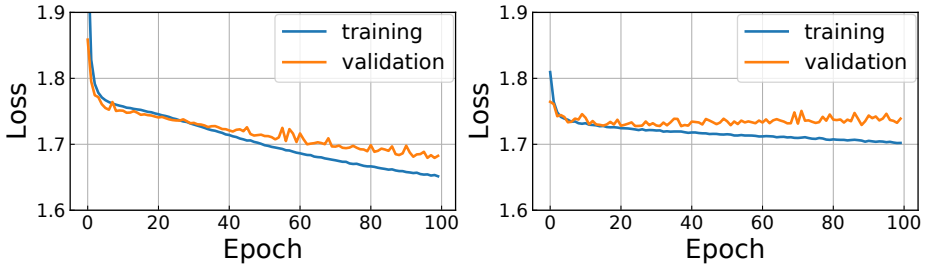
さらに上記のそれぞれのデータセットについて，学習用データのうち 10% を検証用，残りの 90% を実際の訓練用データとした．本実験では Wouters らによって提案された CNN[81] を用いてプロファイリング攻撃を実施し，その攻撃成功確率（SR: Success Rate）を評価した．Wouters らは使用するデータセットに応じてモデルの構造を変更することを推奨しているため，本稿でもそれに従った．学習時のオプティマイザーには Adam [82] を使用し，学習率は推奨値である 0.0001 とした．使用したラベルはすべてのデータセットで，最初のラウンドの S-box の出力の HW とした．バッチサイズは 50，エポック数は 100 をすべてのデータセットに用いた．使用したラベルは，平文と秘密鍵から最初のラウンドの AES S-box の出力の HW を使用した．SR の評価は，1,000 回攻撃を行った際の正解鍵の平均のランクを用いて行った．各評価では，一様ランダムにテストデータセットから 500 波形を取り出して，ランクの計算を行った．本実験では，CPU として Intel Xeon W-2145 CPU，GPU が GeForce GTX 2080 Ti で，128GB メモリを搭載したワークステーションで実験を行った．使用したライブラリは Tensorflow 2.4.1 である．

4.7.2　尤度の比較

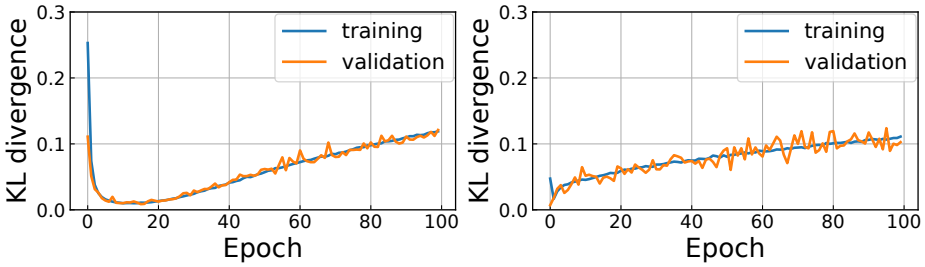
最初に，提案手法の有効性を HW と鍵の尤度の比較実験通して確認する．モデルの学習には従来の HW の尤度を使用する．ここでは，ASCAD の学習のみ過小適合を防ぐために学習率を 0.005 に設定した．

図 4.2 にモデルの学習中の NLL と KL ダイバージェンスを示す．ここで，横軸がエポック数，縦軸は図 4.2(a) がロス，図 4.2(b) が KL ダイバージェンスを表す．図 4.3 は学習時の各エポックのモデルの SR を示す．図 4.3(a) が HW の尤度であり，(b) が提案する鍵の尤度を使用した結果を示す．図 4.4 は，500 波形使用して攻撃を行った際の，各エポックの SR を示す．

図 4.2(b) から，AES_RD と ASCAD データセットともに，学習初期は KL ダイバージェンスが非常に小さく，学習が進むにつれて徐々に上昇することが確認できる．これは，モデルが学習初期に二項分布に適合し，その後正解ラベルの確率を上昇させるように学習が進むことを示している．また，図 4.2(a) から，ASCAD ではモデルの検証ロス



(a) AES_RD データセット（左）と ASCAD データセット（右）の学習曲線



(b) AES_RD データセット（左）と ASCAD データセット（右）の KL ダイバージェンス

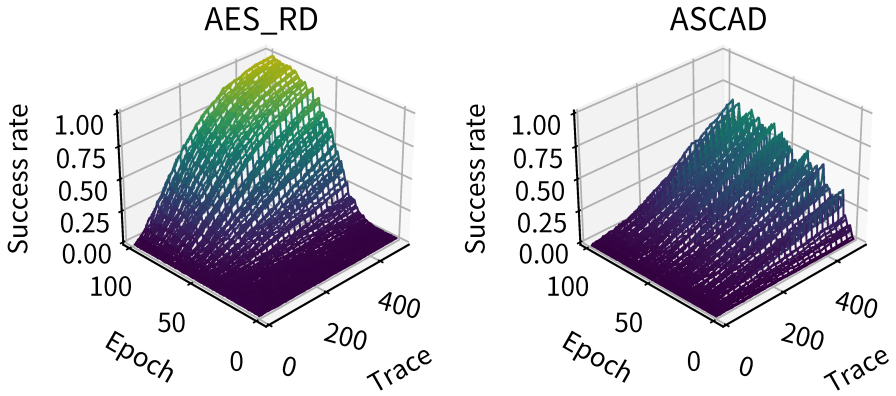
図 4.2: NLL ロスと KL ダイバージェンス

が 20 エポックで最小値を取り，それから徐々に上昇していることがわかる．検証データと学習データの分布の差が十分に小さいのであれば，20 エポック以降は過剰適合していることを意味する．しかし，図 4.3(a) に示すとおり，HW の NLL を使用して ASCAD データセットへ攻撃を行った場合は，過剰適合する 20 エポック後も攻撃性能が向上している．これは DL-SCA において過剰適合が必ずしも悪影響を持たないことを示唆している．すでに 4.4 節で述べたとおり，KL ダイバージェンスを大きくするように学習を促すことで，データのクラスの不均衡によって生じる悪影響を軽減することができる．過剰適合は，その意味で攻撃に有利に働いたと考えられる．

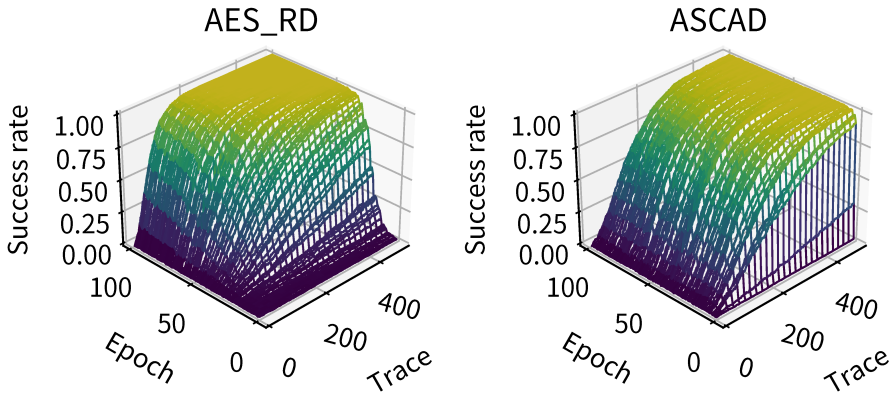
次に，図 4.3 と 4.4 から，提案する尤度のほうが HW の尤度よりも圧倒的に SR が向上していることがわかる．加えて，KL ダイバージェンスが小さいときほど HW と鍵のどちらの尤度を使用したかによって攻撃結果が大きく異なっている．これは，たとえ学習済みモデルの出力が二項分布に近いような状況であっても，提案する鍵の NLL を使用することで，その悪影響を劇的に小さくできることを示している．

4.7.3　データ拡張手法との比較

次に，KL ダイバージェンスの観点から，データ拡張の影響について解析する．本実験では，従来手法と同様に SMOTE を使用し，実装にあたってオープンソースライブラリ



(a) 従来の HW の尤度による AES_RD データセット（左）と ASCAD データセット（右）の SR



(b) 鍵の尤度による AES_RD データセット（左）と ASCAD データセット（右）の SR

図 4.3: 従来手法と提案手法による SR

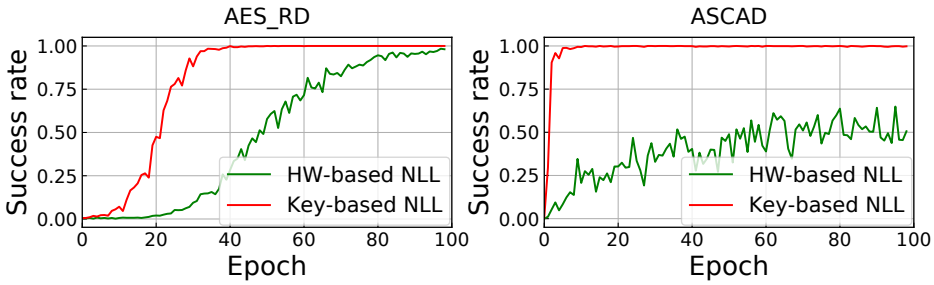
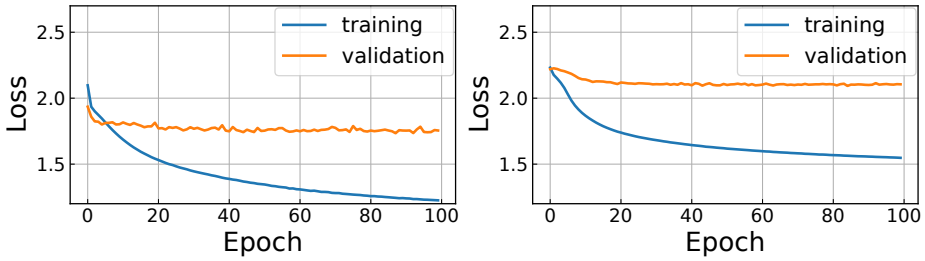
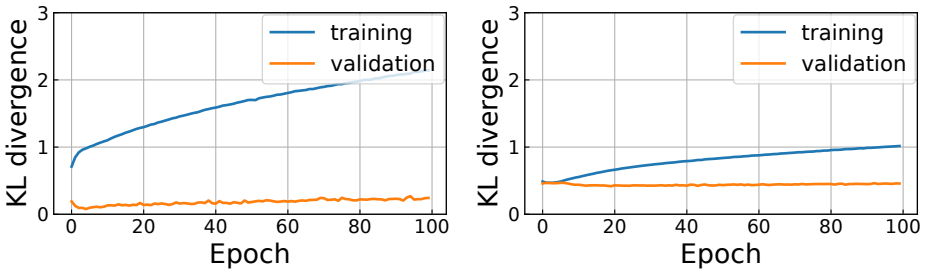


図 4.4: 攻撃時に 500 波形使用したときの攻撃成功確率

“imbalanced-learn” を使用した [83]．使用する近傍点の数は 5 とし，少数派クラスのサンプル数をすべてのクラスラベルの数が同数になるまで，データ拡張を行った．データ拡張を行うのは学習データのみとし，検証データにはデータ拡張を適用しなかった．したがって，検証ロスとその KL ダイバージェンスはテストデータにモデルがどれくらい適合しているかを表す．推論時には，従来の HW の尤度を使用した．



(a) データ拡張した場合のときの AES_RD データセット（左）と ASCAD データセット（右）の学習曲線



(b) データ拡張した場合のときの AES_RD データセット（左）と ASCAD データセット（右）の KL ダイバージェンス

図 4.5: SMOTE を使用したときの NLL ロスと KL ダイバージェンス

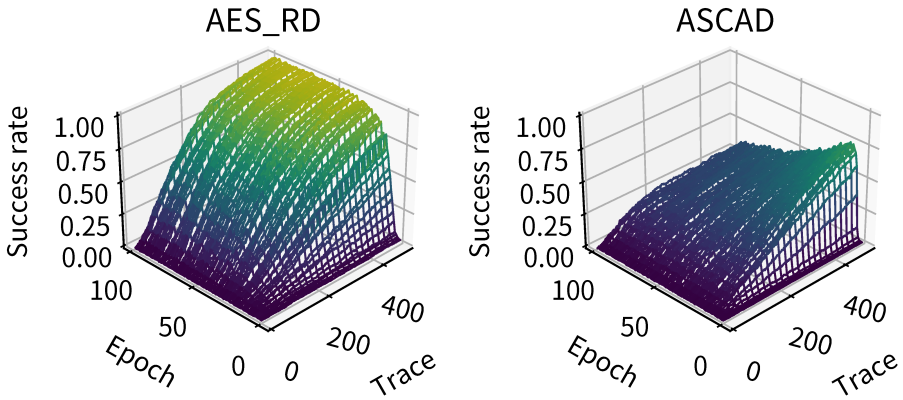
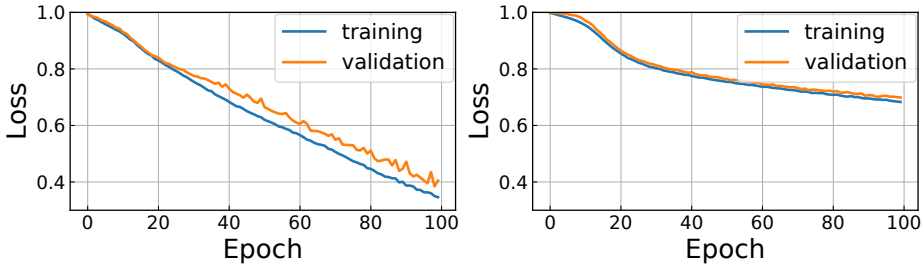
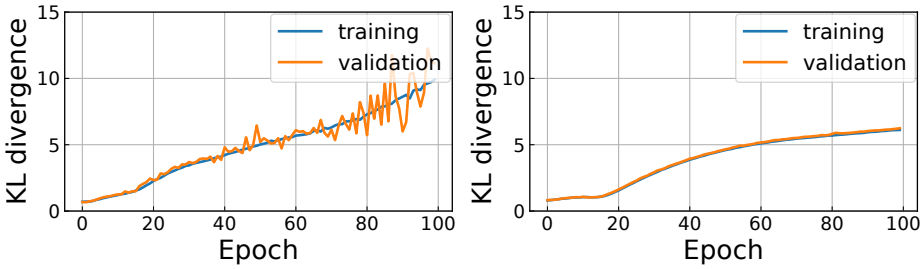


図 4.6: SMOTE を使用したときの攻撃成功確率

図 4.5 に SMOTE によりデータ拡張したデータセットで学習したモデルの，NLL ロスと KL ダイバージェンスの値を示す．結果から，データ拡張後のデータセットで学習させたモデルのほうが，元のデータセットで学習させた場合と比べて KL ダイバージェンスが上昇していることがわかる．これは SMOTE で少数派クラスの数を増加させたことで，ラベルの正規確率が一様になったためだと考えられる．加えて，学習データとテストデータの間の分布間距離が，データ拡張により大きくなったために，検証ロスも増加してしまっている．



(a) AES_RD データセット（左）と ASCAD データセット（右）の学習時の CER ロスの値



(b) CER ロスを使用したときの，AES_RD データセット（左）と ASCAD データセット（右）の学習時の KL ダイバージェンスの値

図 4.7: CER ロスと KL ダイバージェンス

図 4.6 に学習済みモデルを使用して鍵の推定を行った際の SR を示す．図 4.3(a) との比較から，SMOTE は HW の NLL において SR を向上させることが読み取れる．一方，提案する鍵の尤度の結果と比較すると，SMOTE は SR を悪化させてしまっていることがわかる．これは，SMOTE がクラス不均衡を改善させる一方で，人工的に追加したサンプルによってデータセットの品質が劣化してしまったことが原因だと考えられる．SMOTE と異なり，提案する鍵の尤度はサンプルの品質を劣化させずにデータの不均衡による影響のみを改善することから，よりよい結果を与える．

4.7.4　ロス関数の比較

CER ロスを使用して学習したモデルを利用して鍵推定を行った．CER ロスの計算は，文献 [73] に従った．CER の分母の鍵候補によるクロスエントロピーの平均の計算に使用したシャッフルの回数は 100 とした．

図 4.7 に CER ロスを用いて学習を行ったときの，学習中のモデルのロスと KL ダイバージェンスの値を示す．学習が進むに連れて，KL ダイバージェンスの値は急速に増加しており，それに対して CER ロスの値は急激に減少していることがわかる．これは，4.4 節の解析結果と一貫性のある結果である．図 4.8 に CER ロスで学習したモデルを用いて

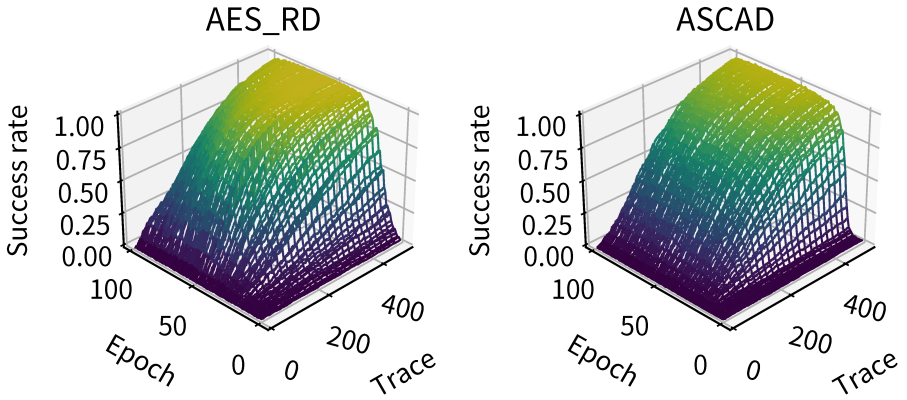


図 4.8: AES_RD データセット（左）と ASCAD データセット（右）で CER ロスを使用したときの攻撃成功確率

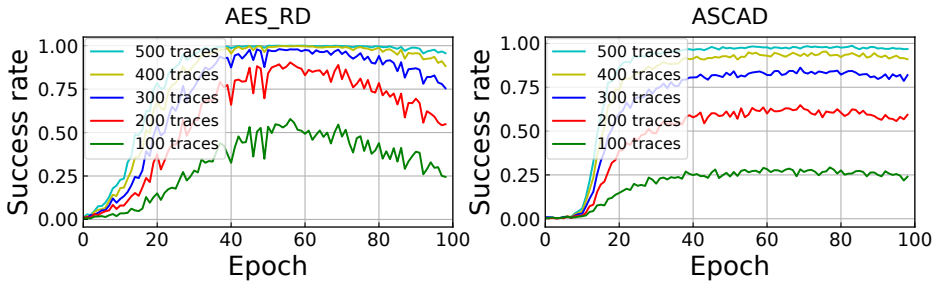


図 4.9: CER ロスを使用したモデルにより攻撃を行ったときの SR

SR を計算した結果を示す．結果から，NLL ロスを使用した場合と比べて，SR が明らかに良くなっていることがわかる．これは，KL ダイバージェンスの増加が，モデル出力と二項分布の距離を離すことで，不均衡データの悪影響を緩和したことが理由であると考えられる．

一方で，図 4.7(a) と 4.8 から，検証ロスが単調に減少しているにもかかわらず，SR が必ずしも上昇していないことがわかる．図 4.9 に，波形数が 100, 200, 300, 400, 500 のときの SR を示す．明らかに 100 エポックで AES_RD データセットでは，SR が減少している．これは，文献 [73] の CER ロスが DL-SCA のための有効な評価指標であるという記述と整合しない．4.4 節で述べたとおり，この理由は正解鍵とそれ以外の鍵候補の中間値が独立であるという誤った仮定によるものであると考えられる [76]．

最後に，提案手法を使用した場合と，CER ロスで学習を行った場合の比較を行う．図 4.3(b) から，提案手法（鍵の尤度）を使用して鍵の推定を行うほうが，CER ロスで学習を行い HW の尤度を使用して鍵の推定を行うよりも，SR が良いことがわかる．ここで，KL ダイバージェンスが単調に増加することが SR が向上することを必ずしも意味しないことに注意されたい．KL ダイバージェンスはあくまで不均衡データによる悪影響を図る

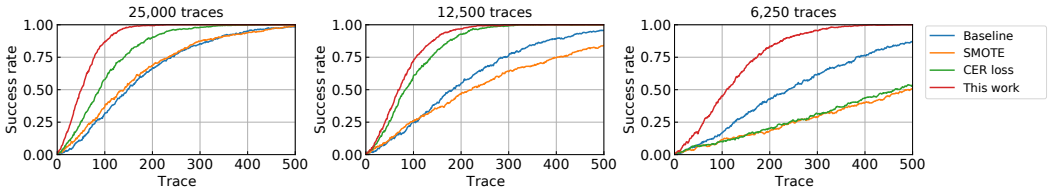


図 4.10: AES_RD データセットにおいて，学習データを 1/4 まで減らしたときの SR

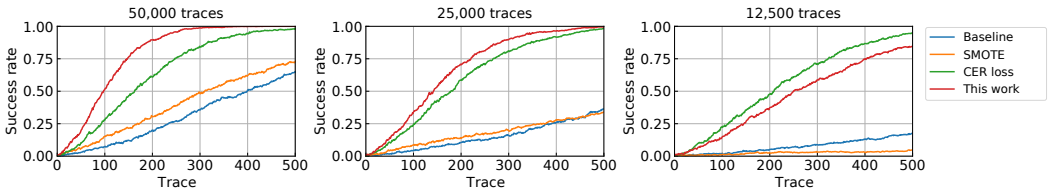


図 4.11: ASCAD データセットにおいて，学習データを 1/4 まで減らしたときの SR

ための指標であり，DL-SCA の攻撃性能を表す指標ではないためである．

4.7.5　学習データを減らしたときの影響

本節では，学習データの波形数を減らした場合の影響を調査した．AES_RD と ASCAD のそれぞれのデータセットに対して，学習データ（波形数）を半分，四分の一にして学習を行い，そのときの攻撃成功確率を調査した．学習時に使用したハイパーパラメータ等は，前節までと同じものを用いた．図 4.10 と 4.11 に，結果を示す．それぞれの図の左がもとのデータの場合，真ん中が学習データが半分の場合，右が学習データを 1/4 にした場合である．学習データを減らした場合，1 エポックあたりのイテレーション回数（パラメータ更新回数）が減少することから，本実験では学習データを減らした量に応じてエポック数を増加させている．例えば，学習データが半分の場合は，エポック数を二倍にしている．図 4.10 と 4.11 には，すべてのエポックの中で最も攻撃成功確率が高いものを示している．図中の Baseline は，学習と攻撃のどちらにも従来の HW ベースの NLL を使用した場合である．

