深層学習を用いた
Arbiter PUFバリアントの安全性評価

八代　理紗

電気通信大学

2023年　3月

深層学習を用いた
Arbiter PUFバリアントの安全性評価

八代　理紗

電気通信大学

大学院情報理工学研究科

博士学位申請論文

2023年　3月

深層学習を用いた
Arbiter PUFバリアントの安全性評価

博士論文審査委員会:

主査:　崎山　一男 教授

委員:　岩本　貢 教授

委員:　大坐畠　智　准教授

委員:　李　陽 准教授

外部審査委員:　産業技術総合研究所

堀　洋平　主任研究員

A Security Evaluation of Arbiter PUF and Its Variants Using Deep Learning

Risa Yashiro

The Internet of Things (IoT) has increased the convenience of life, although various counterfeit electro devices are distributed worldwide. However, there have even been reports of counterfeits of Integrated Circuit (IC) chips implemented to control IoT devices' operation. Since IC chips are the basis of IoT devices, counterfeit IC chips have many security risks: faulty operation or information leakage. Nowadays, authentication technology and information confidentiality are essential to prevent fake IC chips. Authentication methods using information stored in non-volatile memory are usual; however, in the case of IoT devices, attackers have many opportunities to obtain them. Such IoT devices have the problem that confidential information can be stolen by physical attacks such as intrusion attacks by attackers. Physically Unclonable Function (PUF) is a technology that enables authentication using physical characteristics without requiring non-volatile memory by exploiting the physical features of each chip that are different due to manufacturing variations in IC chips. That is, PUFs generate an electrically unique response. Although it is challenging to replicate PUFs physically, it has been reported that it is possible to make a clone mathematically using machine learning.

In this dissertation, the properties of PUFs and attack scenarios are organized. Besides, security evaluations using deep learning for different PUFs and authentication methods are performed. The research dealt with three types of PUFs that are expected to increase the difficulty of mathematical replication while using existing PUF configurations; 1) PUF performing sequential processing on a hardware instance, 2) PUF utilizing parallelized hardware instances, and 3) PUF with intentional error to counteract deep-learning attacks. As for the first type of PUF, the deep-learning prediction rate has increased as iteratively using the same hardware instance. On the other hand, we find that the second type of PUF lowers the prediction rate as the number of parallel implementations increases with the penalty of the hardware cost. Additionally, Monte Carlo simulations on PUF-based authenticators show that legitimate PUFs and clones can be distinguished.

Finally, the third type of PUF, in which we inject intentional errors, can reduce the prediction success rate by increasing the error. This research clarifies the usefulness of the design and evaluation methods for improving the security of PUF.

深層学習を用いたArbiter PUFバリアントの安全性評価

八代　理紗

Internet of Things (IoT) の普及により生活の利便性が増している．一方で，IoT機器の動作を制御するために実装される Integrated Circuit (IC) チップの模倣品が問題となっている．IC チップの模倣品は性能や信頼性が低く，動作不良や情報流出などのセキュリティ上のリスクを引き起こす．そのため，模倣品を防ぐために認証技術や情報秘匿などが重要となる．IC チップの認証方式として，不揮発メモリに保存した情報を用いた方式が一般的だが，IoT 機器の場合攻撃者の手元にある機会も多い．そのように攻撃者が入手した IoT 機器は，侵襲攻撃などの物理的な攻撃によって機密情報の盗難が問題となっている．一対策技術として，Physically Unclonable Function (PUF) があげられる．PUF は不揮発メモリ等を必要とせずに物理特性を用いた認証を可能とする技術である．IC チップの製造時には，チップごとに物理的特徴のばらつきが生じる．そのことを利用して，PUF はチャレンジに対する電気的に固有なレスポンスを生成する．PUF は物理的に複製が困難であるが，CRP から機械学習などを用いて数学的に複製が可能であることが報告されている．