結果から，すべての手法において，波形数を減らしていくにつれて性能が悪くなっていることがわかる．また，データ拡張手法である SMOTE は，学習データが少ないときに必ずしも良い結果にはならないことがわかる．この原因ととして，SMOTE は与えられた学習データに対する線形補間でデータを増やす方法なので，学習データが少ないときは，拡張されるデータのバリエーションも減少してしまい，結果的に過学習を促進してしまっている可能性が考えられる．CER loss を使用した場合は，総じて Baseline よりもよい性

能が得られていることがわかる．一方で，AES_RD データセットの 6,250 波形の結果が示すとおり，波形数が少なすぎる場合は性能が大幅に落ちる場合があることがわかる．最後に提案手法は，ASCAD データセットの 12,500 波形の場合を除いて最も良い性能を示しており，全体的に攻撃成功確率が高い．また波形数を減らした場合に性能の下げ幅がほぼ一定であり，その意味でロバスト性がある．以上から，本実験では全体的に提案手法が安定的に最も性能がよく，次に CER loss を使用した場合が優れた性能を示していることがわかる．どちらも基本的に Baseline よりも性能がよく，不均衡データによる悪影響を除去できている．一方で，SMOTE は性能の上げ幅が小さく，使用することによる利点は小さい．

4.8 結び

本章では，DL-SCA による安全性評価を実施する上で障害となっていた不均衡データ問題について解析を行い，その解決策を提案した．まず，従来の HW/HD の尤度による推定が同問題から受ける影響を KL ダイバージェンスで説明できることを示した上で，鍵の尤度による推定を提案した．また，従来の SMOTE による解決策が同問題に効果的に働く場合があることをベイズの定理から説明した．その上で，提案する鍵の尤度による推定が同問題に対して従来手法と比べてより有効であることを実験的に示した．

第 5 章

公開鍵暗号モジュールの物理攻撃に対する安全性評価手法

5.1 はじめに

本章では，公開鍵暗号（PKE: Public Key Encryption）モジュールとして，耐量子計算機暗号（PQC: Post Quantum Cryptography）を対象とし，その DL-SCA について述べる．対象とする PQC は，現在行われている NIST の PQC コンペティションの第 3 ラウンドの KEM（Key Encapsulation Mechanism）スキームである．本章では，まずすべての PQC KEM スキームで共通して使用されている FO 変換について述べ，FO 変換を対象としたサイドチャネル攻撃を説明する．次に，提案攻撃とその中核をなす平文判定オラクルを導入する．また，各 KEM スキームについて提案攻撃の方法について説明する．最後に提案手法を用いた実験評価を行う．

5.2 関連研究

5.2.1 FO 変換に基づく IND–CCA2 安全 KEM

KEM は，秘密鍵をカプセル化するための公開鍵暗号プリミティブである．KEM は，鍵生成 KeyGen，カプセル化 Encaps，脱カプセル化 Decaps の 3 つの多項式時間アルゴリズムの組として定義される．多くの CCA2 安全な KEM は，CPA 安全な KEM に FO 変換を用いることで構成される [84, 85, 86, 87]．現在 NIST によって行われている PQC のコンペティション [88] における，KEM の候補のほとんどが同様の方法で，CCA2 安全を実現している．

Algorithm 6 に，標準的な FO 変換に基づく KEM の擬似コードを示す．アルゴリズム中の，PKE は CPA 安全な確率的 PKE のことである．PKE は，鍵生成アルゴリズム Gen，暗号化アルゴリズム Enc，復号化アルゴリズム Dec から構成される．ここで例

Algorithm 6 FO 変換に基づく CCA2 安全な KEM (KeyGen, Encaps, Decaps) のアルゴリズム

KeyGen

Require: 1^λ
Ensure: $\mathsf{sk}, \mathsf{pk}, s$
1: **function** KeyGen(1^λ)
2: $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{PKE.Gen}(1^\lambda)$;
3: $s \leftarrow_\$ \mathcal{M}$;
4: **return** $(\mathsf{sk}, \mathsf{pk}, s)$;
5: **end function**

Encaps

Require: pk
Ensure: c, k
1: **function** Encaps(pk)
2: $m \leftarrow_\$ \mathcal{M}$;
3: $r \leftarrow \mathsf{G}(m[, \mathsf{pk}])$;
4: $c \leftarrow \mathsf{PKE.Enc}(\mathsf{pk}, m; r)$;
5: $k \leftarrow \mathsf{H}(m, c)$;
6: **return** (c, k);
7: **end function**

Decaps

Require: $c, \mathsf{sk}, \mathsf{pk}, s$
Ensure: k
1: **function** Decaps($c, \mathsf{sk}, \mathsf{pk}, s$)
2: $m' \leftarrow \mathsf{PKE.Dec}(\mathsf{sk}, c)$;
3: $r' \leftarrow \mathsf{G}(m'[, \mathsf{pk}])$;
4: $c' \leftarrow \mathsf{PKE.Enc}(\mathsf{pk}, m'; r')$;
5: **if** $c = c'$ **then**
6: **return** $\mathsf{H}(m', c)$;
7: **else**
8: **return** $\mathsf{H}_{\mathsf{prf}}(s, c)$;
9: **end if**
10: **end function**

示する KEM の擬似コードでは，不正な暗号文が与えられた場合に，復号失敗を意味する記号 $\perp$ ではなく，疑似乱数を返す．セキュリティパラメータ 1^λ が与えられたとき，KEM.KeyGen は，PKE の鍵生成アルゴリズム PKE.Gen を用いて，鍵ペア $(\mathsf{sk}, \mathsf{pk})$ を返す．次に，メッセージ空間 $\mathcal{M}$ から，PKE のランダムな平分として s を生成する．最後に，鍵と平文のペア $(\mathsf{sk}, \mathsf{pk}, s)$ を返す．

KEM.Encaps は，まずメッセージ空間 $\mathcal{M}$ からメッセージ m を生成し，m もしくは m と pk のペアに対してランダムオラクル G を評価する．このランダムオラクルは実際には，擬似乱数関数（PRF: Pseudo Random Function）か擬似乱数生成器（PRG: Pseudo Random Generator）を用いて実現される．4 行目の，PKE の復号アルゴリズム PKE.Enc は，公開鍵 pk と平文 m，乱数 r に対して実行される．次に，平文 m と暗号文 c に対してランダムオラクル H を評価し，共有鍵 k を生成する．最後に，KEM.Encaps は，共有鍵 k に対応する暗号文 c を返す．

KEM.Decaps は，まず秘密鍵 sk を用いて暗号文 c の復号を行い，平文 m' を入手する．次に，KEM.Encaps と同様に，KEM.Decaps は，$\mathsf{G}(m')$ もしくは $\mathsf{G}(m', \mathsf{pk})$ を計算して r' を生成し，暗号化処理 $\mathsf{PKE.Enc}(\mathsf{pk}, m'; r')$ を行う．この処理は，再暗号化と呼ばれる．5 行目では，KEM.Decaps は再暗号化結果 c' と暗号文 c の等価性チェックを行う．もし，$c = c'$ であれば，同アルゴリズムは暗号文が有効であると判断し，秘密情報 $k = \mathsf{H}(m', c)$ を返す．両者が等しくない場合には，暗号文が不正であると判断し，失敗した結果として疑似乱数 $\mathsf{H}_{\mathsf{prf}}(s, c)$ を返す．ここで，$\mathsf{H}_{\mathsf{prf}}$ は，ランダムオラクルである．KEM のスキームによっては，$\mathsf{H}_{\mathsf{prf}}$ に，H を用いることがある．この疑似乱数を返す処理によって，不正な暗号文による能動的な攻撃に対して，いかなる情報も漏らさないことを保証できる．

現代の多くの KEM スキームでは，ランダムオラクル G，H，$\mathsf{H}_{\mathsf{prf}}$ は，SHAKE か SHA3

により実現される．また，異なる種類の CPA 安全 PKE や，セキュリティモデル，もしくはよりタイトなセキュリティバウンドなどのために，様々な FO 変換の亜種が存在する [85, 86, 87]．しかしながら，原理的には，そのすべての亜種が，Algorithm 6 に示した手順に従っている．

5.2.2　FO 変換に対するサイドチャネル攻撃

■タイミング解析　Guo らは，CRYPTO 2020 において，FO 変換を対象としたサイドチャネル攻撃を提案した [89]．同攻撃では，サイドチャネル情報から得られるタイミング情報を用いて，格子もしくは符号ベース KEM に対する平文判定オラクルを実現する．KEM.Decaps の復号処理 PKE.Dec ではなく，暗号文と再暗号化結果の等価性判定に対してタイミング攻撃を行うため，使用する公開鍵暗号が定数時間実装であっても，同攻撃は適用可能である．

このタイミング攻撃は KEM に対する選択暗号文攻撃であり，格子や符号ベース PKE に対する適応的攻撃を，平文判定オラクルを使用して実現する．平文 m に対応する参照用暗号文と呼ばれる有効な暗号文を c とする．不正な暗号文 c' に対して，平文判定オラクルは c' の復号結果が，m に等しいかどうかを返す．この平文判定オラクル はサイドチャネル情報から得られるタイミング情報によって実現される．攻撃者は，適応的攻撃に従って決定される変数 δ を用いて，不正な暗号文 $c' = c + \delta$ を生成する．格子やコードベース PKE では，δ が十分に小さいとき，c' の復号結果は m と等しくなる．これは，再暗号化結果が暗号文 c と等しいことを意味する．一方で，c' の復号結果と m が等しくならないほどに，δ が大きい場合は，公開鍵暗号の復号結果はランダムな平文 $\hat{m}$ を返す．すると当然，その再暗号化結果 $\hat{c}$ は，暗号文 c とは全く異なるものとなる．PKE.Decaps における等価性判定では，暗号文 $c + \delta$ と，再暗号化結果の比較を行う．格子やコードベース公開鍵暗号では，暗号文が巨大なベクトルとして表現される．したがって，もし 2 つの暗号文が同じであれば，その等価性の評価には多くの時間を要することになる．逆に，暗号文が等しくない場合には，最初の数バイトの等価判定だけで済むため，比較処理は即座に終了することが期待される．以上から，比較処理にかかる時間の長さにより，復号結果と平文 m の等価性を知ることができる．これが，平文判定オラクルの実現方法である．完全な秘密鍵の復元には，異なる δ に対する平文判定オラクルの複数回の評価が必要となる．

文献 [89] で，Guo らは FrodoKEM への同攻撃の適用例を示した．タイミング情報に基づく平文判定オラクルの実現にあたっては，サイドチャネル情報の信号ノイズ比（SNR）が問題となり得るものの，完全な鍵の復元が十分に可能であることが示されている．同攻撃の公開により，安全な c と c' の比較のために，定数時間条件付きムーブ処理 `cmov` が，多くの PQC の実装において用いられている．

■電力・電磁波解析 文献 [90] で，Ravi らは格子ベース KEM に対するサイドチャネル攻撃を提案した．この攻撃は，サイドチャネル情報を用いた選択暗号文攻撃である．攻撃に使用される暗号文は，復号結果となる平文が 0 か 1 となるように生成された暗号文である．ここで，0 と 1 のどちらに復号されるかは，部分鍵に依存して決まる．FO 変換により，攻撃者は平文を直接確認することはできないが，平文が 0 と 1 のどちらに復号されたかは，再暗号化処理中のサイドチャネル漏洩を用いて知ることができる．攻撃者は，異なる部分鍵に対応する不正な暗号文を，繰り返し問い合わせることで，格子ベース KEM の完全な鍵推定を行う．文献 [90] で，Ravi らは Kyber，Saber，FrodoKEM，Round5，NewHope，LAC の 6 つの格子ベース KEM に，同手法が適用可能なことを示した．Ravi らは，十分実現可能な完全鍵回復を実現するために，テンプレートとウェルチの t 検定を組み合わせたサイドチャネル識別器を提案した．同手法で提案された識別器は，攻撃対象の詳細な実装に関する知識を必要としない．

近年，Bhasin らは，格子ベース KEM の暗号文の等価性判定のためのマスク付き多項式比較スキームのサイドチャネル攻撃脆弱性を報告し，その Kyber への適用例を示した．同攻撃は，Guo らによるタイミング攻撃に基づいている．すなわち，$c = c'$ の多項式比較処理におけるサイドチャネル情報に対して，t 検定のようなテストベクトル漏洩評価を適用することで，平文判定オラクルを実現する．ただし，同攻撃はたかだか 3 つの格子ベース KEM でしか有効性を確認しておらず，加えてサイドチャネル攻撃対策された実装への適用可能性も不明である．よって，攻撃の一般性や現実性は明らかではない．

Bhasin らの提案以降も，格子ベース KEM に対する，サイドチャネル攻撃を利用した様々な選択暗号文攻撃が提案されている [91, 92, 93, 94]．Ngo らは，Saber のマスキング対策済み実装に対する DL-SCA を提案している．同攻撃は，使用されている PKE の実装に合わせて，よりサイドチャネル情報漏えいの多い平分を選択することで，非常に少ないオラクルアクセス回数を実現している．しかし，同攻撃は実装に強く依存している．実際，Ngo らの攻撃は選択暗号文攻撃であるものの，FO 変換ではなく KEM で使用されている公開鍵暗号の特定の部分に着目している．そのため，同攻撃は，FO 変換を対象しているものに比べて，適用可能範囲が非常に限られている．

上記の他にも，符号や同種写像ベース KEM へのサイドチャネル攻撃も提案されている [95, 96, 97, 98]．しかし，これらの攻撃は FO 変換ではなく，使用されている PKE に着目している．符号や同種写像ベース KEM の FO 変換部に対するサイドチャネル攻撃は知られていない．

5.3　提案手法

5.3.1　平文判定オラクル

本節では，提案攻撃のおいて本質的な役割を果たす平文判定オラクルを導入する．平文判定オラクルは，格子や符号，同種写像ベースなどの広域の PKE に対する適応攻撃において用いられる，一般的なオラクルの一つである [99, 89]．平文判定オラクルによる鍵回復攻撃は，KR-PCA（Key-Recovery Plaintext-Checking Attack）と呼ばれる．本章では，適応的攻撃を適応的選択暗号文攻撃の意味で用いる．

与えられた KEM に対して，参照用暗号文と呼ばれる有効な暗号文を c とする．また，参照用平文と呼ばれる対応する平文を m とする．ここで，m は KEM.Decaps の出力ではなく，公開鍵暗号の復号結果を示すことに注意されたい．攻撃者は，カプセル化処理を行うことで，任意の参照用平分に対応する参照用暗号文を手に入れられる．攻撃者は，暗号文 c に改変を加えて，不正な暗号文 c' を生成し，復号オラクルに問い合わせる．暗号文 c' に対応する平文を m' とする．適応的攻撃では，m' が参照用平文 m か，それ以外の平文 $\hat{m}$ に等しいことを利用する．形式的には，平文判定オラクル $\mathcal{O}(c', m)$ は，$m = m'$ のとき 1 を返し，それ以外のとき 0 を返す．FO 変換に基づく KEM の実装では，通常はこのようなオラクルは利用できない．なぜなら，FO 変換によって，IND–CCA2 安全性が保証されているためである．

5.3.2　提案するサイドチャネル攻撃

提案攻撃は，CPA 安全な PKE への選択暗号文攻撃を実施するために，サイドチャネル情報を利用した平文判定オラクルを利用する．提案サイドチャネル攻撃では，攻撃者は，まず PKE.Decaps における参照用暗号文 c に対する PRF 実行時のサイドチャネル情報を入手する．次に，平文判定オラクルを用いて，適応的攻撃のために改変された暗号文 c' を問い合わせる．そして，PKE.Decaps の再暗号化における PRF 実行時のサイドチャネル情報を観測する．もし c' が，参照用平文 m に復号されれば，c' に対応するサイドチャネル漏洩は c のものと非常に近いことが期待される．一方で，c' がそれ以外の平文 $\hat{m}$ へ復号された場合，これらのサイドチャネル情報は全く異なるはずである．すなわち，攻撃者は PRF のサイドチャネル情報を用いて，参照用平文が復号結果と等しいかどうかを知ることができる．提案手法は PRF の漏洩を用いるため，PKE.Dec の実装に関わらずに鍵回復攻撃を行える．

提案する攻撃は，プロファイリングフェーズと攻撃フェーズからなる．プロファイリングフェーズでは，PRF/PRG の入力が参照平文かどうかを，サイドチャネル情報から識

別する分類モデルの学習を行う．本章では，この学習済みモデルをサイドチャネル識別器と呼ぶ．続く攻撃フェーズでは，攻撃者は学習済みモデルを平文判定オラクルとして用いることで，攻撃鍵暗号への適応的攻撃を行う．学習を行うためのプロファイリングフェーズが存在するものの，攻撃対象となるデバイスから学習に必要な学習データを収集できるため，プロファイリングデバイスを別途用意する必要がないことに注意されたい．また，提案するサイドチャネル攻撃では，この分類モデルとして深層学習モデルを使用する．これにより，提案手法は，攻撃対象となるデバイスの実装に関する細かい仕様を必要としない．

5.4　耐量子 KEM への適用

5.4.1　格子ベース KEM

■攻撃の概要　格子ベース公開鍵暗号への KR-PCA のアイデアを述べる．まず復号では，デコードされる前の平文は $\mathsf{Encode}(m)+ke+e'$ で与えられる．ここで，k は秘密鍵，e と e' は誤差，Encode はエンコードアルゴリズムである．Encode は，ノイズ $ke+e'$ を取り除くためのデコードアルゴリズム Decode に対応するエンコードアルゴリズムである．$\mathsf{Encode}(m)+ke+e'$ に対応する有効な暗号文を c とする．暗号文 c は，カプセル化処理を用いて計算できる．格子ベース KEM では，ノイズ $ke+e'$ がある閾値 γ より小さいとき，暗号文が正しく復号され，平文 m へデコードされる．もし，ノイズが γ より大きいときは，誤った平文 $\hat{m}$ へデコードされる．格子ベース公開鍵暗号では，正しく復号を行うために，有効な暗号文の復号が失敗する確率が無視できるくらい小さくなるように設計する．

格子ベース KEM への KR-PCA では，攻撃者は復号オラクルへ改変した暗号文 $c'=c+\delta$ を問い合わせる．ここで，δ は攻撃者によって追加される誤差である．改変された暗号文は，$\mathsf{Encode}(m)+ke+e'+\delta$ へ復号される．ここで，$ke+e'+\delta$ は除去の対象となるノイズである．デコードされた平文を m' とする．もし，$ke+e'+\delta<\gamma$ なら，c' は正しく復号され，復号結果 m' は m と一致する．そうでない場合は，c' は誤って復号され，異なる平文 $\hat{m}$ へデコードされる．すなわち，ノイズが $ke+e'+\delta<\gamma$ を満たせば，$\mathcal{O}(c',m)=1$ であり，そうでなければ $\mathcal{O}(c',m)=0$ である．よって，攻撃者は平文判定オラクルへの問い合わせを通して，$ke+e'+\delta=\gamma$ となるように誤差を適応的に決定できる．これにより，攻撃者は e と e'，δ，γ の値をオラクルを通して知ることができるため，それらから得られる線形方程式を解くことで秘密情報 k を入手できる．さらに，各暗号スキームに合わせて，適切な暗号文を問い合わせることで，オラクルアクセスの回数を減らすことができる [100]．以降では，各暗号スキームについて，問い合わせ方法を説明する．

■FrodoKEM　ここでは，文献 [89] に従って，FrodoKEM への KR-PCA について述べる．Kyber と Saber にも同様の方法で攻撃が可能なため，これらへの攻撃方法の詳細は省略する．同様に，NTRU と NTRU Prime についても同様に攻撃できる．

秘密鍵の行列を $\mathbf{S}$ とし，$\mathbf{S}'$ と $\mathbf{E}$，$\mathbf{E}'$，$\mathbf{E}''$ を誤差行列とする．暗号文行列 $\mathbf{B}'$ と $\mathbf{C}$ のペアに対応する暗号文 (c_0, c_1) が復号オラクルの入力のとき，オラクルは平文行列

$$\begin{aligned}\mathbf{M} &= \mathbf{C} - \mathbf{B}'\mathbf{S} \\ &= \mathsf{Frodo.Encode}(m) + \mathbf{E}\mathbf{S}' - \mathbf{E}'\mathbf{S} + \mathbf{E}''\end{aligned}$$

を返す．ここで，$\mathsf{Frodo.Encode}(m)$ は平文のエンコード結果か，初期シードを表す．$\mathsf{Frodo.Encode}(m)$ に対応するデコード $\mathsf{Frode.Decode}(\mathbf{M})$ は，ノイズ $\mathbf{E}\mathbf{S}' - \mathbf{E}'\mathbf{S} + \mathbf{E}''$ を除去することで，平文 m を取り出す．

KR-PCA では，$\mathbf{C} + \Delta$ に対応する暗号文 c_0，c_1' から構成される改変済み暗号文を生成する．ここで，Δ は攻撃者によって追加された誤差行列である．暗号文 (c_0, c_1') を問い合わせたとき，復号オラクルは

$$\mathbf{M}' = \mathsf{Frodo.Encode}(m) + \mathbf{E}\mathbf{S}' - \mathbf{E}'\mathbf{S} + \mathbf{E}'' + \Delta$$

を返す．ノイズ要素 $\mathbf{E}\mathbf{S}' - \mathbf{E}'\mathbf{S} + \mathbf{E}'' + \Delta$ を $\mathbf{Q}$ とする．ここで，もし $\mathbf{Q}$ のすべての成分が，閾値 γ より小さいとき，$\mathbf{M}'$ は平文 m へ正しくデコードされ，そうでない場合は，異なる平文 $\hat{m}$ へ誤ってデコードされる．したがって，攻撃者は，平文判定オラクルへ (c_0, c_1') を問い合わせることで，$\Gamma = Q$ となるような Δ を見つけられる．ここで，Γ は，すべての成分が γ であるような定数行列である．これにより，秘密鍵 $\mathbf{S}$ 以外の $\mathbf{Q}$ のすべての成分が入手できるため，攻撃者は線形方程式 $\Gamma = \mathbf{Q}$ を解くことで秘密鍵 $\mathbf{S}$ を復元できる．

Algorithm 7 に，FrodoKEM への KR-PCA の擬似コードを示す．攻撃者は，事前に FrodoKEM.Encaps を実施することで，参照用平文と対応する正規の参照用暗号文を用意する．次に 2 行目で，サイズ $n \times \bar{n}$ の行列 Δ を値 0 に初期化する．ここで，n と $\bar{n}$ は FrodoKEM における行列サイズを表す．3–7 行目のループで，行列 Δ の (i, j) 成分を順次決定する．6 行目で，改変された暗号文 $(c_0, c_1^{(i,j,\delta)})$ を平文判定オラクルに問い合わせる．ここで，$c_1^{(i,j,\delta)}$ は，行列 $\mathbf{C}$ の (i, j) 成分に δ が加算された行列に対応する暗号文である．もし行列 $\mathbf{Q}$ の (i, j) 成分が γ より小さいとき，対応する平文行列 $\mathbf{M}'$ は平文 m へ正しくデコードされる．そうでないとき，$\mathbf{M}'$ はそれ以外の平文へデコードされる．特に，$\mathbf{Q}$ の (i, j) 成分が γ に等しいときそのときに限り，$\mathcal{O}((c_0, c_1^{(i,j,\delta)}), m) = 0$ かつ $\mathcal{O}((c_0, c_1^{(i,j,\delta-1)}), m) = 1$ が成立する．この条件を用いることで，攻撃者は平文判定オラクルを通して $\Delta_{i,j}$ の情報を入手できる．すべての i と j について $\Delta_{i,j}$ が入手できれば，攻撃者は線形方程式 $\Gamma = \mathbf{Q}$ を解くことで，秘密行列 $\mathbf{S}$ を復元できる．

Algorithm 7 FrodoKEM への KR-PCA

Require: Reference ciphertext (c_0, c_1), reference plaintext m, and noise matrices $\mathbf{S}'$, $\mathbf{E}$, $\mathbf{E}'$, and $\mathbf{E}''$
Ensure: Secret key sk (i.e., Secret matrix $\mathbf{S}$)

function AttackOnFrodoKEM($(c_0, c_1), m, \mathbf{S}', \mathbf{E}, \mathbf{E}', \mathbf{E}''$)
  $\Delta \leftarrow$ ZeroMatrix$(n, \bar{n})$;
  for $i = 0$ to $n - 1$ **do**
    for $j = 0$ to $\bar{n} - 1$ **do**
      for $\delta \in \{0, 1, \ldots, \gamma - 1\}$ **do**
        if $\mathcal{O}((c_0, c_1^{(i,j,\delta)}), m) = 0$
          and $\mathcal{O}((c_0, c_1^{(i,j,\delta-1)}), m) = 1$ **then**
          $\Delta_{i,j} \leftarrow \delta$;
        end if
      end for
    end for
  end for
  Solve linear equation $\Gamma = \mathbf{E}\mathbf{S}' - \mathbf{E}'\mathbf{S} + \mathbf{E}'' + \Delta$ about $\mathbf{S}$;
  return $\mathbf{S}$;
end function

最も愚直な方法では，各 i と j のペアについて，$\Delta_{i,j}$ を決定するために必要なオラクルアクセス回数は，最大で γ 回となる．しかし，Guo らは二分探索を用いることで，オラクルアクセス回数を $\log_2 \gamma$ 回へ減らせることを示した．これを用いることで，Algorithm 7 による完全な鍵回復に必要なオラクルアクセス回数は，$n\bar{n} \log \gamma$ 回で良い．さらに，文献 [100] では，疎な暗号文行列を利用することで，オラクルアクセス回数を更に減らす方法を提案している．$\mathbf{D}^{(i)} = (\vec{0}, \ldots, \vec{0}, \vec{1}, \vec{0}, \ldots, \vec{0})$ のような，ある i 列目だけが 1 であるような行列を考える．攻撃者は，2 つの暗号文行列 $(\mathbf{D}^{(i)}, \mathbf{C})$ を問い合わせるとする．このとき，FrodoKEM.Decaps の復号処理で，行列 $\mathbf{M} = \mathbf{C} - \mathbf{D}^{(i)}\mathbf{S} = \mathbf{C} - \mathbf{Z}$ をえる．ここで，$\mathbf{Z}$ の最初の行は，$\mathbf{S}$ の i 行目であり，残りは 0 である．$\mathbf{C}$ を改変し，デコードされた平文が 0 かどうかを平文判定オラクルで確認する．例えば，$\mathbf{C}$ の最初の行が $q/2^{B+1}$ で満たされていて，それ以外の成分が 0 であるような場合を考える．ここで，q は Frodo.Encode の環の剰余の値であり，B はビット長である．クエリとして，$\mathbf{D}^{(1)}$ と $\mathbf{C}$ を問い合わせた場合，得られる平文行列 $\mathbf{M}$ は，最初の行が $q/2^{B+1} - \mathbf{S}_{0,i}, i = 0, 1, \ldots, \bar{n} - 1$ となる．そして，これは $\mathbf{S}_{0,i} > 0$ のときそのときに限り 0 へデコードされる．よって，攻撃者は非常に少ないオラクルアクセスで，秘密行列 $\mathbf{S}$ の係数を直接推定できる．

FrodoKEM.Decaps では，まず平文 m' を PKE.Dec で計算する．次に，m' と公開鍵を連結し，それに対して SHAKE を計算する．SHAKE の入力は，m' と公開鍵にのみ依存するため，提案するサイドチャネル攻撃を FrodoKEM.Decaps に適用できる．

5.4.2 Kyber と Saber

FrodoKEM に対する疎行列と平文判定オラクルを組み合わせた攻撃と同様の方法が Kyber と Saber にも存在するため，提案するサイドチャネル攻撃はこれらの暗号スキームにも適用できる．具体的には，Kyber は，Kyber-512 の NIST PQC 第 2 ラウンドへの攻撃方法 [101] を元に鍵回復を行う．また，Saber は，Huguenin-Dumittan と Vaudenay の LightSaber への攻撃と，Osumi らの Saber と FireSaber への攻撃と同等の手法を用いる．

5.4.3 NTRU

NTRU には NTRU-HPS と NTRU-HRSS の二種類の KEM スキームが存在する．2 つの KEM における公開鍵暗号の公開鍵を h，平文を 2 つの短い多項式のペア (r, m)，そして暗号文を $c = h \cdot r + \mathsf{Lift}(m) \in \mathbb{Z}[x]/(q, x^n - 1)$ とする．ここで，Lift は同型写像である．NTRU の暗号文空間は $\{\, c \in \mathbb{Z}[x]/(q, x^n - 1) \mid c \equiv 0 \pmod{(q, x-1)} \,\}$ であり，これは $\mathbb{Z}[x]/(q, (x^n - 1)/(x - 1))$ と同型である．

NTRU への既存の適応的選択暗号文攻撃のうち，KR-PCA に適用可能な候補が 2 つある．1 つは Hoffstein と Silverman[102] と Jaulmes と Joux[103] が提案した方法である．同攻撃では，暗号文 $c = h \cdot r + \mathsf{lift}(m)$ を $c' = c + \delta$ に改変する．そして，復号された平文 (r, m) のうちの m が期待される平文 m_{guess} と等しいかどうかを判定する．ここで，期待される平文のうちの半分 r_{guess} は，暗号文 c' を用いて $r_{\mathrm{guess}} = (c' - \mathsf{lift}(m_{\mathrm{guess}})) \cdot h^{-1}$ と求められることに注意されたい．同手法を NTRU に適用するにあたって発生する課題は，元の NTRU と平文空間が異なることである．そのため，$\delta \equiv 0 \pmod{(q, x-1)}$ をみたすように δ を設定しなければならない．この制約は解析を複雑にし，KR-PCA の適用を著しく困難にする．

もう一つの適用可能な適応的選択暗号文攻撃は，文献 [104] と [105] で提案されたものである．これらの攻撃では，まず m_{guess} を 0 に固定し，次に r' と r_{guess} を改変し，暗号文 $c = h \cdot r'$ を計算して，最後に復号された平文 (r, m) が推定値　$(r_{\mathrm{guess}}, m_{\mathrm{guess}})$ と一致するかどうかを調べる．NTRU の設計上，$h \equiv 0 \pmod{(q, x-1)}$ を満たすため，暗号文について $c \equiv 0 \pmod{(q, x-1)}$ が成立する．

NIST のコンペティションにおける第 2 と 3 ラウンドの NTRU は，FO 変換と類似の方法である SXY [86] を使用している．SXY では，暗号化処理 PKE.Enc が確率的ではなく確定的なアルゴリズムであるため，$r' \leftarrow G(m')$ の計算を含まない．さらに，NTRU は再暗号化を明示的には実行しない．しかし，NTRU は脱カプセル化の処理において暗号文の正当性の確認を行うため，提案サイドチャネル攻撃を通して，依然として攻撃可能

である．加えて，ARM の Cortex-M4 向けのオープンソースライブラリである pqm4 では，NTRU の脱カプセル化プログラムで，$k = \mathsf{H}(r, m)$ と $k' = \mathsf{H}_{\mathsf{prf}}(s, c)$ のどちらの計算も行ってから，暗黙的な再暗号化テストの結果として両者のうちの片方だけを出力する実装になっている．そのため，それぞれのハッシュの計算のサイドチャネル情報を用いて，$m_{\mathrm{guess}} = 0$ かどうかを判定することができる．

5.4.4 NTRU Prime

NTRU Prime には，sntrupr（Streamlined NTRU Prime ）と ntrulpr（NTRU LPRime ）の 2 つの KEM スキームがある．NTRU LPRime は Kyber ，Saber ，FrodoKEM と同様の構造を持つため，文献 [106] と同様の方法で，KR-PCA が可能である．この場合，復号結果となる平文は 1^{ℓ} か，任意の i について $1^{e-1} \parallel 0 \parallel 1^{\ell-i-1}$ のようなベクトルとなる．

Streamlined NTRU Prime は NTRU と同様の構造を持つ．Streamlined NTRU Prime では平文は r，暗号文は $c = \mathsf{Round}(h \cdot r)$ である．ここで，$\mathsf{Round}(x)$ は，x の各係数を $3\mathbb{Z}$ の最も近い要素に丸める関数である．しかし，NTRU への従来の KR-PCA を Streamlined NTRU Prime へ適用するには，いくつかの技術的な問題がある．ます，Streamlined NTRU Prime の PKE.Dec は，内部的には各復号された平文 r のハミング重みを確認し，もしこのテストに失敗したら平文 r を固定した平文 r_{fixed} に置き換える．Jaulmes と Joux らによって提案された NTRU への選択暗号文攻撃をうけて，Ravi らは Streamlined NTRU Prime への 2 つの鍵回復サイドチャネル攻撃を提案した．

1 つ目の攻撃は平文判定オラクルに基づいており，内部変数が 0 であるかどうかを調べることで行われる．もし内部の復号された平文が 0 か，特定の多項式であれば，ハミング重みは不正なものであり，PKE.Dec の出力は r_{fixed} となる．この攻撃に従って平文判定オラクルを実装するためには，PRF ではなく $\mathsf{PKE.Dec}(\mathsf{sk}, c)$ の計算時のサイドチャネル情報を解析する必要がある．これは，本章で提案するサイドチャネル攻撃の範囲外である．

2 つ目の攻撃は，復号失敗オラクルに基づいており，復号により得られる平文が r と同じかどうかを調べることで行われる．もし一致する場合，r のハミング重みは有効である．一方，復号に失敗した場合，復号された平文のハミング重みは有効ではなく，固定された平文 r_{fixed} に置き換えられる．Ravi らは，再暗号化テストにおけるサイドチャネル情報を解析することで，この復号失敗オラクルを実装した．本章でも，この復号失敗オラクルに基づく手法を採用し，以下に示す改良した方法を用いることにする．

Streamlined NTRU Prime への提案攻撃は以下の 2 ステップから成る．

1. 最初のステップでは，正しい平文 r_{valid} に対応する暗号文 $c = \mathsf{Round}(h \cdot r_{\mathrm{valid}})$ から，改変した暗号文 $c' = c + \delta$ を生成し，その復号結果を確認することで，秘密鍵

の 1 と衝突する δ を探す．もし復号に失敗したことを検知した際には，δ を c_{base} として使用する．Ravi らの手法では，δ の構造を注意深く設計している．提案手法では，攻撃成功確率を上昇させるために，δ の構造を少しだけ変更する．すなわち，δ に対する添字の範囲 $i_1, \ldots, i_m, j_1, \ldots, j_n$ を $[0, p)$ から $[\lfloor p/2 \rfloor, p)$ へ変更し，sntrup653 と sntrup1277 についてはパラメータ $(m, n) = (1, 3)$ とする．それ以外については，Ravi らの手法と同じ戦略で δ を設計し，適切な δ を得られる確率を推定する．ここで，sntrup653 と sntrup1277 については，それぞれパラメータを $(k_1, k_2) = (96, 282)$ と $(152, 486)$ とする．また，約 20% のノイズ $n'[i]$ については，$a[i] > q/2$ とする．詳しくは，[94, Section 4.1 and 4.2] を参照されたい．

2. 2 つ目のステップでは，攻撃者は 4 つの改変された暗号文 c と c_{base} を問い合わせ，秘密鍵の係数を決定するために復号結果が r_{valid} と r_{fixed} のどちらであるかを確認する．

NTRU と同様に PKE.Enc が決定的アルゴリズムなので, NTRU Prime の使用する FO 変換の亜種は乱数の計算を含まないことに注意されたい．ただし，NTRU Prime は明示的な再暗号化テストを用いており，追加のハッシュ $\mathsf{HashConfirm}(r, \mathsf{pk})$ を PKE の暗号文に加算する．ここで，$\mathsf{HashConfirm}(r, \mathsf{pk}) = \mathsf{Hash}(\mathsf{0x02} \parallel \mathsf{Hash}(\mathsf{0x03} \parallel r) \parallel \mathsf{Hash}(\mathsf{0x04} \parallel \mathsf{pk}))$ であり，$\mathsf{Hash}(z)$ は $\mathsf{SHA\text{-}512}(z)$ の最初の 32 バイトである．したがって，再暗号化テストの脱カプセル化アルゴリズムは $\mathsf{HashConfirm}(r', \mathsf{pk})$ を計算する．よって，r' の情報を，$\mathsf{HashConfirm}(r', \mathsf{pk})$ のサイドチャネル情報から窃取できる．

5.4.5　符号ベース KEM

■HQC　HQC は符号理論の問題に基づいているものの，Kyber ，Saber ，FrodoKEM ，NTRU LPRime などの格子ベース KEM と同じような構造を持っている．よって，HQC への KR-PCA は同じような戦略で実行できる．実際，HQC のラウンド 2 に対する Huguenin-Dumittan と Vaudenay らの KR-PCA [101] は，Bătetu [100] の異なる符号ベース PKE Lepton への攻撃を模倣したものである．HQC は第 2 ラウンドから第 3 ラウンドの間で，パラメータとデコーダが変更されたが，攻撃時のパラメータ設定を変更することで KR-PCA を実行できる．詳しくは，文献 [106] を参照されたい．彼らの攻撃では，復号された平文は 0^ℓ か，$0^{i-1} \parallel 1 \parallel 0^{\ell-i-1}$ となる．再暗号化において復号された平文を手に入れるために，HQC は SHAKE を使用する．よって，提案するサイドチャネル攻撃を SHAKE の処理に適用することで，平文判定オラクルを実現できる．

■BIKE　ラウンド 3 の BIKE では，QC-MDPC（Moderate Density Parity-Check）符号を用いた Niederreiter PKE ベースの KEM スキームを用いている．文献 [107] で，Guo

らは QC-MDPC に対する鍵回復攻撃として GJS（Guo-Johansson-Stankovski）攻撃を提案した．この攻撃では，秘密鍵の半分 $h_0 \in \mathbb{F}_2^n$ の距離プロファイル $\mu(h_0)$ を，復号オラクルを用いて回復する．距離プロファイルは，$d = 1, 2, \ldots, n/2$ として，(d, μ_d) を含む．すなわち，h_0 から距離 d を持つ μ_d 個の 1 のペアが複数存在する．80 ビットセキュリティ用のパラメータの場合では，距離プロファイル $\mu(h_0)$ から h_0 を回復することが，実際に可能であることを Guo らは報告している．Xagawa らは，平文判定オラクルが存在する場合に，BIKE の第 3 ラウンドに対して，QC-MDPC への GJS 攻撃が部分的に適用可能であることを報告した．彼らの手法は，128 ビットセキュリティ用のパラメータにおいて，約 4 分の 1 の距離プロファイルが回復できる．再暗号化で，BIKE の脱カプセル化は PRF を使用するため，平文判定オラクルをサイドチャネル情報から実装できる．

GJS 攻撃は，(d, μ_d) を計算するために，細工された不正なランダムな平文から作られた複数の暗号文を問い合わせし，復号結果が正しいかどうかを確認する．そのため，同攻撃においてテンプレートとなる平文は固定化できない．

■Classic McEliece　Classic McEliece の PKE に対する適応的攻撃は知られていないため，提案攻撃は Classic McEliece には適用できない．しかし，Classic McEliece の脱カプセル化に関して，平文判定オラクルを実現することはできる．したがって，もし Classic McEliece に対する KR-PCA が発見されれば，提案手法を Classic McEliece にも用いることが可能となる．

5.4.6　同種写像ベース KEM

ここでは，FO 変換に着目して，SIKE への新たなサイドチャネル攻撃を提案する．提案するサイドチャネル攻撃は Jao と De Fao の超特異同種暗号システムへの適応的攻撃を元にしている．SIKE.Decaps への提案サイドチャネル攻撃では，同攻撃を修正を行った．以下でそれについて説明する．

e_A と e_B を十分大きな正整数とし，$p = 2^{e_A}3^{e_B} \pm 1$ とする．E_0 を $\mathbb{F}_{p^2}$ 上のモンゴメリ曲線とし，P_A と Q_A，P_B，Q_B を E_0 の公開生成元とする．sk_2 と sk_3 をそれぞれアリスとボブの秘密鍵とする．アリスとボブの秘密の点を $R_A = P_A + [\mathsf{sk}_2]Q_A$ と $R_B = P_B + [\mathsf{sk}_3]Q_B$ とおく．$\mathbb{F}_{p^2}$ 上の楕円曲線 E_A と $\mathbb{F}_{p^2}$ 上定義された分離的な同種写像 $\phi_A : E_0 \to E_A$ を $\ker \phi_A = \langle R_A \rangle \subset E_0(\mathbb{F}_{p^2})$ となるように構成する．同様に，E_B と ϕ_B についても構成する．pk_2 と pk_3 をそれぞれアリスとボブの公開鍵とする．現実的には，アリスが攻撃者であり，ボブがサーバなどの攻撃対象となる．よって，SIKE への攻撃における攻撃者のゴールは，復号オラクルへの暗号文の問い合わせによって，ボブ側の秘密鍵 sk_3 を推定することである．

SIKE.Encaps では，アリスは参照用暗号文 (c_0, c_1) を生成するために，秘密の点 R_A と公開曲線 E_A と $\tilde{P_A}$ と $\tilde{Q_A}$ を計算する．ここで，この参照用暗号文に対応する参照用 j 不変量は，曲線 $E_0/\langle R_A, R_B\rangle$ である．ただし，$E_0/\langle R_A, R_B\rangle$ は，有限群 $\langle R_A, R_B\rangle \coloneqq \{\, [n_A]R_A + [n_B]R_B \mid n_A \in \{0, 1, \ldots, 2^{e_A} - 1\}, n_B \in \{0, 1, \ldots, 3^{e_B} - 1\}\,\}$ を核としてもつ同種写像に対して，E_0 と同種な楕円曲線を表す．SIKE.Encaps の有効な暗号文は，$c_0 = \mathsf{pk}_2 = (E_A, \tilde{P_A}, \tilde{Q_A})$ と $c_1 = m \oplus \mathsf{SHAKE}(j(E_0/\langle R_A, R_B\rangle))$ で与えられる．ここで，m は一様分布 $U(\{0,1\}^n), n \in \{128, 192, 256\}$ からサンプルされた乱数である．

攻撃において，秘密鍵の三進数表現 $\mathsf{sk}_3 = 3^0\beta_0 + 3^1\beta_1 + \cdots + 3^i\beta_i + \cdots + 3^{e_B-1}\beta_{e_B-1}, \beta_i \in \{0, 1, 2\}$ を考える．適応的選択暗号文攻撃では，三進数表示において，下位桁から上位桁に向かって鍵復元を各桁ごとに繰り返し行う．今，ある $i-1$ 桁目までの値 $\beta_0, \beta_1, \ldots, \beta_{i-1}$ がわかっているときに，i 桁目の値 β_i への攻撃を考える．秘密鍵の推定された部分を $K_i = 3^0\beta_0 + 3^1\beta_1 + \cdots + 3^{i-1}\beta_{i-1}$ とおく．攻撃者は，改変された暗号文 $(c_0^{(\tau,i)}, c_1), \tau \in \{0, 1, 2\}$ を生成する．ただし，各 τ に対応する $c_0^{(\tau,i)}$ は，$\tilde{P_A}$ と $\tilde{Q_A}$ を

$$\begin{aligned}\tilde{P}_A^{(\tau,i)} &= \tilde{P_A} - [3^{e_B-i-1}K_i + 3^{e_B-1}\tau]\tilde{Q_A},\\ \tilde{Q}_A^{(\tau,i)} &= \tilde{Q_A} + [3^{e_B-i-1}]\tilde{Q_A}\end{aligned}$$

と置き換えたものにする．参照用暗号文 (c_0, c_1) に対応する SIKE.Decaps における同種核の巡回群の生成元を $R_{AB} = \tilde{P_A} + [\mathsf{sk}_3]\tilde{Q_A}$ とする．すなわち，正しい暗号文に対して，復号オラクルは巡回群の生成元として R_{AB} を計算する．一方で，復号オラクルに $(c_0^{(\tau,i)}, c_1)$ を問い合わせたとき，生成元は

$$\begin{aligned}R_{AB}^{(\tau,i)} &= (\tilde{P_A} - [3^{e_B-i-1}K_i + 3^{e_B-1}\tau]\tilde{Q_A})\\ &\quad + [\mathsf{sk}_3](\tilde{Q_A} + [3^{e_B-i-1}]\tilde{Q_A})\\ &= R_{AB} + [3^{e_B-i-1}(\mathsf{sk}_3 - K_i) - 3^{e_B-1}\tau]\tilde{Q_A}\end{aligned}$$

となり，$E_A/\langle R_{AB}^{(\tau,i)}\rangle$ の j 不変量を計算する．ここで，点 $\tilde{Q_A}$ の位数は 3^{e_B} であるため，$\tau = \beta_i$ に等しいときそのときに限り，

$$\begin{aligned}&[3^{e_B-i-1}(\mathsf{sk}_3 - K_i) - 3^{e_B-1}\tau]\tilde{Q_A}\\ &= [3^{e_B-i-1}\sum_{j=i}^{e_B-1} 3^j\beta_j - 3^{e_B-1}\tau]\tilde{Q_A},\\ &= [3^{e_B-1}(\beta_i - \tau)]\tilde{Q_A}\end{aligned}$$