本研究では，PUF の性質と攻撃シナリオについて体系化し，異なる PUF や認証方式に対して深層学習を用いて安全性評価を行う．この研究では既存の PUF の構成を利用し，数学的複製困難性を向上できると期待される次の 3 つの PUF を取り扱う．1）ハードウェアインスタンスに対して時系列処理を行った PUF，2) 並列実装されたハードウェアインスタンスを用いる PUF，3) 深層攻撃への対策として意図的にエラーを注入した PUF の 3 つである．1) の時系列処理を行った PUF では，同じハードウェアインスタンスを繰り返し動作させることによって深層学習による予測成功率が上がった．そのため，数学的複製困難性の向上には適さないことが分かった．一方で，2) の並列実装された PUF では，並列実装の数は予測成功率を低下させるため数学的複製困難性の向上が可能である．しかし，ハードウェアコストが増加する．最後に 3) の意図的にエラーを注入した PUF ではエラーを増やすことで予測成功率を低下させるため数学的複製困難性の向上が可能である．また，モンテカルロシミュレーションを用いて認証システムに対する考察を行い，正規 PUF とクローンの判別が可能なことを示す．

結果として，安全性評価を行った 3 つの PUF では，3) のエラーを注入した PUF が低い実装コストで，効率よく深層学習攻撃を対策できることを示した．本研究を通じて，PUF の安全性向上に対する設計および評価手法の有用性が明らかとなった．

目次

略語一覧

BER	Bit Error Rate
CRP	Challenge and Response Pair
DAPUF	Double Arbiter PUF
DDoS	Distributed Denial of Service
ES	Evaluation Strategy
FF-PUF	Feed-Forward PUF
FPGA	Field Programmable Gate Array
IC	Integrated Circuit
IoT	Internet of Things
LR	Logistic Regression
PUF	Physically Unclonable Function
ReLU	Rectified Linear Unit
RFID	Radio Frequency Identification
RG-DTM PUF	Response Generate Delay Time Measurement PUF
ROPUF	Ring Oscillator PUF

SR Set-Reset

SVM Support Vector Machine

tanh Tangent Hyperbolic

第 1 章

序論

1.1　背景

1.1.1　IoT 時代とセキュリティインシデント

昨今の情報社会の発展に伴い，様々なモノがインターネットに接続され，通信を行なっている．これをモノのインターネット (Internet of Things: IoT) と呼ぶ．2023 年現在，約 400 億個の IoT 機器がインターネットに接続し，通信をしていると推定されている [72]．IoT 機器には，冷蔵庫や照明などといった家電をはじめ，自動車や工場の機械のような大型の機器や医療機器やセンサーなどの小型な機器まで存在している．IoT 機器がインターネットにつながり利便性が上がる一方で攻撃に晒される機会も増えており，IoT 機器のセキュリティは重要である．実際に IoT 機器に起きたセキュリティインシデントとして 2 つの例を挙げる．

■Mirai　IoT 機器のセキュリティの脆弱性をついた攻撃として 2016 年に発表された Mirai が挙げられる [5, 24]．Mirai は IoT 機器に感染してボットネットを構築し，Distributed Denial of Service (DDoS) 攻撃を行うマルウェアの一種である．Mirai は IoT 機器に対して，よく使われ

るパスワードリスト (例: [4]) を用いて辞書攻撃を行い，不正ログインを試みる．そして，ログイン可能な端末ではボットをダウンロードし，さらに不正ログインが可能な IoT 機器を探索する．感染した IoT 機器は一斉に DDoS 攻撃を行い，サービス提供等の障害を引き起こす．Mirai に感染された IoT 機器の中には，辞書攻撃によって不正ログインが可能であり，なおかつ ID やパスワードが変更できない機器も存在した．また，Mirai はソースコードが公開されており [1]，変異種が登場しつづけていること [31] から，対策技術を常に更新する必要がある．Mirai によって，セキュリティが十分ではない IoT 機器が流通しており，サービス停止等の障害を起こすことを示している．