より，$R_{AB} = R_{AB}^{(\tau,i)}$ が成り立つ．よって，もし $R_{AB} = R_{AB}^{(\tau,i)}$ ならば，PKE の復号結果は参照用平分に等しく，そうでなければ，これらは等しくない．この事実を用いることで，平文判定オラクルを通して攻撃者は i 番目の桁 β_i を求めることができる．この攻撃は，

Algorithm 8 SIKE への KR-PCA

Require: Reference ciphertext (c_0, c_1) and reference plaintext m
Ensure: Secret key sk_3
1: **function** AttackOnSIKE($(c_0, c_1), m$)
2: $K_0 \leftarrow 0$;
3: **for** $i = 0$ to $e_B - 1$ **do**
4: **for all** $\tau \in \{0, 1, 2\}$ **do**
5: $\tilde{P}_A^{(\tau,i)} \leftarrow \tilde{P_A} - [3^{e_B-i-1}K_i + 3^{e_B-1}\tau]\tilde{Q_A}$;
6: $\tilde{Q}_A^{(\tau,i)} \leftarrow \tilde{Q_A} + [3^{e_B-i-1}]\tilde{Q_A}$;
7: $(c_0^{(\tau,i)}, c_1) \leftarrow ((E_A, \tilde{P}_A^{(\tau,i)}, \tilde{Q}_A^{(\tau,i)}), c_1)$;
8: **if** $\mathcal{O}((c_0^{(\tau,i)}, c_1), m) = 1$ **then**
9: $\beta_i \leftarrow \tau$;
10: $K_{i+1} \leftarrow K_i + 3^i\beta_i$;
11: **end if**
12: **end for**
13: **end for**
14: **return** K_{e_B};
15: **end function**

下位桁から順番に鍵値の復元を行うため，攻撃時に必要なオラクルアクセス回数は e_B に対して線形となる．

Algorithm 8 に SIKE への KR-PCA を示す．SIKE.Decaps では，j 不変量の値は c_0 にのみに依存するため，PKE の復号結果は固定された j 不変量と c_0 に対しては，いつもおなじになる．したがって，G への入力が参照用平文 m かそうでないかを G のサイドチャネル情報から知ることができ，これにより SIKE に対する平文判定オラクルを実現できる．Algorithm 8 では，8 行目で平文判定オラクル $\mathcal{O}$ を使用する．完全な鍵回復を行うために必要なオラクルアクセス回数は最大で $3e_B$ である．ただし，鍵値の各桁の推定において，取りうる 3 パターンのうち 2 パターンが外れた段階で，3 パターン目であることが確定するため，必要なオラクルアクセス回数は $2e_B$ まで減らせることに注意されたい．

5.4.7 攻撃の複雑さ

表 5.1 に NIST PQC KEM の第三ラウンドにおける各候補に対して，提案手法を用いて完全な鍵回復を行うにあたって必要なオラクルアクセス回数を示す．簡単のために，表 5.1 には，セキュリティレベルが AES128 と AES256 に等しい場合のときの結果のみを示す．

表から，鍵回復に必要なオラクルアクセス回数は，十分に現実的な範囲の数に収まっていることがわかる．最も攻撃が難しい BIKE において，セキュリティレベルが 1 の場合では，部分鍵推定に 300 万回のオラクルアクセスが必要だが，他の殆どの KEM は 6 万回ほどのオラクルアクセスで破ることができる．コードベースの KEM と比べて，Kyber,

表 5.1: 提案手法による鍵回復に必要なオラクルアクセス回数（Classic McEliece を除く）

KEM type	Scheme	Instance	# Oracle accesses
Lattice	Kyber	Kyber-512	1536 ($= 3 \times 512$)
		Kyber-1024	3072 ($= 3 \times 1024$)
	Saber	LightSaber-KEM	3072 ($= 4 \times 512 + 2 \times 512$)
		FireSaber-KEM	3072 ($= 3 \times 1024$)
	FrodoKEM	FrodoKEM-640	25600 ($= 5 \times 5120$)
		FrodoKEM-1344	43008 ($= 4 \times 10752$)
	NTRU	ntruhrss701	≈ 2804 ($= 4 \times 701$)
		ntruhps2048509	≈ 1018 ($= 2 \times 509$)
		ntruhps4096821	≈ 1642 ($= 2 \times 821$)
	NTRU Prime	ntrulpr653	1306 ($= 2 \times 653$)
		ntrulpr1277	2554 ($= 2 \times 1277$)
		sntrup653	2712 in avg. ($= 100/1 + 4 \times 653$)
		sntrup1277	5175 in avg. ($= 100/1.5 + 4 \times 1277$)
Code	HQC	hqc128	≈ 18111 ($= 46 + \log(46) + 46 \times (384 + \log(384))$)
		hqc256	≈ 58536 ($= 90 + \log(90) + 90 \times (640 + \log(640))$)
	BIKE	Level 1	3M ($= 2000 \times 1500$)
		Level 5	N/A
	Classic McEliece	Any	N/A
Isogeny	SIKE	SIKEp434	274 ($= 2 \times 137$)
		SIKEp751	478 ($= 2 \times 239$)

Saber, NTRU, NTRU Prime, SIKE の推定しなければならない秘密係数の数は少ないため，攻撃に必要なオラクルアクセス回数も少なくなる．

従来の格子暗号に対する，選択暗号文攻撃ベースのサイドチャネル攻撃に比べて，提案手法はより多くのオラクルアクセス回数が必要になる場合がある．これは，提案手法が特定のスキームや実装に依存しないためである．ただし，それにより提案手法は，比較的ブラックボックスな実装に対しても適用できる点で優位性がある．

5.5　サイドチャネル識別器の設計

本節では，DL ベースのサイドチャネル識別器の設計について説明する．提案手法で用いたモデルは CNN と MLP である．これらはいくつかの DL-SCA の従来研究で，その有効性や実用性が示されている．実験で使用する CNN/MLP は，攻撃対象となる PRF の実装の違い（ソフトウェア実装やハードウェア実装，もしくはマスキングなどの対策の有無）をすべて吸収できるような，十分な表現力をもつものを使用した．提案するサイドチャネル攻撃では，PRF の入力が参照用平文かどうかの 2 クラス分類を行う必要があるため，モデルの出力は一次元とし，Sigmoid 活性化関数を使用した．

鍵回復を正しく行うために，KR-PCA では非常に高い精度をもつモデルが必要となる．

しかし，使用したモデルの精度は，第 4 章で述べたサイドチャネル波形に含まれるノイズや対策によって，無視できないほどに小さくなる可能性がある．モデルによって実現されるオラクルの精度を上昇させるためには，複数波形を使って一回の平文判定オラクルアクセスを実現する必要がある．具体的には，攻撃者は，改変した暗号文 c' に対して N 個の波形を取得し，それぞれの波形に対して推論を実行し，すべての推論結果にたいする多数決として PRF の入力の推定を行う．モデルの精度を ρ とすると，一回のオラクルアクセスの成功確率は

$$\rho_N = 1 - \sum_{s=0}^{\lceil N/2 \rceil} \binom{N}{s} \rho^s (1-\rho)^{N-s} \tag{1}$$

と与えられる．攻撃全体において達成したい攻撃成功確率を γ とし，表 5.1 に示した必要なオラクルアクセス回数を u とすれば，$\gamma \geq \rho_N^u$ を満たす必要がある．

以上の多数決による方法は，シンプルな一方で，ニューラルネットワークの出力が確率値であることを無視しており，情報損失が存在する．多数決よりも効率的なオラクルの実現方法として，尤度の使用が考えられる．モデルのパラメータを $\hat{\theta}$，入力波形を $\boldsymbol{x}_i$，オラクルの出力を $z \in \{0, 1\}$ とすれば，モデルを条件付き確率 $p(z \mid \boldsymbol{x}_i; \hat{\theta})$ と表現できる．これを用いて，負の対数尤度は

$$\mathrm{NLL}(\hat{\theta}) = -\frac{1}{N} \sum_{i=1}^{N} \log p(z \mid \boldsymbol{x}_i; \hat{\theta}) \tag{2}$$

となる．多数決による方法と比べて，モデルの確率（確信度）を用いている点で，より効率的に平文判定オラクルを実現できる．実際，確率分布 $p(z \mid \boldsymbol{x}; \hat{\theta})$ が真の分布に一致している理想的な状況では，この尤度比較は最強力検定となることが知られている [108]．その意味で，尤度（比）に基づく方法は，多数決よりも優位性がある．

5.6 実験

5.6.1 実験環境

本節では実験環境について説明する．モデルの学習には，ライブラリとして CUDA 11.0，cuDNN 8.0.5，Tensorflow 2.4.1 と Keras 2.4.0 を使用し，ワークステーションは Intel Xeon W-2145 3.70 GHz と NVIDIA GeForce GTX 2080 が搭載されたものを使用した．学習率は 0.001 とし，バッチサイズは 32，エポック数は 100 とした．表 5.2 に波形のサンプル数が 1,000 のときの，CNN のアーキテクチャを示す．表の上側が入力側であり，下側が出力側のパラメータである．表の “Input” の列の $S_1 \times S_2$ は，S_1 が入力の長さ（波形のサイズ）であり，S_2 が入力チャンネルサイズである．“Operator”

表 5.2: ニューラルネットワークのハイパーパラメータ

(a) 未対策ソフトウェア，未対策ハードウェア，マスク対策されたハードウェア実装向けのアーキテクチャ

	Input	Operator	Output	Activation function	Batch normalization	Pooling	Stride
Conv1	1000×1	conv1d(3)	4	SELU	Yes	Avg (2)	2
Conv2	500×4	conv1d(3)	4	SELU	Yes	Avg (2)	2
Conv3	250×4	conv1d(3)	4	SELU	Yes	Avg (2)	2
Conv4	125×4	conv1d(3)	8	SELU	Yes	Avg (2)	2
Conv5	62×8	conv1d(3)	8	SELU	Yes	Avg (2)	2
Conv6	31×8	conv1d(3)	8	SELU	Yes	Avg (2)	2
FLT	15×8	flatten	120	-	-	-	-
FC1	120	dense	20	SELU	No	No	-
FC2	20	dense	20	SELU	No	No	-
FC3	20	dense	1	Sigmoid	No	No	-

(b) マスク対策されたソフトウェア実装向けのアーキテクチャ

	Input	Operator	Output	Activation function	Batch normalization	Pooling	Stride
FC1	100	dense	32	SELU	No	No	-
FC2	32	dense	32	SELU	No	No	-
FC3	32	dense	20	SELU	No	No	-
FC4	20	dense	20	SELU	No	No	-
FC5	20	dense	1	Sigmoid	No	No	-

の列の conv1d(F) は各層の演算の種類と，F はフィルタサイズを表す．表 5.2(a) は対策なしソフトウェア/ハードウェア実装と，マスク対策済みハードウェア実装用のアーキテクチャである．使用した CNN は 6 層の畳み込み層 *Conv1*，*Conv2*, . . . , *Conv6* と，三層の全結合層 *FC1*，*FC2*，*FC3* からなる．表 5.2(b) は，マスク対策済みソフトウェア実装用のものであり，5 つの全結合層 *FC1*，*FC2*，. . . ，*FC5* からなる．(a) と (b) の出力層は 1 次元であり，活性化関数に Sigmoid 関数を用いた．サイドチャネル波形が与えられたとき，CNN/MLP の出力は PRF の入力が参照用平分である確率を与える．

表 5.3 に実験で使用した実装と，学習に使用した波形数を示す．[*1] 本実験では，対策なし AES と SHAKE ソフトウェア実装，対策なし AES ハードウェア実装，マスキング対策ありの AES ソフトウェア実装，マスキング対策ありの AES ハードウェア実装の 5 つの実装を用いた．マスキング対策実装として，ソフトウェアについてはビットスライスソ

[*1] 表で示したもの以外に，`ntruhrss701` の NTRU ソフトウェア実装を `pqm4` 上で評価した．ここでは，`owcpa_dec` の正当性判定の処理におけるサイドチャネル情報を用いて攻撃を行った．本実験では対象としなかったが，正当性判定の部分ではなく，NTRU.Decaps の PRF の処理の漏洩を用いても，NTRU の鍵回復攻撃を実現可能である．実験の結果として，DL に基づくサイドチャネル識別器により，99.8% の精度を達成し，尤度比を用いることで 2 波形で 100% のテスト精度を実現した．

表 5.3: 実験に使用した実装と使用した波形数

	Non-protected AES/SHAKE software	Non-protected AES hardware	Masked bit-sliced AES software	Masked AES hardware based on TI
Reference	`pqm4` [109, 110]	SASEBO IP [111]	Schwabe and Stoffelen [112, 113]	Ueno et.al. [33]
Device	STM32F415RGT6	Xilinx Kintex-7	STM32F407VGT6U	Xilinx Kintex-7
Board	NewAE Technology STM32F	SAKURA-X	STM32F407G-DISC1	SAKURA-X
Side-channel trace	Supply voltage current	Supply voltage current	EM radiation	Supply voltage current
Measurement interface	NewAE technology chip-whisperer CW308	On-board coaxial connector	Langer EMV-Technik RF-U T-2 probe	On-board coaxial connector
Oscilloscope	Keysight Technologies MSOX6004A			
# Training traces	30,000	30,000	900,000	980,000
# Validation traces	10,000			
# Test traces	10,000			

フトウェア，ハードウェアについては TI に基づくハードウェアを使用した．再現性のために，対策なしのソフトウェア/ハードウェア実装と対策ありのソフトウェア実装については，オープンソース実装を利用した．一方，TI によるマスキング対策ありのハードウェア実装については，利用可能なオープンソース実装が存在しなかったため，文献 [33] の再現実装を行い，評価した．本実験で使用した対策あり実装の殆どは，大多数の KEM スキームで使用されている SHAKE ではなく，AES である．これは，サイドチャネル攻撃の対策の研究の多くが AES を対象としたものであり，SHAKE への対策手法については，現状では十分に検討されていないためである．事実，マスキング対策がされた SHAKE の実装で，利用可能なオープンソースのものは，我々が知る限り存在しない．ただし，同じマスキングスキームを使用して対策を行った場合は，AES と SHAKE で攻撃結果に大きな違いは発生しないと予想される．

識別攻撃の性能評価のために，固定とランダムの間の TVLA（Test Vector Leakage Assessment）と同様の方法で，サイドチャネル波形の取得を行った．具体的には，AES については，鍵を固定し，入力平文が固定とランダムのそれぞれの場合のサイドチャネル波形を取得した．SHAKE についても，AES の場合と同様に平文を与えた．

マスキング対策実装に対する実験では，サイドチャネル波形から最初のマスクの初期化を除去して，マスキング対策された演算処理部分のみを使用した．いくつかの先行研究で，PKE.Dec を含む脱カプセル化処理のすべてをマスキング対策した実装が提案されており，このような実装への攻撃を想定して，マスクされた演算からの漏洩のみを用いた．

表 5.4: PRF の入力の分類精度

	未対策 ソフトウェア	未対策 ハードウェア	マスク付き ソフトウェア	マスク付き ハードウェア
精度	0.998	0.999	0.960	0.515

5.6.2　精度評価

表 5.4 に，学習済みモデルのテストデータに対する精度を示す．表から，マスキング対策済みハードウェア実装を除いたすべての実装で，学習済みモデルが十分に高い精度を達成したことを確認できる．対策なしのソフトウェアとハードウェア実装に対して，それぞれ 99.8% と 99.9% のテスト精度を得た．加えて，マスク対策ありのソフトウェア実装であっても 96.0% の精度だった．マスキング対策は識別攻撃の性能を減少させるが，攻撃に対する対策としては不十分であることがわかる．一方で，TI に基づくマスキング対策ハードウェア実装は，識別攻撃が難しいことが確認された．

次に，多数決と尤度の比較のそれぞれの場合における，オラクルの精度の評価を行った．多数決については，奇数の波形数の場合について，式 (1) を用いて解析的な評価を実施した．尤度比較については，解析的に調べることが難しいため，攻撃成功確率を実際の攻撃評価から算出した．具体的には，まず一様ランダムに正解ラベル z_{true} を 0 か 1 に決め，テストデータから正解ラベルのクラスのサンプルを，ランダムに N 波形取り出す．そして，各クラスラベル（0 もしくは 1）に関する NLL を，サンプリングした波形から計算し，それらの大小関係から予測された正解ラベル $\hat{z_N}$ を求める．N 波形を攻撃に使用したときの正答率は，予測ラベル $\hat{z_N}$ が z_{true} に一致したかどうかを，この操作を何度も繰り返して数え上げることで計算できる．本実験では，この操作を 1 万回実施して，正答率を求めた．結果的に，尤度の比較による方法を使用したとき，対策なしソフトウェアとハードウェア実装，およびマスク対策ありのソフトウェア実装に対して，それぞれ 2，2，5 波形で 100.0% の精度が得られた．一方で，多数決を使用したときは，99.999% を達成するにあたって，それぞれ 5，5，11 波形を必要とした．多数決と比べて，尤度を使用したほうが精度が良い結果が得られたのは，尤度比較のほうがモデルの出力をより有効活用できるためだと考えられる．一方で，対策ありのハードウェア実装については，5,000 波形使用した場合でも 100% の精度を達成することは困難なことが判明した．

表 5.5: 提案攻撃に必要なサイドチャネル波形数

KEM type	Scheme	Instance	# Traces for attack phase	
			Non-masked implementations	Masked software
Lattice	Kyber	Kyber-512	3,072	7,680
		Kyber-1024	6,144	15,360
	Saber	LightSaber-KEM	6,144	15,360
		FireSaber-KEM	6,144	15,360
	FrodoKEM	FrodoKEM-640	51,200	128,000
		FrodoKEM-1344	86,016	215,040
	NTRU	ntruhrss701	5,608	14,020
		ntruhps2048509	2,036	5,090
		ntruhps4096821	3,284	8,210
	NTRU Prime	ntrulpr653	2,612	6,530
		ntrulpr1277	5,108	12,770
		sntrup653	5,424	13,560
		sntrup1277	10,350	25,875
Code	HQC	hqc128	36,222	90,555
		hqc256	117,072	292,680
	BIKE	Level 1	6M	15M
		Level 5	N/A	N/A
	Classic McEliece	Any	N/A	N/A
Isogeny	SIKE	SIKEp434	548	1,370
		SIKEp751	956	2,390

5.6.3 鍵復元に必要な波形数の評価

表 5.5 に前節で述べたサイドチャネル識別器を使用して提案攻撃を行った場合に，各 KEM スキームで必要な波形数を示す．尤度比較を使用した場合，100% の精度を達成するのに必要な波形数が 2 から 5 であることから，表 5.1 の値の 2 から 5 倍程度の波形数が攻撃に必要となる．

表から，最も攻撃が難しいのは，150 万波形を必要とする BIKE であることがわかる．ただし，マスク対策に対するサイドチャネル攻撃評価では，100 万や 1,000 万波形が使用されることも多いため [114, 36, 115]，150 万波形はまだ攻撃が可能だと考えられる．

一方で，TI による対策がされたハードウェア実装では，KR-PCA を行うのに必要な精

度を達成できなかったため，表 5.5 には掲載しなかった．よって，TI が提案手法に対する有効な対策手法になると考えられる．ただし，DL-SCA の性能が今後向上していけば，現実的な波形数で攻撃が可能になる可能性がある．

5.7　結び

本章では，暗号モジュールの物理的安全性評価として，DL-SCA を用いた公開鍵暗号モジュールの脆弱性検知手法を提案した．本手法において対象としたのは，PQC に基づく KEM スキームである．提案手法では，平文判定オラクルを DL-SCA により実現し，CCA2 を行うことで秘密鍵の窃取が可能となることを示した．具体的には，提案する DL-SCA は，平文判定オラクルを実現するために，PRF の入力が参照用平文かどうかをサイドチャネル情報から識別するモデルを使用する．NIST PQC の第三ラウンドの KEM スキームのうち，Classic McEliece を除く全てに対する攻撃方法について述べ，DL-SCA を通してその有効性を明らかにした．提案攻撃に対して，NIST PQC のすべての KEM スキームにおいて，適切な対策（例えば，ハードウェアマスキングなど）が必要であることを示した．

第 6 章

結言

以上，第 2 章から第 5 章まで，サイドチャネル攻撃対策された暗号モジュールの安全性評価手法の確立を目的として，(1) 論理的安全性評価手法としてガロア体算術に基づく暗号ハードウェアの効率的な検証技術の開発，(2) 物理的安全性評価手法として暗号モジュールの DL-SCA に対する安全性評価について述べた．

第 2 章では，暗号モジュールの安全性評価手法に関する基礎的考察を行った．まず，暗号とその実装技術に関する概要を述べ，暗号アルゴリズムに関する基礎的考察を行った．共通鍵暗号方式と公開鍵暗号方式について説明し，それぞれの代表的なアルゴリズムである AES と ECC の概要を述べた．また，アルゴリズム自体の安全性だけでなく，実装の安全性も重要となることを述べ，物理攻撃について概説した．物理攻撃の中でも，サイドチャネル攻撃について，その分類と対策手法を述べた．

第 3 章では，暗号ハードウェアの設計時における安全性評価手法の開発を目的として，設計仕様とゲートレベルネットリスト間の等価性検証の開発をした．まず，暗号モジュールでは，特定の入力でのみ引き起こすような故障であっても，秘密情報の漏洩に直結することを述べ，完全検証の必要性を説明した．次に，多くの暗号で用いられるガロア体算術演算回路の検証手法について述べ，従来手法における問題点を説明した．本稿では，その解決策として，ZDD に適した多項式簡約アルゴリズムと，多標数ガロア体算術演算回路の等価性検証のための新たな検証手法の提案を行った．AES や ECC などの実用的な暗号回路の検証実験を通して，提案手法の有効性を示した．加えて，提案手法の応用として，特定の入力でのみ故障を引き起こす HT の検知手法を提案した．

第 4 章では，共通鍵暗号への DL-SCA による安全性評価技術を開発した．まず，従来の DL-SCA による安全性評価において問題となっていた不均衡データ問題について述べ，モデルの分布がラベルの生起確率（二項分布）にバイアスされることが原因であることを説明した．次に，不均衡データによってモデルが受ける悪影響を定量的に図るための方法として，KL ダイバージェンスを用いた指標を提案した．提案指標は，モデルの分布と二