■スマートホームに対するハッキング　IoT 機器から構成されるスマートホームに対する攻撃が報告されている．国際会議 Black Hat 2021 にてスマートホーム化されたカプセルホテルの IoT 機器に対するハッキングが報告された [2]．攻撃されたカプセルホテルでは，各部屋に専用のルータがあり，そのルータに専用のアプリをダウンロードした iPad を接続することで電灯や換気扇，ベッドのリクライニングといった IoT 機器の制御・管理をしていた．しかし，ルータに設定されたパスワードに脆弱性があり，さらに，ルータ接続する iPad の認証や通信の秘匿化が行われていなかったため，不正にルータにアクセスすることができ，他の部屋の IoT 機器の制御を行うことができた．

この報告 [2] には含まれていないが，IoT 機器の状態を遠隔で入手できる場合には，プライバシー情報の推定が可能になることがある．例えば，IoT 機器の中でもシンプルな仕様である照明について考える．もしも照明の点灯時間が平日の決まった時間であれば，ユーザは日勤で働いていることが推定可能である．また，毎日同じ時間についている電灯が，ある日その時間になってもつかなかった場合，ユーザが家に不在であることが推定できる．この場合，スマートフォンを用いて遠隔操作し点灯する装置 [70] などにより在宅状況の秘匿などは対策を施すこ

とが可能である．上記のように IoT 機器が取り扱う情報には個人のプライバシー情報が含まれており，セキュリティ対策を検討しなければいけない問題が多く報告されている [30, 57, 61]．なりすましなどの第三者による情報の悪用やユーザのプライバシー保護のためには，情報の盗聴や改ざんを防ぐ必要がある．盗聴や改ざんを防ぐためには，通信内容の秘匿化や接続先が正接続先が正しい機器やユーザか認証することしい機器やユーザか認証することなどが有効である．

1.1.2 模倣品の流通と抑制技術

世界中から様々な物品の入手が容易になっている一方で模倣品の流通被害も増加している．2016 年に模倣品による経済被害は，世界貿易額の約 2.5% に相当する約 500 億ドル (55 兆円) であると報告されている [43]．模倣品による被害は，自動車部品や医薬品，食料など様々な産業分野で報告されている [71]．また，高級ブランド商品に対して，「スーパーコピー」と称する素材などにこだわった精巧な模倣品がネット上で売買されている．このスーパーコピーは個人消費の範囲では違法にならないため，取り締まるのが難しい．電子部品に対しても模倣品は報告されており，被害規模は経済被害の約 15% である約 83 億ドル (8 兆円) を占めているとされている [43]．一般的に模倣品は正規品に比べて安価で取引されているが，正規品よりも性能や信頼性が低い．電子部品の場合，消費者が模倣品だと気づくことも難しいため，消費者が正規品だと思い込み使用し，メーカーの信頼性を失墜させる恐れもある．実際に，消費者が電子機器の模倣品を使用したことで発火を起こした事故も報告されている．そのため，模倣品に対する対策は急を要する．

とりわけ，電子機器の動作を制御する Integrated Circuit (IC) チップが模倣品だった場合には，セキュリティ上のリスクは深刻となる．実際に，IC チップの模倣品被害が報告されてい

る [60, 62]．IC チップによるセキュリティインシデントとして，動作不良やセキュリティ上の安全性の低下などが挙げられる．模倣 IC チップが重要インフラの電子機器に使用されていた場合には，より甚大な経済的損失を起こす．また，ユーザが模倣品に気づかずに使用しつづけることで，プライバシー情報が流出する．

流通している IoT 機器の中に模倣 IC チップが紛れ込んでいる可能性は高い．模倣 IC チップの流通を防ぐためには正規品か確認するための認証が不可欠である．しかし，IoT 機器は小型な機器が多く，IC チップの回路規模には制約がかかる．そのため，認証に必要な回路を追加で実装することが困難な場合がある．そこで，低コストで実装可能なセキュリティプリミティブを IC チップに実装し，認証や情報秘匿を行うことが重要となる．