項分布の間の KL ダイバージェンスから計算され，値が小さいほど攻撃が難しくなることを表す．そして，先行研究で報告されてた CER ロス関数が，提案した指標を増加させるように学習を行うことで，不均衡データ問題を解消していることを述べた．また，本論文では，不均衡データ問題の解決策として，鍵値に基づく尤度を提案した．提案する鍵の尤度は，生起確率が不均衡ではない鍵の確率から導出されるため，従来の HW/HD の尤度における不均衡データ問題を回避できる．先行研究で，データ拡張手法の 1 つである SMOTE が，不均衡データ問題に対して有効であることが報告されていたが，その理由を鍵の尤度とベイズの定理から説明した．最後に，2 つのデータセットに対する様々な対照実験を通して，本章で述べた仮説の妥当性と，提案手法の有効性を確認した．

第 5 章では，NIST PQC の KEM スキームに対する平文判定オラクルを使用した攻撃の可能性を指摘した．まず，現在行われている KEM のコンペティションのすべての候補において，FO 変換が共通して用いられていることを述べた．そして，FO 変換の再暗号化による暗号文の正当性確認処理に対して，平文判定オラクルを用いることで，適応的選択暗号文攻撃が可能であることを説明した．本論文では，FO 変換で使用されるランダムオラクル（ハッシュ関数）のサイドチャネル情報に対して，DL-SCA を適用することで，平文判定オラクルを実現可能なことを示した．また，Classic McEliece を除く全ての KEM スキームについて，提案する平文判定オラクルを用いた適応的選択暗号文攻撃の実現方法を述べた．最後に，実験を通して，マスキング対策された AES ハードウェア実装を除く，すべてのハッシュ関数で提案攻撃に対する脆弱性が存在することを実証した．また，ハッシュ関数にマスク付きハードウェア実装を用いることが，提案攻撃に対する対策となることを示した．

今後の展望としては，ソフトウェア実装の安全性評価手法を確立することが挙げられる．本稿では，ハードウェア実装の設計時における等価性検証手法を提案したが，ソフトウェア実装についても同等の手法が不可欠であると考えられる．また，本論文で提案した等価性検証手法では，脆弱性検知のために，ゲートレベルネットリストの設計仕様が必要となる．しかし，現実の設計開発では，必ずしもネットリストの詳細な設計仕様が入手できない可能性がある．したがって，細かな設計仕様がない場合にも，脆弱性を効率的に検知可能な手法の検討が今後の課題として挙げられる．第 4 章と第 5 章では，DL-SCA を用いた物理的安全性評価手法の提案を行った．しかし，今後もサイドチャネル攻撃の高度化・最適化が行われていくことを考えれば，DL-SCA のような特定の攻撃を用いた安全性評価は，その場しのぎの対策手法でしかない．したがって，今後は，マスキング対策における d-Probing モデルのような，数学的な背景に基づく証明可能安全性の立場から，特定の攻撃によらない安全性評価を行う必要がある．

付録 A

マスキング対策の理論的安全性

A.1　はじめに

本章では，証明可能安全なマスキングスキームについて理論的な解析を行う．第 2.3.3 節で述べたとおり，マスキングは最も一般的なサイドチャネル攻撃対策である．マスキング対策実装では秘密の中間値をシェアと呼ばれる乱数値に加法的に分解する．ここで，シェアの数（マスキング次数）と実装コストとの間にはトレードオフが存在し，攻撃成功に必要な波形数と実装コストはマスキングの次数に対してそれぞれ指数オーダと二次オーダで増加するとされている [116]．したがって，サイドチャネル耐性を有する暗号モジュール設計時には安全性と設計コストの間のトレードオフを考慮してマスキングを実装する必要がある．一方で，これまで知られているマスキング次数と必要な波形数の関係の評価はタイトさ・正確さの観点から実用が困難である．以上の背景から，マスキング次数を増加させたときの攻撃難易度（攻撃成功確率や攻撃成功に必要な波形数）を正確に評価する方法が強く望まれている．

そこで本章では，情報理論的観点から，マスキング対策された暗号実装に対するサイドチャネル攻撃の攻撃成功確率 (SR: Success rate) の上界を導出する．先行研究において，サイドチャネル攻撃を通信路としてモデル化し，さらにサイドチャネルによる情報漏洩の強度を相互情報量として表現することで，未対策実装に対する SR の上界が暗号演算の中間値とサイドチャネル波形との間の相互情報量で与えられることが示された [76]．本章では，これを拡張することで，マスキング対策された実装に対する SR の上界を，各シェアに関する情報だけから導出する．導出された上界を用いることで，マスキング対策実装に対するサイドチャネル攻撃において攻撃成功（秘密鍵回復）に最低限必要な波形数を推定できる．本章では，(1) 各シェアの値と波形（漏洩）との間の相互情報量が既知の場合と，(2) 波形（漏洩）におけるシェアの値の条件付き確率が既知の場合の 2 種類の上界を導く．各シェアの値に関する相互情報量は，(2) の条件付き確率から求められるため，条件 (2) の方が条件 (1) よりも強い（上界としてタイトな）不等式が得られる．ただし，条件付き

確率は，相互情報量よりも推定が難しい．したがって，本章で導く二つの上界には利便性と正確性との間でトレードオフが存在する．よって，評価者が状況に応じて，(1) と (2) のどちらの不等式を用いるか選択できる．

条件 (1) と (2) に対応する上界の導出は，さらに次の二つの貢献を与える．一つは，条件 (1) の上界を用いて，マスキングの次数を増やすことで SR を指数的に減少可能であることを証明できる．同様の結果は，文献 [117] で Duc らによって示されていたが，同文献の結果が成立するのは，各シェアに関する相互情報量 $I(S;\boldsymbol{L})$ が 2^{-2n+1} 以下のときのみであった．ここで n は中間値のビット数である．これは，例えば中間値が 8 ビットであれば，$I(S;\boldsymbol{L}) < 2^{-15} \approx 3.05 \times 10^{-5}$ を満たさなければ，マスキング対策の有効性が不明だったことを意味する[*1]．一方，本章で得られる結果は，$I(S;\boldsymbol{L}) < 1/(2\ln(2)) \approx 0.72$ で成立するため，Duc らの結果よりも，より一般的な場合で，マスキング対策が有効であることを示している．

もう一つの貢献は，条件 (2) の SR の上界を導出する際に必要な条件付き確率を，深層学習を用いて推定する手法の提案である．これまで条件 (1) の上界の計算に必要な各シェアの相互情報量の推定手法は従来研究で報告されているものの，条件 (2) の上界の計算に必要な条件付き確率の推定手法はほぼ報告されていないためである．提案手法を用いることで，任意のマスキング次数に対する SR の上界をより正確に推定可能である．実験により，提案手法を用いて導出する上界が従来のマスキング実装に対する SR の上界と比較してよりタイトなことを示す．

A.2　関連研究

マスキング対策されたブロック暗号に対するサイドチャネル攻撃の SR と攻撃に必要な波形数の関係の理論的な解析を行った先行研究は，(i) Noisy Leakage Model (NLM) に基づくもの，(ii) 相互情報量に基づくものの二つに大別される．

(i) の NLM は攻撃者がサイドチャネル情報から “ノイズ” が付加された配線値[*2] を得られるというモデルである．Duc らは，波形が与えられたときの中間値の分布と，中間値の生起確率分布との間の統計（全変動）距離を用いて，“ノイズ” をモデル化することで，NLM を Random probing model (RPM) へ帰着可能なことを証明した．RPM における攻撃成功確率の上界は容易に計算可能であるため，結果として NLM における SR の上界

[*1] もし中間値がガウスノイズとともに漏洩する場合，SNR はおおまかに SNR $\approx 2I(S;\boldsymbol{L})$ と近似される．すなわち，Duc らの結果は，SNR が 6.1×10^{-5} を下回らなければ意味をなさず，実際のデバイスで SNR がこれほど小さいことは非現実的なことから，今まではマスキング対策による安全性は不明瞭であったと言える．

[*2] ソフトウェアで実現される論理回路評価を含むため，必ずしもハードウェア実装を対象としないことに注意されたい．

が間接的に求められる．Duc らは，この結果から，SR がマスキング次数の増大に対して指数的に減少することを示した．一方で，NLM から RPM への帰着は，ノイズが極めて大きい場合にしか意味を成さないことが知られている [118]．事実，文献 [76] では，サイドチャネル波形の SNR が 10^{-4} を下回らなければ，Duc の不等式は攻撃成功に少なくとも 1 波形必要という，自明な評価しか与えないことを実験的に示している．多くの現実のデバイス・計測環境では SNR が 10^{-4} を上回るため，暗号モジュールの設計時の評価に Duc らの結果を用いることは難しい．

(ii) の解析では，Chérisey らが文献 [76] において，サイドチャネル攻撃を通信路とみなし，中間値とサイドチャネル情報との間の相互情報量を用いて SR の波形数に対する上界を与えた．同文献では，未対策のブロック暗号に対するサイドチャネル攻撃を対象としている．さらに，Masure らは TCHES 2020 で，DL-SCA で一般的に使用される Cross Entropy（CE）損失関数が，Perceived Information に漸近的に一致し，相互情報量の推定に使用できることを述べた [119]．また，同文献で，Chérisey らの上界を DL-SCA を用いて推定可能なことを示した．一方で，Chérisey らの通信路のモデル化は，マスキング対策された実装の評価に使用できない．また，Masure らの方法では，実際に暗号モジュールへ DL-SCA を行わなければ攻撃成功確率の評価ができないため，マスキングの次数 d が増加したときの SR の推定に使用することは難しい．

以上要するに，(i) の解析手法は実用性に欠けるという課題があり，(ii) の解析手法は (i) の課題を解決する方向性を示しているが，現在搭載が一般的となっているマスキング対策されている実装に適用困難という課題を抱えていた．そこで，本章では，(ii) の解析手法を拡張することで，マスキング対策された実装にも適用可能な手法の提案を行う．

A.3 数学的準備

本節では，証明で使用するアダマール変換について述べ，さらにマスキング対策された実装へのサイドチャネル攻撃の，通信路に基づくモデル化を示す．

A.3.1 アダマール変換

本節では，証明に使用するアダマール変換 (WHT) について述べる．アダマール変換とは $\mathbb{F}_2 = \mathbb{Z}/2\mathbb{Z}$ における離散フーリエ変換 (DFT) であり，関数 $f : \mathbb{F}_2^n \to \mathbb{R}$ の WHT は

$$\hat{f}(w) = \sum_{s \in \mathbb{F}_2^n} f(s)(-1)^{\langle w \cdot s \rangle}$$

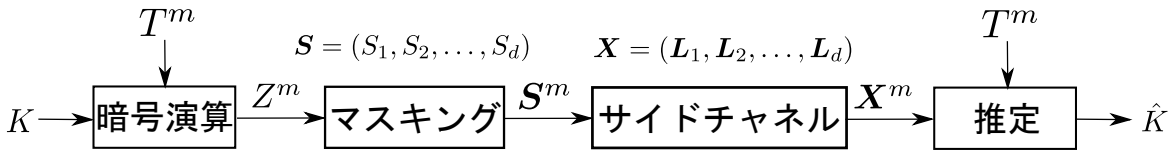


図 A.1: 通信路

と定義される．ここで，$\langle w \cdot s \rangle$ は，w と s の，各要素の積和を表す．WHT には逆変換が存在し

$$f(s) = \frac{1}{2^n} \sum_{w \in \mathbb{F}_2^n} \hat{f}(w)(-1)^{\langle w \cdot s \rangle}$$

が成り立つ．WHT により $\mathbb{F}_2^n$ 上の畳込みを積に分解できる．$\mathbb{F}_2^n$ から $\mathbb{R}$ への二つの関数 f_1, f_2 の畳込み $f(z)$ を

$$f(z) = \sum_{s_1 \oplus s_2 = z} f_1(s_1) f_2(s_2) = (f_1 * f_2)(z)$$

と定義する．このとき，これらの WHT に対し $\hat{f}(w) = \hat{f}_1(w)\hat{f}_2(w)$ が成り立つ．また，WHT においてもパーセバルの等式が成立し，

$$\sum_z f^2(z) = 2^{-n} \sum_w \hat{f}^2(w) = 2^{-n} \sum_w \hat{f}_1^2(w)\hat{f}_2^2(w)$$

が成り立つ．以上の議論は，2 以上の関数の畳み込みの場合でも成立する．

A.3.2 サイドチャネル攻撃の通信路モデル

本章では，Chérisey らの解析 [76] と同様にサイドチャネル攻撃を通信路としてモデル化する．図 A.1 に本章で用いる通信路モデルを示す．このモデルは Chérisey らのモデルの拡張である．図中の記号の意味を以下に示す．

- $m \in \mathbb{N}$ は波形数を表す．
- K と $\hat{K}$ は秘密鍵と推定鍵の確率変数を表す．これらの鍵は $\{0,1\}^n$ に値を取る n ビット部分鍵とする（例えば AES なら $n = 8$）．
- $T^m = (T_1, T_2, \ldots, T_m)$ は，平文もしくは暗号文の確率変数を表す．ここで，T^m は m 個の系列であることを意味する．$T_1, T_2, \ldots, T_m$ は，n ビット部分平文もしくは暗号文空間 $\{0,1\}^n$ 上の確率変数である．本章では，入力 $T_1, T_2, \ldots, T_m$ は，独立同分布（IID）かつ一様分布に従うと仮定する．
- $Z^m = (Z_1, Z_2, \ldots, Z_m)$ は，部分鍵 K と入力 T^m から計算される中間値の確率変数である．本章では，文献 [76] に従い，中間値 Z は全単射な関数 $\phi : \{0,1\}^n \to$

$\{0,1\}^n$ によって，$Z = \phi(K \oplus T)$ と与えられるとする．例えば，AES であれば，$i \in \{1,\ldots,m\}$ として，$Z_i = \mathrm{Sbox}(K \oplus T_i)$ と計算される．ここで，T_i が IID かつ一様分布であることから，Z_i も IID かつ一様分布となる．

- $\boldsymbol{S}^m = (\boldsymbol{S}_1, \boldsymbol{S}_2, \ldots, \boldsymbol{S}_m)$ は，マスキングのシェアの確率変数を表す．中間値 Z と，そのシェア $\boldsymbol{S} = (S_1, S_2, \ldots, S_d)$ について，$Z = S_1 \oplus S_2 \oplus \cdots \oplus S_d$ が成立する．ここで，d はシェアの数（$d-1$ がマスキングの最大次数）を表す．本章では，各シェア $S_1, S_2, \ldots, S_d$ は IID かつ一様分布と仮定する．
- $\boldsymbol{X}^m$ は，サイドチャネル波形の確率変数を表す．各観測におけるサイドチャネル波形 $\boldsymbol{X}$ は，各シェアに対応する漏洩（部分波形）$\boldsymbol{L}_1, \boldsymbol{L}_2, \ldots, \boldsymbol{L}_i, \ldots, \boldsymbol{L}_d$ を含む．各漏洩 $\boldsymbol{L}_i$ は $\mathbb{R}^{n_l}$ 上の確率変数とする．ここで，n_l は漏洩のサンプル点数とする．漏洩 $\boldsymbol{L}_i$ はシェア S_i にのみ依存し他のシェアには依存しないと仮定する．また，$\boldsymbol{L}_1, \boldsymbol{L}_2, \ldots, \boldsymbol{L}_d$ は IID とする．

図 A.1 の通信路から，マルコフ連鎖 $K \leftrightarrow Z^m \leftrightarrow \boldsymbol{S}^m \leftrightarrow \boldsymbol{X}^m \leftrightarrow \hat{K}$ が成立する．

この通信路は，マスキング対策実装に対するサイドチャネル攻撃では，秘密中間値 Z が d 個のシェア $S_1, S_2, \ldots, S_d$ に分解（マスキング）されており，攻撃者は各シェア S_i に関する情報を対応する漏洩 $\boldsymbol{L}_i$ から得ることを表している．このとき，マスキングの次数は高々 $d-1$ である．さらに，秘密中間値 Z に関して漏洩 $\boldsymbol{L}_i$ を全て含むサイドチャネル波形全体 $\boldsymbol{X}$ を用いてマスクされる前の秘密中間値 Z に関する情報を得ることを想定している．これは，これは $d-1$ 次マスキングに対する d 次プロービング攻撃に相当する．本章では，攻撃者が S_i に関して $\boldsymbol{L}_i$ から得る情報を相互情報量として $I(S_i; \boldsymbol{L}_i)$ と表す．また，Z に関して $\boldsymbol{X}$ から得る情報を同様に $I(Z; \boldsymbol{X})$ と表す．

A.4 攻撃成功確率の上界の導出

本節では，マスキング対策実装に対しても適用可能なサイドチャネル攻撃成功確率の上界を導出する．本節で導出する上界は，Cherisey らの結果のマスキング実装への一般化である．本節では，(1) 各シェアの値と波形（漏洩）との間の相互情報量が既知の場合と，(2) 波形（漏洩）におけるシェアの値の条件付き確率が既知の場合の 2 種類の上界を導く．そのために，まず，中間値とサイドチャネル情報との間の相互情報量 $I(Z^m; \boldsymbol{X}^m \mid T^m)$ を用いて SR の上界を導く（補題 3）．次に，m 波形使用して得られる情報量 $I(Z^m; \boldsymbol{X}^m \mid T^m)$ が，1 波形あたりに取得できる情報量 $I(Z; \boldsymbol{X})$ の m 倍で上から抑えられることを示す（補題 4）．また，相互情報量 $I(Z; \boldsymbol{X})$ を，各シェアの条件付き確率 $q(S_i \mid \boldsymbol{L}_i), i \in \{1,\ldots,d\}$ のアダマール変換で上から押さえられることを示す（補題 5）．補題 4 から得られる命題 2 と補題 5 から，攻撃成功確率は各シェアの条件付き確率を用いて上から抑えられること

がわかる（定理 4）．条件 (2) の場合は，ここまでの結果から SR の上界が得られる．一方，条件 (1) の場合は，各シェアの条件付き確率の分布は未知であるため，さらに補題 5 から，各シェアの相互情報量 $I(S_1;\boldsymbol{L}_1),\ldots,I(S_d;\boldsymbol{L}_d)$ を用いた上界を導く（定理 5）．ここで条件 (1) に対応するのが定理 5 であり，条件 (2) に対応するのが定理 4 である．これらを用いることで，マスキング実装に対する SR の上界が推定できる．

A.4.1　相互情報量と攻撃成功確率の関係

まず，次の補題を導入する．

補題 3. 図 *A.1* の通信路において，次式が成り立つ．

$$\xi(\mathrm{SR}) \le I(\boldsymbol{X}^m; Z^m \mid T^m).$$

ここで，$\xi(r) = H(K) - (1-r)\log_2(2^n-1) - H_2(r)$ であり，$\mathrm{SR} = \Pr(K=\hat{K})$ は攻撃成功確率，H_2 は二値エントロピー関数（$H_2(r) = -r\log(r) - (1-r)\log(1-r)$）である．

Proof. 証明の大筋は，文献 [76] と同様のため省略する． □

後ほど命題 3 で示すとおり，補題 3 の関数 ξ は非負であり，$\xi(2^{-n}) = 0$ で最小となり，$\xi(1) = n$ で最大となる狭義凸関数である．直感的には，ξ は，確率を情報量に変換する関数である．鍵が n ビットのとき，鍵候補は 2^n 個存在するため，正解鍵が全くわからないとき，$\mathrm{SR} = \Pr(K=\hat{K}) = 2^{-n}$ である．これは秘密鍵に関する攻撃者がもつ情報量が 0 であることを意味する．実際，$\xi(2^{-n}) = 0$ である．一方で，$\mathrm{SR} = 1$ のとき，攻撃者は鍵のエントロピである n ビットをすべて知っている必要がある．この場合は，$\xi(1) = n$ である．このように関数 ξ は，攻撃成功確率から，秘密鍵に関して知っていなければならない情報量（ビット数）を得る関数であるといえる．補題 3 は，この情報量が，たかだか相互情報量 $I(\boldsymbol{X}^m; Z^m \mid T^m)$ で上から押さえられることを意味する．もし，サイドチャネルから得られる情報が $I(\boldsymbol{X}^m; Z^m \mid T^m) = 0$ であれば，$\xi(\mathrm{SR}) \le 0$ となって，当然 $\mathrm{SR} = 2^{-n}$ となる．一方で，サイドチャネル攻撃によって得られる情報量 $I(\boldsymbol{X}^m; Z^m \mid T^m) = \mathbb{E}\log q(\boldsymbol{X}^m; Z^m \mid T^m)/(q(\boldsymbol{X}^m \mid T^m)q(Z^m \mid T^m))$ は，m 次の積分を含むため[*3]，解析的に求めるのは困難である．そこで，簡単化のために次の補題を用いる．

補題 4. 相互情報量 $I(\boldsymbol{X}^m; Z^m \mid T^m)$ と，$I(\boldsymbol{X}; Z)$ の間で，$I(\boldsymbol{X}^m; Z^m \mid T^m) \le mI(\boldsymbol{X}; Z)$ が成り立つ．

[*3] 期待値の演算 $\mathbb{E}$ は，積分を含むことに注意されたい．

Proof. 相互情報量がエントロピーを用いて分解できることを用いると，

$$\begin{aligned} I(\boldsymbol{X}^m; Z^m \mid T^m) &\overset{(a)}{=} H(\boldsymbol{X}^m \mid T^m) - H(\boldsymbol{X}^m \mid Z^m, T^m) \\ &\overset{(b)}{=} H(\boldsymbol{X}^m \mid T^m) - H(\boldsymbol{X}^m \mid Z^m) \\ &= H(\boldsymbol{X}^m) - H(\boldsymbol{X}^m \mid Z^m) - (H(\boldsymbol{X}^m) - H(\boldsymbol{X}^m \mid T^m)) \\ &\overset{(c)}{=} I(\boldsymbol{X}^m; Z^m) - I(\boldsymbol{X}^m; T^m) \\ &\leq mI(\boldsymbol{X}; Z) \end{aligned}$$

である．ここで，(a) と (c) は相互情報量の定義から，(b) は $(\boldsymbol{X}^m, Z^m)$ は，T^m に依存しないことから従う． □

補題 4 から，m 波形を使用したときの相互情報量 $I(\boldsymbol{X}^m; Z^m \mid T^m)$ は，1 波形あたりに入手できる情報量 $I(\boldsymbol{X}; Z)$ に，攻撃に使用した波形数 m を乗じたもので，上から抑えられることがわかる．$I(\boldsymbol{X}^m; Z^m \mid T^m)$ の計算に比べ，$I(\boldsymbol{X}; Z)$ の推定は圧倒的に簡単であることに注意されたい．ここから，次の主張が成り立つ．