模倣品流通を防ぐために流通している認証技術として，ホログラム [15] や Radio Frequency Identification (RFID) [9] などが代表例である．認証対象が有する固有情報を用いて認証を行う人工物メトリクス [38] や Physically Unclonable Function (PUF) [46, 47] は比較的新しい技術である．

■ホログラム　ホログラムは 1948 年に発見されたレーザーを利用した立体画像写真のことを指す．ホログラムは，偽造防止策の身近な例として紙幣に使用されている．紙幣に使用されるホログラムは使用者が本物か否かを視覚的に簡単に見分けられるようになっている．例えば，1 万円札の場合，紙幣を見る角度によって「10000」,「桜」,「日本銀行のマーク」の三種類の画像を見ることが可能である．ホログラムは作製時の原版が入手できないと複製が困難であり，製造に必要な装置のコストも高額なため偽造が困難だとされており，さらに偽造防止のために透かしやマイクロ文字といった偽装防止技術と組み合わせて流通させている．

■RFID　Radio Frequency Identification (RFID) は，電波を用いた通信により非接触で半導体チップ上のメモリの読み書きを行う技術であり，流通管理などに利用される．RFID は非接触で通信を行うことができるため，汚れや位置に影響を受けづらい．また，通常のバーコードの 1 対 1 通信だけではなく，1 対多通信が可能である．そのため，倉庫に貯蓄されている部品の管理などが短期間で行える．さらに，作業行程や流通ルートをメモリに記録することも可能である．流通に使用される RFID は，コストを抑えるために軽量であることが望まれるが，軽量 RFID にはセキュリティ上の問題があることが指摘されている [25, 69]．RFID にセキュリティ技術を追加するためには，暗号化，認証および物理的な攻撃に耐える耐タンパー性が要求される．また，RFID は不揮発性メモリに情報を格納しているため，メモリ上のデータの読み込み・書き込みの保護が不十分な場合には情報が漏れ，なりすましが可能となる．

■人工物メトリクス　人工物メトリクスは，人工物の固有性を顕微鏡などにより認証する技術である．人工物メトリクスの例として印刷時に生じる固有性を用いて判別を行う Shachihata Authentication Management Programs (SAMP) [73, 75] がある．SAMP は対象となる物体および範囲を事前に決定しておき，その写真の固有値を事前に抽出しておく．認証時には既定の位置に対象を配置することで固有値の抽出および判別を行う．具体的な実用例としては，印刷されたラベルの固有値を用いて，医薬品の在庫管理として破棄情報やトラッキングを管理番号などを用いずに行うことができる．一方で，人工物メトリクスは高性能な顕微鏡など特殊な機器を利用する必要がある．

1.2 Physically Unclonable Function (PUF)

メモリを使用せずに IC チップなどの認証に利用できる技術のひとつに Physically Unclonable Function (PUF) [47] がある．PUF は，IC チップに生じる半導体ばらつきなどの物理的特徴を基に固有な出力を導出するセキュリティ回路である．物理的特徴は製造時に意図せずに生じるため，意図的な物理的複製が困難である．この物理的特徴を情報秘匿のための秘密鍵生成や IC チップの認証に用いることができる．また，PUF で用いる物理的特徴は秘密情報を保存する不揮発メモリを要せずに利用できる．さらに，PUF は比較的低コストで実装可能なため，IoT 機器に対するセキュリティプリミティブとして期待されている．

PUF は，チャレンジ空間 (入力の種類の多さ) によって大きく 2 種類に分類することができる．分類された PUF をそれぞれ Confined PUF と Extensive PUF と呼ぶ．Confined PUF と Extensive PUF は ISO/IEC 20897 [3] で新たに定義された呼称であり，それ以前から用いられていた Weak PUF，Strong PUF [18, 50] とそれぞれ同じである [8]．Confined PUF はチャレンジ空間が小さい PUF の総称であり，SRAM PUF [18]，ラッチ PUF [58] や Butterfly PUF [26] などがある．Confined PUF の代表例である SRAM PUF は電源投入時の初期値が SRAM セルごとに異なることを物理的特徴として利用する．Extensive PUF はチャレンジ空間が大きい PUF の総称であり，Arbiter PUF [16]，Ring Oscillator PUF (ROPUF) [59]，Feed-Forward PUF (FF-PUF) [27] などがある．代表例である Arbiter PUF は数ビットのバイナリをチャレンジとし，回路内で発生する遅延時間を基にレスポンス値を出力する．

1.2.1 PUF とセキュリティ要件

IC チップに実装された PUF が安全に使用可能かどうか判断するために，セキュリティ要件が提唱されている．具体的には，再現性，ユニーク性，ランダム性，耐タンパー性，物理的複製困難性，数学的複製困難性といった性質が一定の基準を満たすことが必要とされている．再現性，ユニーク性，ランダム性についてはレスポンスを用いた評価指標 [21, 37] が提案されており，使用目的に合わせて基準を設定する必要がある．

■再現性　再現性は，同じ PUF に対して同じチャレンジを入力した時，同じレスポンスが出力される性質である．再現性が低い場合，出力されるレスポンスは安定しない．不安定なレスポンスを秘密鍵生成に用いると，検証者と被認証者で値が異なるため，同じ秘密鍵を共有することができず，暗号化を利用した安全な情報共有が不可能となる．そこで再現性の低い PUF の利用に関しては，不安定なレスポンスを破棄できるように長いレスポンス長にしたり，誤り訂正を用いたりするなどの工夫が必要になる．

■ユニーク性　ユニーク性は，同じチャレンジを異なる PUF に入力した時に異なるレスポンスが出力される性質である．ユニーク性が低い場合，全ての PUF は同じチャレンジに対して類似したレスポンスを出力する．その結果，PUF を認証することが困難になる．つまり，ユニーク性が低い PUF を認証した場合，ある ID に対する認証情報 (レスポンス）が複数の PUF で同じになるため識別が困難となる．

■ランダム性　ランダム性は，出力される未使用の入力に対してレスポンスが予測困難である性質のことを指す．ランダム性が低い場合，レスポンスはあるパターンに沿って出力される．ランダム性が低いと，攻撃者は攻撃コストをかけなくても出力されるレスポンスが予測可能と

なり，なりすましが可能になる．

■耐タンパー性　耐タンパー性は，攻撃者が PUF に直接アクセスしても情報を取得することが困難な性質である．例えば不揮発性メモリのように情報が IC チップ内に残っている場合にはなんらかの物理手段により情報を読み取ることができる．PUF の場合は，ばらつきに関する物理情報を取得しようとすると，PUF そのものの性質が変わる可能性が高い．例えば，攻撃者が PUF の出力を見るためにチップを無理やり開封した場合，レスポンスにエラーが生じることになる．この場合，認証が通らなくなり，攻撃者および異常が起こった PUF を排除することが可能である．

■物理的複製困難性　物理的複製困難性は，同じ PUF を物理的に作製することが困難な性質である．PUF はわずかな物理的特徴の違いに基づきレスポンスを決定する．そのため，回路構成や物理的特徴が既知でも同じ物理的特徴を再現することが難しいため，PUF は物理的複製困難性を有しているといえる．

■数学的複製困難性　数学的複製困難性は入出力の関係性を明らかにすることが困難である性質のことを指す．Extensive PUF に対しては，攻撃者が入出力のペアをいくつか取得し，機械学習などを用いて入出力の関係性を明らかにする数学的複製が可能であると報告されている．

1.2.2　PUF を秘密鍵として利用した認証方式

PUF を秘密鍵として認証に利用する方式が提案されている [35, 36]．IoT 機器として様々な用途があるドローンの認証に PUF を適用する研究も近年報告されている [42, 45]．ドローンは飛行情報に関するプライバシー情報を含んでいる．また，機器を乗っ取られると大事故につながる．実際にドローンをハッキングした事例なども挙げられており [44]，認証機能を付与す

る必要性が高まっている．さらに，耐量子計算機暗号に対しても秘密鍵として PUF を利用する方法が提案されている [11].