命題 2. 攻撃成功確率 SR と，相互情報量 $I(\boldsymbol{X}; Z)$ について次が成り立つ．

$$\xi(\mathrm{SR}) \leq mI(\boldsymbol{X}; Z).$$

Proof. 補題 3 と 4 より，自明． □

命題 2 より，提案通信路モデル下でも，相互情報量 $I(\boldsymbol{X}; Z)$ により SR の上界を与えられることが分かる．

A.4.2　各シェアの条件付き確率分布が既知の場合

本節では，各シェアの条件付き確率 $q(S_1|\boldsymbol{L}_1), q(S_2|\boldsymbol{L}_2), \ldots, q(S_d|\boldsymbol{L}_d)$ が既知の場合の攻撃成功確率の上界の導出を行う．そのためにまず相互情報量 $I(\boldsymbol{X}; Z)$ と，各シェアの条件付き確率分布の間の関係式を導く．具体的には次が成り立つ．

補題 5. 相互情報量 $I(\boldsymbol{X}; Z)$ について，次式が成り立つ．

$$I(\boldsymbol{X}; Z) \leq \log \left(\sum_{w} \prod_{i=1}^{d} \mathbb{E} \hat{q}^2_{S_i|\boldsymbol{L}_i}(w \mid \boldsymbol{L}_i) \right).$$

ここで，$\hat{q}_{S_i|\boldsymbol{L}_i}$ は確率分布 $q_{S_i|\boldsymbol{L}_i}$ のアダマール変換である．

Proof. 相互情報量の定義から，

$$\begin{aligned} I(\boldsymbol{X};Z) &= H(Z) - H(Z \mid \boldsymbol{X}) \\ &= \mathbb{E}\log q(Z \mid \boldsymbol{X}) + H(Z) \\ &\overset{(a)}{\leq} \log \mathbb{E} q(Z \mid \boldsymbol{X}) + n \\ &\overset{(b)}{=} \log \mathbb{E}\left[\mathbb{E}[q(Z \mid \boldsymbol{X}) \mid \boldsymbol{X}]\right] + n \\ &= \log \mathbb{E}\left[\sum_z q^2(z \mid \boldsymbol{X})\right] + n \end{aligned} \tag{A.1}$$

が成り立つ．ここで，(a) は Jensen の不等式，(b) は期待値のタワープロパティを用いた．条件付き確率 $p(z \mid \boldsymbol{X})$ が

$$\begin{aligned} q(z \mid \boldsymbol{X}) &\overset{(a)}{=} \sum_{s_1,\ldots,s_d} q(z, s_1, s_2, \ldots, s_d \mid \boldsymbol{L}_1, \ldots, \boldsymbol{L}_d) \\ &\overset{(b)}{=} \sum_{s_1,s_2,\ldots,s_d} q(z \mid s_1, s_2, \ldots, s_d) q(s_1, s_2, \ldots, s_d \mid \boldsymbol{L}_i) \\ &\overset{(c)}{=} \sum_{s_1,s_2,\ldots,s_d} q(z \mid s_1, s_2, \ldots, s_d) \prod_{i=1}^{d} q(s_i \mid \boldsymbol{L}_i) \\ &\overset{(d)}{=} \sum_{s_1 \oplus s_2 \oplus \cdots \oplus s_d = z} \prod_{i=1}^{d} q(s_i \mid \boldsymbol{L}_i) \end{aligned} \tag{A.2}$$

のように各シェアの条件付き確率の畳込みで表せることに注意する．ただし，ここで (a) は，$\boldsymbol{X} = (\boldsymbol{L}_1, \ldots, \boldsymbol{L}_d)$ であること，(b) はマルコフ連鎖 $Z \leftrightarrow \boldsymbol{S} \leftrightarrow \boldsymbol{L}$，(c) はシェア $(S_1, \boldsymbol{L}_1), \ldots, (S_d, \boldsymbol{L}_d)$ の独立性，(d) はシェアと中間値の関係

$$q(z \mid s_1, \ldots, s_d) = \begin{cases} 1 & (z = s_1 \oplus s_2 \oplus \cdots \oplus s_d) \\ 0 & (\text{otherwise}) \end{cases}$$

を利用した．式 (A.2) より，中間値と波形の間の条件付き確率は，WHT を用いて $\hat{q}(w \mid \boldsymbol{X}) = \prod_i \hat{q}(w \mid \boldsymbol{L}_i)$ のように表せる．ただし，$\hat{}$ は関数の WHT を表す．よって，式 (A.1) に対して，パーセバルの等式を適用することで，

$$I(\boldsymbol{X};Z) \leq \log \sum_w \prod_{i=1}^{d} \mathbb{E}\hat{q}^2_{S_i \mid \boldsymbol{L}_i}(w \mid \boldsymbol{L}_i).$$

□

補題 5 から，相互情報量 $I(\boldsymbol{X};Z)$ は，各シェアの確率分布の WHT によって上から押さえられることがわかる．よって，次の定理が成立する．

定理 4 (各シェアの条件付き確率が既知の場合の SR の上界)**.** 攻撃成功確率 SR と，各シェアの条件付き確率の WHT の間で

$$\xi(\mathrm{SR}) \leq m \log \sum_{w} \prod_{i=1}^{d} \mathbb{E}\hat{q}^2_{S_i|\boldsymbol{L}_i}(w \mid \boldsymbol{L}_i).$$

が成立する．

Proof. 補題 5 と，命題 2 から自明． □

定理 4 と深層学習を組み合わせた，高精度な攻撃成功確率推定手法は，次章で述べることとし，次節では，まず各シェアの相互情報量を用いた不等式を導く．

A.4.3　各シェアの相互情報量が既知の場合

本節では，確率分布の WHT を各シェアの相互情報量を用いて抑える方法を述べる．これにより，中間値と波形との間の相互情報量 $I(Z;\boldsymbol{X})$ は，各シェアの相互情報量 $I(S_i;\boldsymbol{L}_i)$ の積によって上から抑えられることがわかる．具体的には確率分布の WHT と各シェアの相互情報量の間には次の関係が成り立つ．

補題 6. $d \in \mathbb{N}$ をシェアの数とし，$\boldsymbol{L}_1, \boldsymbol{L}_2, \ldots, \boldsymbol{L}_d$ を各シェアに関する漏洩情報とする．各シェアの確率分布 $q_{S_1|\boldsymbol{L}_1}, q_{S_2|\boldsymbol{L}_2}, \ldots, q_{S_d|\boldsymbol{L}_d}$ の WHT をそれぞれ $\hat{q}_{S_1|\boldsymbol{L}_1}$, $\hat{q}_{S_2|\boldsymbol{L}_2}, \ldots, \hat{q}_{S_d|\boldsymbol{L}_d}$ とする．すべての $i \in \{1, 2, \ldots, d\}$ について，$\hat{q}^2_{S_i|\boldsymbol{L}_i}(0 \mid \boldsymbol{L}_i) = 1$ と，$\forall w \in \{1, 2, \ldots, 2^n - 1\}, \hat{q}^2_{S_i|\boldsymbol{L}_i}(w \mid \boldsymbol{L}_i) \leq 2\ln(2) I(S_i \mid \boldsymbol{L}_i)$ が成り立つ．ここで，ln は自然対数である．

Proof. WHT の定義より，

$$\hat{q}_{S_i|\boldsymbol{L}_i}(w \mid \boldsymbol{L}_i) = \sum_{s \in \mathcal{S}} q_{S_i|\boldsymbol{L}_i}(s \mid \boldsymbol{L}_i)(-1)^{\langle w \cdot s \rangle}$$

が成り立つ．ここで，$\langle w \cdot s \rangle$ は w と s をそれぞれ $\mathbb{F}_2^n$ の元とみなして各要素の積和を行う演算を表す．まず，$w = 0$ のときは，

$$\hat{q}_{S_i|\boldsymbol{L}_i}(0 \mid \boldsymbol{L}_i) = \sum_{s \in \mathcal{S}} q_{S_i|\boldsymbol{L}_i}(s \mid \boldsymbol{L}_i) = 1.$$

次に $w \neq 0$ のときを考える．まず，指示関数 $\mathbb{1}$ を用いて，WHT を

$$\hat{q}_{S_i|\boldsymbol{L}_i}(w \mid \boldsymbol{L}_i) = \sum_{s \in \mathcal{S}} q_{S_i|\boldsymbol{L}_i}(s \mid \boldsymbol{L}_i)\mathbb{1}_{\{\langle w \cdot s \rangle = 0\}} - \sum_{s \in \mathcal{S}} q_{S_i|\boldsymbol{L}_i}(s \mid \boldsymbol{L}_i)\mathbb{1}_{\{\langle w \cdot s \rangle = 1\}}$$

と分解する．ここで，新たに確率変数 $Y_w = \langle w \cdot S_i \rangle$ を定義すれば，

$$\begin{aligned}\hat{q}_{S_i|\boldsymbol{L}_i}(w \mid \boldsymbol{L}_i) &= \mathbb{E}\left[\mathbb{1}_{\{\langle w \cdot S_i \rangle = 0\}} \mid \boldsymbol{L}_i\right] - \mathbb{E}\left[\mathbb{1}_{\{\langle w \cdot S_i \rangle = 1\}} \mid \boldsymbol{L}_i\right] \\ &= q_{Y_w|\boldsymbol{L}_i}(0 \mid \boldsymbol{L}_i) - q_{Y_w|\boldsymbol{L}_i}(1 \mid \boldsymbol{L}_i)\end{aligned}$$

となる．任意の $w \neq 0$ について，$\langle w \cdot s \rangle = 1$ をみたす s の個数は，集合 $\mathcal{S}$ の半分であるため，$q_{Y_w}(0) = q_{Y_w}(1) = 1/2$ が成り立つことに注意されたい．ここから，

$$\begin{aligned}|\hat{q}_{S_i|\boldsymbol{L}_i}(w \mid \boldsymbol{L}_i)| &= |q_{Y_w|\boldsymbol{L}_i}(0 \mid \boldsymbol{L}_i) - q_{Y_w|\boldsymbol{L}_i}(1 \mid \boldsymbol{L}_i)| \\ &= \sum_y |q_{Y_w|\boldsymbol{L}_i}(y \mid \boldsymbol{L}_i) - q_{Y_w}(y)| \\ &\leq \sqrt{2\ln(2) D_{\mathrm{KL}}(q_{Y_w|\boldsymbol{L}_i} \parallel q_{Y_w})}.\end{aligned}$$

ここで Pinsker の不等式を使用した．ただし，ln は自然対数，D_{KL} は Kullback-Leibler（KL）ダイバージェンスを表す．ここで両辺を自乗して，期待値をとることで，

$$\mathbb{E}\hat{q}^2_{S_i|\boldsymbol{L}_i}(w \mid \boldsymbol{L}_i) \leq 2\ln(2) D_{\mathrm{KL}}(q_{Y_w,\boldsymbol{L}_i} \parallel q_{Y_w} q_{\boldsymbol{L}_i}) = 2\ln(2) I(Y_w; \boldsymbol{L}_i)$$

を得る．確率変数 Y_w が Z の関数とみなせることから，データ処理不等式から $I(Y_w; \boldsymbol{L}_i) \leq I(S_i; \boldsymbol{L}_i)$ が成立する．以上より，

$$\mathbb{E}\hat{q}^2_{S_i|\boldsymbol{L}_i}(w \mid \boldsymbol{L}_i) \leq 2\ln(2) I(S_i; \boldsymbol{L}_i).$$

□

これらの補題から次の定理が導かれる．

定理 5 (各シェアの相互情報量が既知の場合)**.** 攻撃成功確率 SR と各シェアの相互情報量には

$$\xi(\mathrm{SR}) \leq m \log\left((2^n - 1)(2\ln(2))^d \prod_{i=1}^{d} I(S_i; \boldsymbol{L}_i) + 1\right)$$

の関係が成り立つ．

Proof. 相互情報量の間の関係式については，定理 4 と補題 6 から，

$$\begin{aligned}\xi(\mathrm{SR}) &\leq \log\left(\sum_{w \neq 0} \prod_{i=1}^{d} 2\ln(2) I(S_i; \boldsymbol{L}_i) + 1\right) \\ &= \log\left((2^n - 1)(2\ln(2))^d \prod_{i=1}^{d} I(S_i; \boldsymbol{L}_i) + 1\right)\end{aligned}$$

と主張は示される．□

A.4.4　マスキングの次数と攻撃成功確率の関係

定理 5 から，SR と相互情報量の間に関係性があることがわかる．マスキング対策では，各シェアの相互情報量が小さいときに，攻撃成功確率が指数的に小さくなることが Duc らによって示されている．同様の結果は定理 5 から次のように得られる[*4]．

命題 3. ある実数 $0 < \epsilon < 1$ が存在し，シェアの相互情報量の最大値に対して $\max_i I(S_i; \boldsymbol{L}_i) < \epsilon/(2\ln(2))$ が成立するとする．このとき任意の整数 $n, m \in \mathbb{N}$ について，$\mathrm{SR}_d - 1/2^n = O(\epsilon^{d/2})$ $(d \to \infty)$ が成り立つ．ただし，SR_d はシェア数 d のときの攻撃成功確率を表す．

Proof. 連続関数 $\xi : [0,1] \to \mathbb{R}_+$ を $\xi(r) \coloneqq n - (1-r)\log(2^n - 1) - H_2(r)$ と定義する．関数 ξ は，開区間 $(0,1)$ で C^2 級関数であることに注意されたい．証明は以下に示す 3 ステップから成る．

■**関数 ξ は狭義凸関数**:　$(1-r)\log(2^n-1)$ は r に関する 1 次関数であるため，凸関数であり，また $-H_2(r)$ は狭義凸関数である．よって，それらの和である $(1-r)\log(2^n - 1) - H_2(r)$ は狭義凸関数であり，それらと定数だけ異なる ξ も狭義凸関数である．狭義凸関数は，極小値が存在するとき，極小値が最小値であり，かつただ一点に限られる．ξ の極小値を取る点 r_0 は，

$$\frac{\partial \xi(r_0)}{\partial r} = \log(r_0) - \log(1 - r_0) + \log(2^n - 1) = 0$$

より，$r_0 = 1/2^n \in (0,1)$ である．また $\xi(r_0) = 0$ が成り立つ．

■**ξ の下界の導出**:　整数 $d_0 > 0$ を

$$\min\{\xi(0), \xi(1)\} > m\log(e)(2^n - 1)\epsilon^{d_0}$$

を満たすように取る．任意の整数 $d \geq d_0$ について，半開区間 $\mathcal{I}_d = \{\, j \in [0,\infty) \mid j < m\log(e)(2^n-1)\epsilon^d \,\}$ を定義する．また，集合 $\mathcal{U}_d = \xi^{-1}(\mathcal{I}_d)$ とおく．定義から常に $\mathcal{I}_d \subset \mathcal{I}_{d_0}$ であるため，$\mathcal{U}_d \subset \mathcal{U}_{d_0}$ である．このとき，$\mathcal{U}_d$ は点 r_0 を含み，また ξ の凸性から開区間であることに注意する．また，$\mathcal{U}_d \subset (0,1)$ であるため，関数 ξ は $\mathcal{U}_d$ で C^2 級関数である．したがって，テイラーの定理から，$r \in \mathcal{U}_d$ に対して，ある実数 $c \in \mathcal{U}_d \subset \mathcal{U}_{d_0}$

[*4] これは定理 5 が単に SR を相互情報量の積で押さえているのであれば自明だが，実際には関数 ξ が存在するため自明ではない．

が存在し，

$$\xi(r) = \frac{(r-r_0)^2}{2}\frac{\partial^2\xi(c)}{\partial r^2} \geq \frac{(r-r_0)^2}{2}\inf_{r'\in\mathcal{U}_{d_0}}\frac{\partial^2\xi(r')}{\partial r^2} \tag{A.3}$$

が成り立つ．ここで，右辺の $\inf_{r'\in\mathcal{U}_{d_0}}\partial^2\xi(r')/\partial r^2$ は r と d に依存せず，また ξ の強凸性から，$\partial^2\xi/\partial r^2$ は下に有界で，正の実数となることに注意されたい．そこで，$\xi_c = \inf_{r'\in\mathcal{U}_{d_0}}\frac{\partial^2\xi(r')}{\partial r^2}$ とおく．

■主張の証明:　シェア数が d のときの攻撃成功確率を SR_d とおくと，定理 5 と $\ln(1+x) \leq x$ より，すべての $d > d_0$ について，

$$\xi(\mathrm{SR}_d) < m\log\left((2^n-1)\epsilon^d + 1\right) < m\log(e)(2^n-1)\epsilon^d$$

とできる．ここから，$\mathrm{SR}_d \in \mathcal{U}_d$ より，式 (A.3) から

$$\frac{(\mathrm{SR}_d - 1/2^n)^2}{2}\xi_c < m\log(e)(2^n-1)\epsilon^d$$

を得る．これを整理することで

$$|\mathrm{SR}_d - 1/2^n| < \sqrt{\frac{2m\log(e)(2^n-1)}{\xi_c}}\epsilon^{d/2}$$

である．$\sqrt{2m\log(e)(2^n-1)/\xi_c}$ は，d によらない定数である．したがって，$\mathrm{SR}_d - 1/2^n = O(\epsilon^{d/2})$ $(d\to\infty)$ である．□

上の命題から，各シェアの相互情報量が十分に小さい時，攻撃成功確率 $\mathrm{SR} = \Pr(\hat{K} \neq K)$ は $1/2^n$ へ収束することがわかる．

また，この結果は先行研究 [117] の結果よりも弱い条件で成立する．文献 [117] では，同様の結果が成立するためには，シェアの相互情報量の最大値が $\max_i I(S_i;\boldsymbol{L}_i) \leq 2^{-2n+1}$ を満たさなければならなかった．例えば AES に対する攻撃 $(n=8)$ の場合，$2^{-15} \approx 3.05\times10^{-5}$ であるため，先行研究の手法 [117] は漏洩強度が極めて弱いとき（例えば波形取得の SNR が極めて悪いとき）でのみ有効であり，現実のデバイスや計測装置に適用できない可能性がある．一方で，本章で示した命題 3 は，$\max_i I(S_i;\boldsymbol{L}_i) < 1/(2\ln(2)) \approx 0.721$ を満たせば，SR がマスキングの次数に対して指数的に減少することを保証でき，先行研究に対してより強力かつ一般化された結果と言える．

A.5　深層学習を用いた条件付き確率推定による SR の上界評価の高精度化

前章までは，(1) 各シェアの相互情報量が既知の場合と，(2) 条件付き確率が既知の場合のそれぞれに対応する SR の上界の導出を行った．条件 (1) の上界を利用するため

には，各シェアの相互情報量を推定を行う必要がある．これは，すでに様々な先行研究 [119, 120, 76] で推定手法が報告されている．一方で，条件 (2) の上界を利用するために必要な，条件付き確率を推定する手法は，これまでほとんど検討されてこなかった．そこで，本節では，各シェアに関する条件付き確率を深層学習を用いて推定する手法を提案する．

定理 4 は，各シェアに関する確率分布の WHT の期待値を用いて，SR の上界が

$$\xi(\mathrm{SR}) \leq \log \left(\sum_{w} \prod_{i=1}^{d} \mathbb{E}\hat{q}^2_{S_i|\boldsymbol{L}_i}(w \mid \boldsymbol{L}_i) \right)$$

と表せることを主張している．ここで，各シェアの漏洩強度が等しいと仮定する．すなわち，$\mathbb{E}\hat{q}^2_{S_i|\boldsymbol{L}_i}$ が i によらずに同一であるとし，以降 i を省略して単に $\mathbb{E}\hat{q}^2_{S|\boldsymbol{L}}$ と表記する．このとき，相互情報量の上界は

$$I(\boldsymbol{X}; Z) \leq \log \sum_{w} \left(\mathbb{E}\hat{q}^2_{S|\boldsymbol{L}}(w \mid \boldsymbol{L}) \right)^d$$

となる．$\hat{q}_{S|\boldsymbol{L}}$ は，確率分布 $q_{S|\boldsymbol{L}}$ のアダマール変換なので，高精度に $q_{S|\boldsymbol{L}}$ を推定できれば，近似的に上界を計算できる．

本節では，上記の近似に深層学習（ニューラルネットワーク）を使用することを考える．ニューラルネットワークが表現する分布を $p(s \mid \boldsymbol{L}; \theta)$ とする．ここで，θ はモデルのパラメータを表す．深層学習による多クラス分類において，一般的に用いられる負の対数尤度[*5]は，漸近的に次式で定義されるクロスエントロピー

$$\mathrm{CE}(q, p) = -\mathbb{E} \log p(S \mid \boldsymbol{L}; \theta) = -\int \sum_{s} q(s, \boldsymbol{l}) \log p(s \mid \boldsymbol{l}; \theta) d\boldsymbol{l}$$

に一致することが知られている．ここで，$\mathrm{CE}(q, p)$ は最小のとき，$q = p$ となる．したがって，深層学習により推定した $\boldsymbol{L}$ に関する S の条件付き分布 p を用いて近似的に確率分布 q を模倣することができる．ここで，学習済みモデルパラメータを $\hat{\theta}$ とすると，相互情報量の上界は

$$I(\boldsymbol{X}; Z) \lessapprox \log \sum_{w} (\mathbb{E}\hat{p}^2_{S|\boldsymbol{L}}(w \mid \boldsymbol{L}; \hat{\theta}))^d$$

と近似される．各シェアの漏洩強度が等しいという仮定から，あるシェアについて，$\mathbb{E}\hat{p}^2_{S|\boldsymbol{L}}(w \mid \boldsymbol{L}; \hat{\theta})$ を計算できれば任意のシェア数 d に対する相互情報量が計算できる．