PUF は環境ノイズの影響により，レスポンスにエラーが生じることがある．秘密鍵として利用する場合，1 ビットでも異なると利用ができない．そこで一般的には，Fuzzy Extractor [**?**] や Reverse Fuzzy Extractor [20] を利用し，誤り訂正を行う．Fuzzy Extractor は基準となるレスポンスを基にヘルパーデータを作成し，ノイズのあるレスポンスから基準となるレスポンスを復元する．Reverse Fuzzy Extractor は Fuzzy Extractor よりも回路規模が小さくて済むため，IoT 機器などに向いている．

1.2.3　Extensive PUF を利用したチャレンジレスポンス認証と攻撃手法

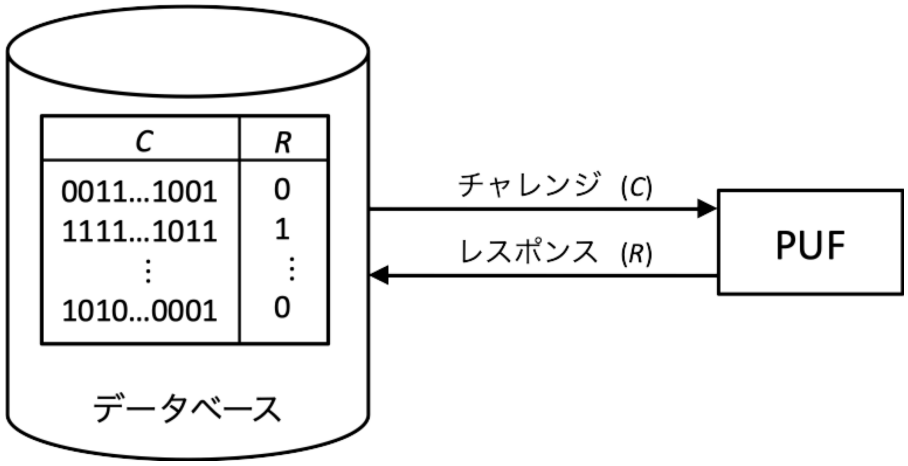

図. 1.1　チャレンジレスポンス認証

PUF の再現性とユニーク性といった性質はチャレンジレスポンス認証に利用可能である．図 1.1 に PUF を用いたチャレンジレスポンス認証の仕組みを示す．チャレンジレスポンス認証のフローは以下になる．

1. 認証者は，流通する前の PUF からチャレンジに対するレスポンスのペア (Challenge–Response Pair, CRP) を取得し，データベースに保存しておく．
2. 認証時には，保存してある CRP の中から複数の CRP を選択し，チャレンジのみを被認証者に与える．
3. 被認証者は与えられたチャレンジを PUF に入力し，それに対応するレスポンスを出力し，そのレスポンスを認証者に返す．
4. 認証者は被認証者から送られてきたレスポンスをデータベースに保存してあるレスポンスと比較し，レスポンスの一致がある閾値以上であれば被認証者を正規と判断する．

PUF のレスポンスは物理情報を基に出力が決定されるため同じチャレンジを入力した時には基本的には同じレスポンスが出力される．しかし，実際には，PUF のレスポンスは環境ノイズなどの影響を受けて，違う値のレスポンスが出力されることがある．環境ノイズによってエラーが多ければ多いほど，再現性は低くなる．先ほど挙げたチャレンジレスポンス認証において，再現性が低い PUF を利用する場合，誤りを許容するための閾値を低く設定する必要がある．閾値を低くすることで PUF は認証を通ることが可能だが，閾値を低くしすぎると認証者が攻撃者を誤認証する可能性が高まる．

PUF のユニーク性が高ければ同じ構成の PUF でも CRP が異なるため，正規の PUF を特定できる．同じ PUF に対して複数回同じチャレンジを与えて認証を行なった場合には，リプレイ攻撃が可能となる．リプレイ攻撃は，攻撃者が認証者と正規 PUF 間で通信された CRP を盗聴し，同じチャレンジに対して盗聴したレスポンスを送信することで誤認証させる攻撃手法のことを指す．この攻撃手法を防ぐために，CRP は使い切りとし，一度認証に使用した CRP は再利用しないようにする必要がある．

閾値を低くした時に脅威となるのが，Arbiter PUF の内部の遅延時間を推測するモデリング攻撃 [28, 29, 50, 51] である．モデリング攻撃は，Arbiter PUF の遅延時間の推測を行う攻撃手法である．前述のチャレンジレスポンス認証の場合，通信路にチャレンジとレスポンスが流れている．そのため，攻撃者は通信路を盗聴することでチャレンジとそれに対するレスポンスを取得可能である．Arbiter PUF は遅延時間を基にチャレンジに対するレスポンスを決定しているため，複数の CRP を集めることによって，その Arbiter PUF の遅延時間を推定することが可能となる．遅延時間の推定により，攻撃者が取得した CRP だけではなく，未知のチャレンジに対するレスポンスが推定可能となる．つまり，実際の Arbiter PUF のクローンの作製が可能となる．再現性の低い PUF を認証するために，閾値を下げると，クローンを誤認証する可能性が高くなる．

Arbiter PUF の遅延時間差はチャレンジに対して線形で表現可能であることが 2004 年に Lim によって指摘されている [28, 29]．この論文では，Arbiter PUF で発生する遅延時間差が 3 つの要素から構成されており，平均が 0 で分散が σ のガウス分布に従うことを指摘し，チャレンジと遅延時間差が線形関係であること用いてチャレンジから遅延時間差を推定する手法を提案した．Rührmair らは Lim らの攻撃に使用した関係式に基づき，より扱いやすいモデリング式を提案した [50, 51]．Rührmair らはモデリング式を用いて，Arbiter PUF，*n*-XOR PUF，FF-PUF の攻撃を行い，*n*-XOR PUF では n の値に対して指数的に攻撃困難性が上昇することを実証した．

1.2.4 Arbiter PUF とノイズ

再現性の低い PUF は，認証にとって悪い影響を与える一方で，真性乱数生成器への利用が期待できる．真性乱数生成器は PUF と似た回路構成をしているセキュリティプリミティブで

ある．真性乱数生成器には，動作のたびに異なる出力をし，次に取得する出力が予測困難な性質が求められる．

2015 年に，山本らはノイズが混ざった Confined PUF のレスポンスを多値レスポンスとして利用する方式を提案している [65]. 具体的には安定しているレスポンスと不安定なレスポンスを分類し，安定なレスポンス (0, 1) はそれぞれ 00, 11, 不安定なレスポンスは 10 を出力する回路を設計した．Confined PUF は回路規模が小さいことも利点であるため，不安定なレスポンスを破棄せずに有効に秘密鍵生成に用いることはインパクトが大きい．

2018 年に Danger らは 1024 個の Confined PUF であるラッチ PUF を IC チップに実装し，得られるレスポンスの特性を調査した [12]．その結果，PUF と真性乱数生成器を実現させる理想的な条件が同じであることを報告している．実装された PUF は NOR ゲートを用いた Set-Reset (SR) ラッチによって構成されている．ラッチ PUF は初期状態として High が入力されており，動作時には同時に SR の両方へ Low が入力される．つまり Low 信号がラッチ PUF のチャレンジとなる．チャレンジを入力した際，ラッチ PUF のレスポンスは 0/1/振動状態 (Metastable) のいずれかになる．安定して出力された 0/1 は PUF のレスポンスとして利用でき，全ての PUF のレスポンスを XOR をとると真性乱数生成器として利用が可能となる．実装された 1024 個のラッチ PUF には，閾値電圧を変更するために NOR ゲートのボディバイアスを制御する回路と入力のタイミングを遅延させる回路が実装されており，レスポンスを制御することができる．遅延制御によって，Danger らはラッチ PUF のレスポンスが 0 から振動状態，そして 1 に推移することを報告している．振動状態はレスポンスが安定せず，どのレスポンスが出力されるか予測がつかない．そのため，振動状態のラッチが多ければ多いほど XOR をとった後の値が予測困難になるため，真性乱数器として性能が向上する．PUF として使用する時には，レスポンスの推定がされないようにパターンが存在しないことが理想的である．彼