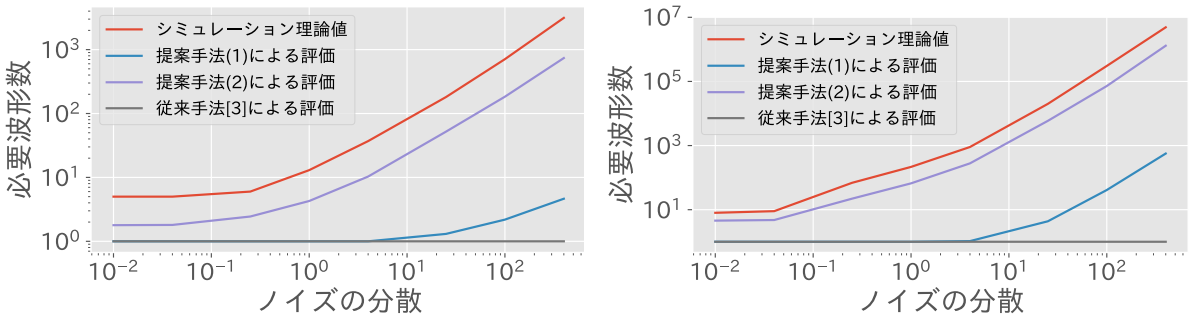
[*5] カテゴリカルクロスエントロピー損失関数とも呼ばれる．

A.6　実験

本節では，提案する 2 つの上界による SR および必要波形数推定の精度を評価する．定理 5 から導出される相互情報量から得られるものを提案手法（1），第 A.5 章で提案した深層学習から推定される条件付き確率から得られるものを提案手法（2）とする．　提案手法（2）においてニューラルネットワークの学習を実行して評価を行う必要があるため，ここではシミュレーションデータを生成して調査した．ここでは，第 1 ラウンドの AES の S-box の出力 $\mathrm{Sbox}(k^* \oplus T)$ を秘密情報 Z とし，それがシェアの和 $Z = S_1 \oplus \cdots \oplus S_d$ として分解されるとする．ただし，k^* は秘密鍵である．シェアの数は $d = 1, 2$ の 2 種類を対象とし，各シェアは，ハミング重みをとった上で，ガウスノイズが付加された状態で漏洩するとした．すなわち，$L = \mathrm{HW}(S) + N$ であり，ここで N はガウスノイズを表す．ガウス型通信路の相互情報量の上界は，シャノン=ハートレーの定理から $\log(1+\mathrm{SNR})/2$ であることが知られている．したがって，ガウスノイズの量を増やして SNR を減少させることで，シェアの相互情報量 $I(S;L)$ を調整することができる．学習に使用したネットワークは，4 層の全結合層からなるモデルであり，入力側から出力次元を 128, 256, 128, 256 とした．出力層の活性化関数は Softmax 関数であり，それ以外の層ではすべて ELU とした．オプティマイザには Adam を使用し，損失関数には負の対数尤度を用いた．学習およびテストに使用したデータ数はそれぞれ 500 万である．また，上界（1）による評価のため，各シェアの相互情報量 $I(S;L)$ の推定も行った．推定には，python のオープンソースライブラリである NPEET を使用して推定を行った[*6]．推定に使用したデータ数は 1,000 万である．

図 A.2(a) と A.2(b) に，それぞれ $d = 1$ と $d = 2$ の場合の結果を示す．ここでは，必要波形数の下界として与えられる．図中の理論値は漏洩の分布が既知の状態で，500 回テンプレート攻撃を実行したときの SR を示している．漏洩の分布が既知であることから，このテンプレート攻撃は理論上最も強力な攻撃となる [121]．ここで，この理論値はシミュレーションでのみ計算可能であり，実験的には計算不可能であることに注意されたい．また，比較のため文献 [117] から得られる結果も示す．なお，文献 [76, 119] の手法はマスキング対策への適用が不可能のため省略する．結果から，提案手法（1）および（2）による推定がいずれも先行研究よりもタイトであることを確認できる．また，提案手法（2）のほうが提案手法（1）の結果よりも強力なことが確認できる．これは，定理 4 が各シェアの条件付き確率を直接利用しており，定理 5 よりも多くの情報を用いているためである．ただし，定理 4 を利用するためには，推定が相互情報量よりも難しい条件付き確率の

[*6] https://github.com/gregversteeg/NPEET



(a) $d = 1$, SR $= 0.80$ での必要波形数の評価　　(b) $d = 2$, SR $= 0.80$ での必要波形数の評価

図 A.2: 攻撃が成功するために必要な波形数の評価

高精度な推定が可能な場合に限定されることに注意されたい．実際の暗号モジュールに対する適用可能性の調査は今後の課題である．

A.7　結び

本章では，サイドチャネル攻撃を通信路としてみなすことで，マスキング対策された暗号実装に対するサイドチャネル攻撃の攻撃成功確率の上界を導出した．導出した上界を使用することで，一つのシェアあたりの相互情報量もしくは確率分布から，任意のシェア数によるマスキング対策実装への攻撃難易度を評価できる．また，実験的に本稿で述べた解析の妥当性を評価した．

参考文献

[1] E. Rescorla. The Transport Layer Security (TLS) protocol version 1.3. Internet Engineering Task Force (IETF), RFC 8446, October 2018. https://datatracker.ietf.org/doc/rfc8446/.

[2] 情報処理推進機構. 耐タンパー性調査研究委員会報告書. https://www.ipa.go.jp/security/enc/CRYPTREC/fy15/documents/INSTAC_rep.pdf, 2003.

[3] R. Anderson, M. Bond, J. Clulow, and S. Skorobogatov. Cryptographic processors-a survey. *Proceedings of the IEEE*, 94(2):357–369, 2006.

[4] Takeshi Sugawara. 3-share threshold implementation of AES S-box without fresh randomness. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2019:123–145, 2019.

[5] Samaneh Ghandali, Georg T. Becker, Daniel E. Holcomb, and Christof Paar. A design methodology for stealthy parametric trojans and its application to bug attacks. In *CHES*, pages 625–647. Springer, 2016.

[6] David Knichel, Pascal Sasdrich, and Amir Moradi. Silver – statistical independence and leakage verification. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2020*, pages 787–816, Cham, 2020. Springer International Publishing.

[7] Jinpeng Lv, Priyank Kalla, and Florian Enescu. Efficient Gröbner Basis Reductions for Formal Verification of Galois Field Arithmetic Circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 32(9):1409–1420, September 2013.

[8] IPA Information-technology Promotion Agency, Japan : IPA/ISEC : JCMVP(Japan Cryptographic Module Validation Program). https://www.ipa.go.jp/security/english/jcmvp.html.

[9] Ryad Benadjila, Emmanuel Prouff, Rémi Strullu, Eleonora Cagli, and Cécile Dumas. Deep learning for side-channel analysis and introduction to ASCAD database. *Journal of Cryptographic Engineering*, 10(2):163–188, June 2020.

[10] Shin-ichi Minato. Zero-suppressed BDDs and their applications. *International Journal on Software Tools for Technology Transfer*, 3(2):156–170, May 2001.

[11] OECD Guidelines for the Security of Information Systems and Networks: Towards a Culture of Security - OECD. `https://www.oecd.org/sti/ieconomy/oecdguidelinesforthesecurityofinformationsystemsandnetworkstowardsacultureofsecurity.htm`.

[12] Jonathan Katz and Yehuda Lindell. *Introduction to modern cryptography*. CRC press, 2020.

[13] Jeffrey Hoffstein, Jill Pipher, Joseph H Silverman, and Joseph H Silverman. *An introduction to mathematical cryptography*, volume 1. Springer, 2008.

[14] Information Technology Laboratory Computer Security Division. Cryptographic Module Validation Program — CSRC — CSRC. https://csrc.nist.gov/projects/cryptographic-module-validation-program, October 2016.

[15] ISO/IEC 19790:2012. `https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/05/29/52906.html`.

[16] ISO/IEC 24759:2017. `https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/07/25/72515.html`.

[17] ISO/IEC 18367:2016. `https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/06/22/62286.html`.

[18] Eli Biham, Yaniv Carmeli, and Adi Shamir. Bug attacks. *J. Cryptology*, 29:775–805, 2016.

[19] Maik Ender, Samaneh Ghandali, Amir Moradi, and Christof Paar. The first thorough side-channel hardware trojan. In *ASIACRYPT (1)*, pages 755–780. Springer, 2017.

[20] ISO/IEC 17825:2016. `https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/06/06/60612.html`.

[21] Thorben Moos, Felix Wegener, and Amir Moradi. Dl-la: Deep learning leakage assessment (long paper) - a modern roadmap for sca evaluations. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2021.

[22] Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power analysis attacks: Revealing the secrets of smart cards*, volume 31. Springer Science & Business Media, 2008.

[23] K. Tiri, M. Akmal, and I. Verbauwhede. A dynamic and differential cmos logic with signal independent power consumption to withstand differential power

analysis on smart cards. In *Proceedings of the 28th European Solid-State Circuits Conference*, pages 403–406, 2002.

[24] Daisuke Suzuki, Minoru Saeki, and Tetsuya Ichikawa. Random switching logic: A new countermeasure against DPA and second-order DPA at the logic level. *IEICE transactions on fundamentals of electronics, communications and computer sciences*, 90(1):160–168, 2007.

[25] Zhimin Chen and Yujie Zhou. Dual-rail random switching logic: A countermeasure to reduce side channel leakage. In *Cryptographic Hardware and Embedded Systems - CHES 2006, 8th International Workshop*, volume 4249 of *Lecture Notes in Computer Science*, pages 242–254. Springer, 2006.

[26] Marco Bucci, Luca Giancane, Raimondo Luzzi, and Alessandro Trifiletti. Three-phase dual-rail pre-charge logic. In *Cryptographic Hardware and Embedded Systems - CHES 2006, 8th International Workshop*, volume 4249 of *Lecture Notes in Computer Science*, pages 232–241. Springer, 2006.

[27] Thomas Popp and Stefan Mangard. Masked dual-rail pre-charge logic: Dpa-resistance without routing constraints. In *Cryptographic Hardware and Embedded Systems - CHES 2005, 7th International Workshop, Edinburgh, UK, August 29 - September 1, 2005, Proceedings*, volume 3659 of *Lecture Notes in Computer Science*, pages 172–186. Springer, 2005.

[28] Alexandre Duc, Sebastian Faust, and François-Xavier Standaert. Making masking security proofs concrete (or how to evaluate the security of any leaking device). *Journal of Cryptology*, 32:1263–1297, 2019.

[29] Amir Moradi, Axel Poschmann, San Ling, Christof Paar, and Huaxiong Wang. Pushing the limits: A very compact and a threshold implementation of AES. In *Advances in Cryptology—EUROCRYPT 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 59–88. Springer, 2011.

[30] Thomas De Cnudde, Begül Bilgin, Oscar Reparaz, Ventzislav Nikov, and Svetla Nikova. Higher-order threshold implementation of the AES S-box. In *International Conference on Smart Card Research and Advanced Applications*, volume 9514 of *Lecture Notes in Computer Science*, pages 259–272, 2015.

[31] Begül Bilgin, Benedikt Gierlichs, Svetla Nikova, Ventzislav Nikov, and Vincent Rijmen. Trade-offs for threshold implementations illustrated on AES. *IEEE Transactions on Computer-Aided Design of Integrated and Systems*, 34(7):1188–1200, 2015.

[32] Thomas De Cnudde, Oscar Reparaz, Begül Bilgin, Svetla Nikova, Ventzislav

Nikov, and Vincent Rijmen. Masking AES with $d + 1$ shares in hardware. In *International Conference on Cryptographic Hardware and Embedded Systems*, volume 9813 of *Lecture Notes in Computer Science*, pages 194–212. Springer, 2016.

[33] Rei Ueno, Naofumi Homma, and Takafumi Aoki. Toward more efficient DPA-resistant AES hardware architecture based on threshold implementation. In *International Workshop on Constructive Side-Channel Analysis and Secure Design*, volume 10348 of *Lecture Notes in Computer Science*, pages 50–64, 2017.

[34] Hannes Gross, Stefan Mangard, and Thomas Korak. Domain-oriented masking: Compact masked hardware implementations with arbitrary protection order. In *ACM Workshop on Theory of Implementation Security*, page 3, 2016.

[35] Felix Wegener and Amir Moradi. A first-order SCA resistant AES without fresh randomness. In *International Workshop on Constructive Side-Channel Analysis and Secure Design*, volume 10815 of *Lecture Notes in Computer Science*, pages 245–262, 2018.

[36] Aein Rezaei Shahmirzadi and Amir Moradi. Re-consolidating first-order masking schemes—nullifying fresh randomness. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2019:123–145, 2019.

[37] Zdenek Martinasek, Jan Hajny, and Lukas Malina. Optimization of power analysis using neural network. In Aurélien Francillon and Pankaj Rohatgi, editors, *Smart Card Research and Advanced Applications*, pages 94–107, Cham, 2014. Springer International Publishing.

[38] Eleonora Cagli, Cécile Dumas, and Emmanuel Prouff. Convolutional neural networks with data augmentation against jitter-based countermeasures. In *Cryptographic Hardware and Embedded Systems – CHES 2017*, volume 10529 of *Lecture Notes in Computer Science*, pages 45–68. Springer, 2017.

[39] Gabriel Zaid, Lilian Bossuet, Amaury Habrard, and Alexandre Venelli. Methodology for efficient cnn architectures in profiling attacks. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2020, Issue 1:1–36, 2019.

[40] Benjamin Hettwer, Tobias Horn, Stefan Gehrer, and Tim Güneysu. Encoding power traces as images for efficient side-channel analysis. In *2020 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pages 46–56, 2020.

[41] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.

[42] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning — Data Mining, Inference, and Prediction*. Springer, second edition, 2009.

[43] Shivam Bhasin, Jean-Luc Danger, Sylvain Guilley, Xuan Thuy Ngo, and Laurent Sauvage. Hardware Trojan Horses in Cryptographic IP Cores. In *Workshop on Fault Diagnosis and Tolerance in Cryptography*, pages 15–29. IEEE, August 2013.

[44] Sivappriya Manivannan, N.Nalla Anandakumar, and M.Nirmala Devi. Key Retrieval from AES Architecture Through Hardware Trojan Horse. In Sabu M. Thampi, Sanjay Madria, Guojun Wang, Danda B. Rawat, and Jose M. Alcaraz Calero, editors, *International Symposium on Security in Computing and Communication*, Communications in Computer and Information Science, pages 483–494. Springer Singapore, 2019.

[45] Eli Biham, Yaniv Carmeli, and Adi Shamir. Bug Attacks. In David Wagner, editor, *Advances in Cryptology – CRYPTO 2008*, Lecture Notes in Computer Science, pages 221–240. Springer Berlin Heidelberg, 2008.

[46] Randal E. Bryant. Graph-Based Algorithms for Boolean Function Manipulation. *IEEE Transactions on Computers*, C-35(8):677–691, August 1986.

[47] Randal E. Bryant and Yirng-An Chen. Verification of Arithmetic Circuits with Binary Moment Diagrams. In *Design Automation Conference*, pages 535–541, June 1995.

[48] Randal E. Bryant. On the Complexity of VLSI Implementations and Graph Representations of Boolean Functions with Application to Integer Multiplication. *IEEE Transactions on Computers*, (2):205–213, 1991.

[49] Bernd Becker, Rolf Drechsler, and Ralph Werchner. On the relation between BDDs and FDDs. In Ricardo Baeza-Yates, Eric Goles, and Patricio V. Poblete, editors, *LATIN '95: Theoretical Informatics*, Lecture Notes in Computer Science, pages 72–83, Berlin, Heidelberg, 1995. Springer.

[50] Naofumi Homma, Kazuya Saito, and Takafumi Aoki. A Formal Approach to Designing Cryptographic Processors Based on $GF(2^m)$ Arithmetic Circuits. *IEEE Transactions on Information Forensics and Security*, 7(1):3–13, February 2012.

[51] Naofumi Homma, Kazuya Saito, and Takafumi Aoki. Toward Formal Design of Practical Cryptographic Hardware Based on Galois Field Arithmetic. *IEEE Transactions on Computers*, 63(10):2604–2613, October 2014.

[52] Rei Ueno, Naofumi Homma, Yukihiro Sugawara, and Takafumi Aoki. Formal Approach for Verifying Galois Field Arithmetic Circuits of Higher Degrees.

IEEE Transactions on Computers, 66(3):431–442, March 2017.

[53] Utkarsh Gupta, Priyank Kalla, and Vikas Rao. Boolean Gröbner Basis Reductions on Finite Field Datapath Circuits Using the Unate Cube Set Algebra. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 38(3):576–588, March 2019.

[54] Tim Pruss, Priyank Kalla, and Florian Enescu. Efficient Symbolic Computation for Word-Level Abstraction From Combinational Circuits for Verification Over Finite Fields. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 35(7):1206–1218, July 2016.

[55] Cunxi Yu and Maciej Ciesielski. Formal Analysis of Galois Field Arithmetic Circuits-Parallel Verification and Reverse Engineering. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 38(2):354–365, February 2019.

[56] Michael Brickenstein and Alexander Dreyer. PolyBoRi: A framework for Gröbner-basis computations with Boolean polynomials. *Journal of Symbolic Computation*, 44(9):1326–1345, 2009.

[57] Erkay Savas and Cetin Kaya Koc. Finite field arithmetic for cryptography. *IEEE Circuits and Systems Magazine*, 10(2):40–56, Secondquarter 2010.

[58] Iwan Duursma and Hyang-Sook Lee. Tate pairing implementation for hyperelliptic curves $y^2 = x^p - x + d$. In Chi-Sung Laih, editor, *Advances in Cryptology - ASIACRYPT 2003*, Lecture Notes in Computer Science, pages 111–123, Berlin, Heidelberg, 2003. Springer.

[59] Paulo S. L. M. Barreto, Ben Lynn, and Michael Scott. Efficient Implementation of Pairing-Based Cryptosystems. *Journal of Cryptology*, 17(4):321–334, September 2004.

[60] Daniel J. Bernstein, Niels Duif, Tanja Lange, Peter Schwabe, and Bo-Yin Yang. High-speed high-security signatures. *Journal of Cryptographic Engineering*, 2(2):77–89, September 2012.

[61] Dan Boneh, Ben Lynn, and Hovav Shacham. Short Signatures from the Weil Pairing. *Journal of Cryptology*, 17(4):297–319, September 2004.

[62] Iwan Duursma and Kouichi Sakurai. Efficient algorithms for the jacobian variety of hyperelliptic curves $y^2 = x^p - x + 1$ over a finite field of odd characteristic p. In Johannes Buchmann, Tom Høholdt, Henning Stichtenoth, and Horacio Tapia-Recillas, editors, *Coding Theory, Cryptography and Related Areas*, pages 73–89, Berlin, Heidelberg, 2000. Springer.

[63] Eunjeong Lee, Hyang-Sook Lee, and Yoonjin Lee. Eta pairing computation on general divisors over hyperelliptic curves $y^2 = x^p - x + d$. *Journal of Symbolic Computation*, 43(6):452–474, June 2008.

[64] Randal Bryant and Yirng-an Chen. Verification of Arithmetic Functions with Binary Moment Diagrams. *Technical Report CMUCS*, May 1994.

[65] Yukihiro Sugawara, Rei Ueno, Naofumi Homma, and Takafumi Aoki. System for Automatic Generation of Parallel Multipliers over Galois Fields. In *International Symposium on Multiple-Valued Logic*, pages 54–59. IEEE, May 2015.

[66] Rei Ueno, Naofumi Homma, Takafumi Aoki, and Sumio Morioka. Hierarchical Formal Verification Combining Algebraic Transformation with PPRM Expansion and Its Application to Masked Cryptographic Processors. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E100.A(7):1396–1408, 2017.

[67] Rei Ueno, Naofumi Homma, and Takafumi Aoki. Automatic Generation System for Multiple-Valued Galois-Field Parallel Multipliers. *IEICE Transactions on Information and Systems*, E100-D(8):1603–1610, August 2017.

[68] Samaneh Ghandali, Georg T. Becker, Daniel Holcomb, and Christof Paar. A Design Methodology for Stealthy Parametric Trojans and Its Application to Bug Attacks. In Benedikt Gierlichs and Axel Y. Poschmann, editors, *Cryptographic Hardware and Embedded Systems – CHES 2016*, Lecture Notes in Computer Science, pages 625–647, Berlin, Heidelberg, 2016. Springer.

[69] Ralph Werchner, Thilo Harich, Rolf Drechsler, and Bernd Becker. Satisfiability Problems for OFDDs. In Tsutomu Sasao and Masahiro Fujita, editors, *Representations of Discrete Functions*, pages 233–248. Springer US, Boston, MA, 1996.

[70] Hassan Salmani. COTD: Reference-Free Hardware Trojan Detection and Recovery Based on Controllability and Observability in Gate-Level Netlist. *IEEE Transactions on Information Forensics and Security*, 12(2):338–350, February 2017.

[71] Stjepan Picek, Annelie Heuser, Alan Jovic, Shivam Bhasin, and Francesco Regazzoni. The curse of class imbalance and conflicting metrics with machine learning for side-channel evaluations. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2019, Issue 1:209–237, 2019.

[72] Jaehun Kim, Stjepan Picek, Annelie Heuser, Shivam Bhasin, and Alan Hanjalic. Make some noise. unleashing the power of convolutional neural networks for

profiled side-channel analysis. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2019, Issue 3:148–179, 2019.

[73] Jiajia Zhang, Mengce Zheng, Jiehui Nan, Honggang Hu, and Nenghai Yu. A novel evaluation metric for deep learning-based side channel analysis and its extended application to imbalanced data. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2020, Issue 3:73–96, 2020.

[74] Adrian Thillard, Emmanuel Prouff, and Thomas Roche. Success through confidence: Evaluating the effectiveness of a side-channel attack. In *CHES*, pages 21–36. Springer, 2013.

[75] Victor Lomneacute;, Emmanuel Prouff, Matthieu Rivain, Thomas Roche, and Adrian Thillard. How to estimate the success rate of higher-order side-channel attacks. In *CHES*, pages 35–54. Springer, 2014.

[76] Eloi de Chérisey, Sylvain Guilley, Olivier Rioul, and Pablo Piantanida. Best information is most successful. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2019, Issue 2:49–79, 2019.

[77] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: Synthetic minority over-sampling technique. *J. Artif. Int. Res.*, 16(1):321–357, June 2002.

[78] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[79] Takuya Shimada, Shoichiro Yamaguchi, Kohei Hayashi, and Sosuke Kobayashi. Data interpolating prediction: Alternative interpretation of mixup. In *2nd Learning from Limited Labeled Data Workshop*, 2019.

[80] Jean-Sébastien Coron and Ilya Kizhvatov. An efficient method for random delay generation in embedded software. In *Cryptographic Hardware and Embedded Systems - CHES 2009, 11th International Workshop, Lausanne, Switzerland, September 6-9, 2009, Proceedings*, volume 5747 of *Lecture Notes in Computer Science*, pages 156–170. Springer, 2009.

[81] Lennert Wouters, Victor Arribas, Benedikt Gierlichs, and Bart Preneel. Revisiting a methodology for efficient cnn architectures in profiling attacks. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2020, Issue 3:147–168, 2020.

[82] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimiza-

tion. In *Proceedings of 3rd International Conference on Learning Representations*, 2015.

[83] Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *J. Mach. Learn. Res.*, 18(1):559–563, January 2017.

[84] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *Advances in Cryptology—CRYPTO '99*, volume 1666 of *Lecture Notes in Computer Seience*, pages 537–554, 1999.

[85] Denis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. A modular analysis of the Fujisaki–Okamoto transformation. In *Theory of Cryptography*, volume 10677 of *Lecture Notes in Computer Science*, pages 341–371. Springer, 2017.

[86] Tsunekazu Saito, Keita Xagawa, and Takashi Yamakawa. Tightly-secure key-encapsulation mechanism in the quantum random oracle model. In *Advances in Cryptology—EUROCRYPT 2018*, volume 10822 of *Lecture Notes in Computer Science*, pages 520–551. Springer, 2018.

[87] Nina Bindel, Mike Hamburg, Kathrin Hövelmanns, Andreas Hülsing, and Edoardo Persichetti. Tighter proofs of CCA security in the quantum random oracle model. In *Theory of Cryptography*, volume 11892 of *Lecture Notes in Computer Science*, pages 61–90, 2019.

[88] NIST. Post-quantum cryptogprahy. `https://csrc.nist.gov/projects/post-quantum-cryptography`, 2020.

[89] Qian Guo, Johansson Thomas, and Alexander Nilsson. A key-recovery timing attack on post-quantum primitives using the Fujisaki–Okamoto transformation and its application on FrodoKEM. In *Advances in Cryptology—CRYPTO '20*, volume 12171 of *Lecture Notes in Computer Seience*, pages 359–386, 2020.

[90] Prasanna Ravi, Sujoy Sinha Roy, Anupam Chattopadhyay, and Shivam Bhasin. Generic side-channel attacks on CCA-secure lattice-based PKE and KEMs. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2020:307–335, 2020.

[91] Zhuang Xu, Owen Pemberton, Sujoy Sinha Roy, and David Oswald. Magnifying side-channel leakage of lattice-based cryptosystems with chosen ciphertexts: The case study of Kyber. IACR ePrint archive: Report 2020/912, 2020. `https://eprint.iacr.org/2020/912`.

[92] Prasanna Ravi, Shivam Bhasin, Sujoy Sinha Roy, and Anupam Chattopadyay. On exploiting message leakage in (few) NIST PQC candidates for practical mes-

sage recovery and key recovery attacks. IACR ePrint archive: Report 2020/1559, 2020. https://eprint.iacr.org/2020/1559.

[93] Bo-Yeon Sim, Jihoon Kwon, Joohoo Lee, Il-Ju Kim, Tae-Ho Lee, Hyojin Yoon, Jihoon Cho, and Dong-Gak Han. Single-trace attacks on message encoding in lattice-based KEMs. *IEEE Access*, 8:183175–183191, 2020.

[94] Prasanna Ravi, Martianus Frederic Ezerman, Shivam Bhasin, Anupam Chattopadhyay, and Sujoy Sinha Roy. Generic side-channel assisted chosen-ciphertext attacks on Streamlined NTRU Prime. IACR ePrint archive: Report 2021/718, 2021. https://eprint.iacr.org/2021/718.

[95] Bo-Yeon Sim, Jihoon Kwon, Kyu Young Choi, Jihoon Cho, Aeson Park, and Dong-Guk Han. Novel side-channel attacks on quasi-cyclic code-based cryptography. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2019:180–212, 2019.

[96] Norman Lahr, Ruben Niedarhgen, Richard Petri, and Simona Samardjiska. Side channel information set decoding using iterative chunking: Plaintext recovery from the "Classic McEliece" hardware reference implementation. In *Advances in Cryptology—ASIACRYPT 2020*, volume 12491 of *Lecture Notes in Computer Seience*, pages 881–910, 2020.

[97] Brian Koziel, Reza Azarderakhsh, and David Jao. Side-channel attacks on quantum-resistant supersingular isogeny Diffie–Hellman. In *Selected Areas in Cryptography—SAC 2017*, volume 10719 of *Lecture Notes in Computer Science*, pages 64–81, 2017.

[98] Fan Zhang, Bolin Yang, Xiaofei Dong, Sylvain Guilley, Zhe Liu, Wei He, Fangguo Zhang, and Kui Ren. Side-channel analysis and countermeasure design on ARM-based quantum-resistant SIKE. *IEEE Transactions on Computers*, 69:1681–1693, 2020.

[99] Steven D. Galbraith, Christophe Petit, Barak Shani, and Bo Yan Ti. On the security of supersingular isogeny cryptosystems. In *Advances in Cryptology—ASIACRYPT 2016*, volume 10031 of *Lecture Notes in Computer Seience*, pages 63–91, 2016.

[100] Ciprian Băetu, F. Betül Durak, Huguenin-Dumittan Loïs, Abdullah Talayhan, and Serge Vaudenay. Misuse attacks on post-quantum cryptosystems. In *Advances in Cryptology—Eurocrypt 2019*, volume 11477 of *Lecture Notes in Computer Science*, pages 747–776, 2019.

[101] Loïs Huguenin-Dumittan and Serge Vaudenay. Classical misuse attacks on NIST

round 2 PQC - the power of rank-based schemes. In Mauro Conti, Jianying Zhou, Emiliano Casalicchio, and Angelo Spognardi, editors, *Applied Cryptography and Network Security - 18th International Conference, ACNS 2020, Rome, Italy, October 19-22, 2020, Proceedings, Part I*, volume 12146 of *Lecture Notes in Computer Science*, pages 208–227. Springer, 2020.

[102] Jeffrey Hoffstein and Joseph H. Silverman. Reaction attacks against the NTRU public key cryptosystem. NTRU Technical Report, 1999. Available at `https://ntru.org/resources.shtml`.

[103] Éliane Jaulmes and Antoine Joux. A chosen-ciphertext attack against NTRU. In Mihir Bellare, editor, *Advances in Cryptology - CRYPTO 2000, 20th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2000, Proceedings*, volume 1880 of *Lecture Notes in Computer Science*, pages 20–35. Springer, 2000.

[104] Jintai Ding, Joshua Deaton, Kurt Schmidt, Vishakha, and Zheng Zhang. A simple and efficient key reuse attack on NTRU cryptosystem. IACR ePrint archive: Report 2019/1022, 2019. `https://eprint.iacr.org/2019/1022`.

[105] Xiaohan Zhang, Chi Cheng, Yue Qin, and Ruoyu Ding. Small leaks sink a great ship: An evaluation of key reuse resilience of PQC third round finalist NTRU-HRSS. IACR ePrint archive: Report 2021/168, 2021. `https://eprint.iacr.org/2021/168`.

[106] Keita Xagawa, Akira Ito, Rei Ueno, Junko Takahashi, and Naofumi Homma. Fault-injection attacks against NIST's post-quantum cryptography round 3 KEM candidates. IACR ePrint archive: Report 2021/840, 2021. `https://eprint.iacr.org/2021/840`.

[107] Qian Guo, Thomas Johansson, and Paul Stankovski. A key recovery attack on MDPC with CCA security using decoding errors. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I*, volume 10031 of *Lecture Notes in Computer Science*, pages 789–815, 2016.

[108] Jerzy Neyman and Egon Sharpe Peason. IX. On the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society A*, 231:694–706, 1933.

[109] Matthias J. Kannwischer, Joost Rijneveld, Peter Schwabe, and Ko Stoffelen. `pqm4`: Testing and benchmarking NIST PQC on ARM Cortex-M4. IACR ePrint

archive: Report 2019/844, 2019. https://eprint.iacr.org/2019/844.

[110] Post-quantum crypto library for the ARM Cortex-M4. https://github.com/mupq/pqm4, April 2021.

[111] Tohoku University. Cryptographic hardware project. http://www.aoki.ecei.tohoku.ac.jp/crypto/.

[112] Peter Schwave and Ko Stoffelen. All the AES you need on Cortex-M3 and M4. In *Selected Areas in Cryptography—SAC 2016*, volume 10532 of *Lecture Notes in Computer Science*, pages 180–194, 2016.

[113] Fast, constant-time and masked AES assembly implementations for ARM Cortex-M3 and M4. https://github.com/Ko-/aes-armcortexm, May 2021.

[114] Tobias Schneider and Amir Moradi. Leakage assesment methodology—A clear roadmap for side-channel evaluations. In *Workshop on Cryptographic Hardware and Embedded Systems*, volume 9293 of *Lecture Notes in Computer Science*, pages 495–513. Springer, 2015.

[115] Aein Rezaei Shahmirzadi, Dušan Božilov, and Amir Moradi. New first-order secure AES performance records. *IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES)*, pages 304–327, 2019.

[116] Joan Daemen. Changing of the guards: A simple and efficient method for achieving uniformity in threshold sharing. In *Cryptographic Hardware and Embedded Systems – CHES 2017*, volume 10529 of *Lecture Notes in Computer Science*, pages 137–153. Springer, 2017.

[117] Alexandre Duc, Sebastian Faust, and François-Xavier Standaert. Making masking security proofs concrete - or how to evaluate the security of any leaking device. In *EUROCRYPT (1)*, pages 401–429. Springer, 2015.

[118] Alexandre Duc, Stefan Dziembowski, and Sebastian Faust. Unifying leakage models: From probing attacks to noisy leakage. In *EUROCRYPT*, pages 423–440. Springer, 2014.

[119] Loïc Masure, Cécile Dumas, and Emmanuel Prouff. A comprehensive study of deep learning for side-channel analysis. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2020, Issue 1:348–375, 2019.

[120] Benedikt Gierlichs, Lejla Batina, Pim Tuyls, and Bart Preneel. Mutual information analysis. In *Cryptographic Hardware and Embedded Systems - CHES 2008, 10th International Workshop, Washington, D.C., USA, August 10-13, 2008. Proceedings*, volume 5154 of *Lecture Notes in Computer Science*, pages 426–442. Springer, 2008.

[121] Annelie Heuser, Olivier Rioul, and Sylvain Guilley. Good is not good enough - deriving optimal distinguishers from communication theory. In *CHES*, pages 55–74. Springer, 2014.

発表論文等

学術雑誌論文

1. Rei Ueno, Keita Xagawa, Yutaro Tanaka, Akira Ito, Junko Takahashi and Naofumi Homma, "Curse of Re-encryption: A Generic Power/EM Analysis on Post-Quantum KEMs," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, Vol. 2022, No. 1, pp. 296-322, November 2021.
2. Akira Ito, Rei Ueno and Naofumi Homma, "An Algebraic Approach to Verifying Galois-Field Arithmetic," *IEICE Transactions on Information and Systems,* Vol. 8, pp.1083–1091, August 2021.
3. Akira Ito, Rei Ueno and Naofumi Homma, "Imbalanced Data Problems in Deep Learning-Based Side-Channel Attacks: Analysis and Solution," *IEEE Transactions on Information Forensics and Security,* Vol. 16, pp.1083–1091, June 2021.
4. Akira Ito, Rei Ueno and Naofumi Homma, "Efficient Formal Verification of Galois-Field Arithmetic Circuits Using ZDD Representation of Boolean Polynomials," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, （To be appeared）.
5. Akira Ito, Rei Ueno, Naofumi Homma and Takafumi Aoki, "Characterizing Parallel Multipliers for Detecting Hardware Trojans," *Journal of Applied Logics*, Vol. 5, No. 9, pp. 1815 - 1832, December 2018.

国際会議論文

1. Keita Xagawa, Akira Ito, Rei Ueno, Junko Takahashi and Naofumi Homma, "Fault-Injection Attacks against NIST' s Post-Quantum Cryptography Round 3 KEM Candidates," *27th Annual International Conference on the Theory and Applications of Cryptology and Information Security (ASIACRYPT),* Online,

December 2021.

2. Akira Ito, Rei Ueno and Naofumi Homma, "A Formal Approach to Identifying Hardware Trojans in Cryptographic Hardware," *IEEE 51th International Symposium on Multiple-Valued Logic (ISMVL)*, Online, May 2021.
3. Ville Yli-Mäyry, Akira Ito, Naofumi Homma, Shivam Bhasin, and Dirmanto Jap, "Extraction of Binarized Neural Network Architecture and Secret Parameters Using Side-Channel Information," *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, Daegu, Korea and online, May 2021.
4. Francesco Regazzoni, Shivam Bhasin, Amir Ali Pour, Ihab Alshaer, Furkan Aydin, Aydin Aysu, Vincent Beroulle, Giorgio Di Natale, Paul Franzon, David Hely, Naofumi Homma, Akira Ito, Dirmanto Jap, Priyank Kashyap, Ilia Polian, Seetal Potluri, Rei Ueno, Elena-Ioana Vatajelu, and Ville Yli-Mäyry, "Machine Learning and Hardware security: Challenges and Opportunities," *2020 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, Online, pp. 1–6, November 2020 (Invited).
5. Akira Ito, Rei Ueno and Naofumi Homma, "Effective Formal Verification for Galois-field Arithmetic Circuits with Multiple-Valued Characteristics," *IEEE 50th International Symposium on Multiple-Valued Logic (ISMVL)*, pp. 46–51, Miyazaki Japan and online, January 2020.
6. Dirmanto Jap, Ville Yli-Mäyry, Akira Ito, Rei Ueno, Shivam Bhasin and Naofumi Homma, "Practical Side-Channel Based Model Extraction Attack on Tree-Based Machine Learning Algorithm," *Applied Cryptography and Network Security Workshops*, pp. 93–105, Online, October 2018.
7. Akira Ito, Rei Ueno and Naofumi Homma, "A Non-Reversible Insertion Method for Hardware Trojans Based on Path Delay Faults," *International Workshop on Security Proofs for Embedded Systems (PROOFS)*, pp. 50–67, Amsterdam, Netherlands, September 2018.
8. Akira Ito, Rei Ueno, Naofumi Homma and Takafumi Aoki, "On the Detectability of Hardware Trojans Embedded in Parallel Multipliers, " *IEEE 48th International Symposium on Multiple-Valued Logic*, pp. 62–67, Linz, Austria, May 2018.

国内学会

1. 上野嶺, 草川恵太, 田中裕太郎, 伊東燦, 高橋順子, 本間尚文, “耐量子鍵カプセル化メカニズムに対する一般化サイドチャネル攻撃,” 2022 年暗号と情報セキュリティシンポジウム (SCIS 2022)，No. 1C1-1, 大阪府, January 2022.
2. 草川恵太, 伊東燦, 上野嶺, 高橋順子, 本間尚文, “NIST PQC Round3 候補の鍵カプセル化方式への故障注入攻撃,” 2022 年暗号と情報セキュリティシンポジウム (SCIS 2022)，No. 2A2-1, 大阪府, January 2022.
3. 伊東燦, 上野嶺, 本間尚文, “マスキング対策実装に対するサイドチャネル攻撃成功確率の情報理論的解析,” 2022 年暗号と情報セキュリティシンポジウム (SCIS 2022)，No. 4C1-2, 大阪府, January 2022.
4. 伊藤圭吾, 伊東燦, 上野嶺, 福島和英, 清本晋作, 本間尚文, “軽量暗号 GIMLI-AEAD に対する深層学習を用いたサイドチャネル解析の検討,” 情報セキュリティ研究会, 信学技報, Vol. 121, No. 239, ISEC2021-45, pp. 20–25, Online, November 2021.
5. 齋藤宏太郎, 伊東燦, 上野嶺, 本間尚文, “耐タンパー性を有する CRT-RSA ソフトウェアに対する深層学習に基づく単一波形サイドチャネル攻撃,” ハードウェアセキュリティ研究会，No. 206, Online, October 2021.
6. 小嶋健太，伊東燦, 上野嶺, 本間尚文, “マスキング対策された暗号ハードウェアへの深層学習を用いたサイドチャネル解析,” ハードウェアセキュリティ研究会，No. 121, Online, July 2021.
7. 伊東燦, 上野嶺, 本間尚文, “深層学習を用いたサイドチャネル攻撃の性能評価手法に関する検討,” ハードウェアセキュリティ研究会，No. 1, Online, April 2021.
8. 伊東燦, 齋藤宏太郎, 上野嶺, 本間尚文, “深層学習を用いたサイドチャネル攻撃における不均衡データ問題の解析と解消法,” 2021 年暗号と情報セキュリティシンポジウム (SCIS 2021)，No. 1D1-4, Online, January 2021.
9. 伊東燦, 上野嶺, 本間尚文, “暗号ハードウェアのネットリストに対するハードウェアトロイ検知手法,” ハードウェアセキュリティ研究会，No. 211, Online, January 2021.
10. 伊東燦, 上野嶺, 本間尚文, “決定グラフ表現に基づくハードウェアトロイ検知手法,” 第 43 回多値論理フォーラム，Online, September 2020.
11. Ville Yli-Mäyry, 伊東燦, 上野嶺, Shivam Bhasin, Dirmant Jap, 本間尚文, “FPGA 向け二値化ニューラルネットワークへの電磁波解析攻撃の検討,” ハードウェアセキュリティ研究会，No. 1，Online, April 2020.
12. 伊東燦, 上野嶺, 本間尚文, “暗号ハードウェアに対する形式的ハードウェアトロ

イ検出手法,” 2020 年暗号と情報セキュリティシンポジウム (SCIS 2020)，No. 2E3-1，高知県, January 2020.

13. 伊東燦, 上野嶺, 本間尚文, “多標数ガロア体算術演算回路の形式的検証手法,” 第 33 回多値論理とその応用研究会，No. 17，兵庫県, January 2020.
14. 伊東燦, 上野嶺, 本間尚文, “ブール多項式の ZDD 表現を用いたガロア体算術演算回路の形式的検証手法,” 第 42 回多値論理フォーラム，No. 08，宮崎県, January 2020.
15. 伊東燦, 上野嶺, 本間尚文, “ガロア体算術に基づく暗号ハードウェアの形式的トロイフリー性検証,” セキュリティサマーサミット 2019，No. B-2，高知県, August 2020.
16. 伊東燦, 上野嶺, 本間尚文, “ガロア体演算に基づく暗号ハードウェアにおける HT 検知技術,” LSI とシステムのワークショップ，ポスター No. 9，東京都, May 2019.
17. 伊東燦, 上野嶺, 本間尚文, 青木孝文, “ガロア体ハードウェアアルゴリズムの形式的トロイフリー性検証手法,” 暗号と情報セキュリティシンポジウム (SCIS 2019), No. 2D1, January 2019.
18. 伊東燦, 上野嶺, 本間尚文, 青木孝文, “ハードウェアトロイに耐性を有する算術演算回路の構成とその評価,” ハードウェアセキュリティ夏のワークショップ, ポスター No. 9, September 27, 2018.
19. 伊東燦, 上野嶺, 本間尚文, 青木孝文, “パス遅延故障に基づくハードウェアトロイの系統的挿入法とその評価,” 夏のセキュリティワークショップ 2018, pp. 349–356, 北海道, July 2018.
20. 伊東燦, 上野嶺, 本間尚文, 青木孝文, “ハードウェアトロイ挿入が困難な公開鍵暗号データパスに関する検討,” ハードウェアセキュリティフォーラム 2017, ポスター No.9, 東京都, December 2017.
21. 伊東燦, 上野嶺, 本間尚文, 青木孝文, “算術演算ハードウェアアルゴリズムの改変検知に関する検討,” 第 40 回多値論理フォーラム, Vol.40, No.16, 飛鳥村, 奈良県, September 2017.
22. 伊東燦, 上野嶺, 本間尚文, 青木孝文, “乗算アルゴリズムに対するハードウェアトロイ挿入可能性の評価,” 平成 29 年度電気関係学会東北支部連合大会, 1E05, August 24, 2017.

受賞等

1. SCIS 論文賞，伊東燦; “深層学習を用いたサイドチャネル攻撃における不均衡データ問題の解析と解消法,” 2022 年暗号と情報セキュリティシンポジウム (SCIS 2022). （著者：伊東燦, 上野嶺, 本間尚文）
2. 多値論理フォーラム奨励賞，伊東燦, 上野嶺，本間尚文; “ブール多項式の ZDD 表現を用いたガロア体算術演算回路の形式的検証手法,” 第 42 回多値論理フォーラム. （著者：伊東燦, 上野嶺，本間尚文）
3. ハードウェアセキュリティ研究会若手優秀賞，伊東燦; “ガロア体算術に基づく暗号ハードウェアの形式的トロイフリー性検証,” セキュリティサマーサミット 2019. （著者：伊東燦, 上野嶺，本間尚文）

謝辞

本論文は，著者が東北大学大学院工学研究科通信工学専攻環境調和型セキュア情報システム研究分野（本間研究室）において行った研究を取りまとめたものです．本研究を推し進めるにあたり，多くの方々からご協力とご助言を頂きました．

恩師 本間尚文教授には，熱心なご指導と終始変わらぬ励ましを頂きました．先生の研究・教育に対する真摯なご姿勢から多くを学ばせて頂けたことを銘記し，ここに改めて深く感謝の意を表します．

本論文をまとめるにあたり，羽生貴弘教授ならびに堀尾喜彦教授より，それぞれの御専門の立場から大変有意義なご意見を賜りました．ここに深く感謝いたします．

本研究室の助教である上野嶺博士には，著者が研究室へ配属されて以来，直接研究に対する熱心なご指導を頂きました．研究に関する様々なご助言や励ましをいただくだけでなく，生活面においても大変お世話になりました．ここに改めて深く感謝の意を表します．

また，在学中には多くの研究者の方々と共同研究させて頂く機会に恵まれました．共同研究を通じ，新たな洞察と有意義な研究成果を得ることができました．皆様には改めて感謝の意を表すとともに，以下に名前を挙げさせていただきます．草川恵太博士（NTT 社会情報研究所），高橋順子博士（NTT 社会情報研究所），永田真教授（神戸大学），林優一教授（奈良先端科学技術大学院大学），藤本大介助教（奈良先端科学技術大学院大学），Ville Yli-Maeyry 博士（現 Secure-IC），伊藤圭吾氏，小嶋健太氏（現 東京工業大学），齋藤宏太朗氏，田中裕太郎氏．

インターンシップにおいて在籍した Telecom ParisTech の Jean-Luc Danger 教授，Ulrich Kühne 助教には，ハードウェアトロイ検知のための等価性検証に関する様々なご意見やご協力を頂きました．ここにお礼申し上げます．

日頃の研究室生活において様々な面で御協力いただいた研究室諸氏に心よりお礼申し上げます．

日本学術振興会特別研究員制度による助成をして頂いたことを感謝いたします．

最後に，長きにわたる研究生活を応援し支えてくれた両親に感謝の意を表し，本論文を結びます．

2022 年 2 月 12 日