らは結論として，閾値電圧や入力の制御設定について，PUF のレスポンスの偏りが少なく，つまり一様性が高くなる設定とレスポンスが振動状態の PUF が多くなるときの設定が同じになることを報告している．ただし，この時レスポンスが振動状態の PUF が多いということはレスポンスの値が不安定になるため再現性が理想から離れる．再現性を高くするためには安定して 0/1 を出力する PUF を増やせば良いが，最も再現性が高くなる状態はレスポンスがほとんど 0 または 1 になり，一様性が低い状態になる．つまり，1 つのラッチ PUF に注目した時，再現性を向上させると一様性が低下し，一様性を向上させると再現性が低下するトレードオフ関係がある．このようなトレードオフ関係は Extensive PUF のひとつである Arbiter PUF でも見られる．Arbiter PUF は遅延時間差を基にレスポンスを出力する PUF である．この遅延時間差は平均が 0 で分散が σ のガウス分布に従うことが知られている [28, 29]．再現性が理想値，つまり環境ノイズの影響を全く受けないようにするには，ラッチ PUF と同様に付加電圧等でレスポンスをほとんど 0 または 1 になる状態にすると良い．ただし，それでは偏りが大きくパターンが生じ，一様性が低くなる．レスポンスの偏りをなくすには，レスポンスを決定する閾値を 0 に近い値に設定するのが最適である．しかし，前述のとおり遅延時間差はガウス分布に従うため，0 に近い遅延時間差をもつ CRP が多い．つまり，環境ノイズの影響を受ける CRP が多くなる．全ての性能指標が高いことが理想だが，理想的な PUF を作製することは難しいため，利用するシステムによって各性質を理想値に近づけつつ優先順位を決める必要がある．

PUF の再現性はクローン作製にも影響を与える．Extensive PUF のひとつである *n*-XOR PUF [59] は数学的複製困難性を向上させるために提案された PUF である．*n*-XOR PUF は *n* 個の Arbiter PUF を並列に並べ，最後の出力を XOR させることによって，チャレンジから遅延時間差を線形表現することを困難とし，また Arbiter PUF に与えられるノイズの影響を大きくなることから数学的複製困難性が向上するとされている．一方，レスポンスに含まれるノイ

ズは攻撃に利用可能であることが報告されている [13]．Delvaux らは，再現性の低い CRP に含まれる情報を攻撃に利用する信頼性攻撃を提案した．具体的には，レスポンスの再現性 (以下信頼性と呼ぶ) は，遅延時間差の値と近似できることを指摘した．つまり，同じチャレンジに対するレスポンスを複数回取得し，その信頼性を学習することで 0 か 1 の 2 値しかないレスポンスよりも遅延時間差に近い値を学習に利用可能なため，クローンの作製が容易となる．また，Becker [7] は信頼性攻撃に進化的戦略 (Evolution Strategy, ES) を用いることで n-XOR PUF の数学的複製困難性は n の数に対して，指数的関数から線形的関数にまで低下することを明らかにした．数学的困難性が線形的関数まで低下する理由としては，信頼性攻撃では攻撃対象を n-XOR PUF として捉えるのではなく，n 個の Arbiter PUF として捉えることが可能だからである．具体的には，信頼性の低い CRP が n-XOR PUF に与える影響は，1 つの Arbiter PUF から発生した影響として考えることができ，確率的に不安定な CRP を発生させる Arbiter PUF を推定できるとしている．

1.2.5 深層学習が Extensive PUF に与える脅威

モデリング攻撃に使用されてきた機械学習技術は発展しており，PUF への脅威が増大している．2016 年に特により強力な機械学習の方法として深層学習を用いたモデリング攻撃が有効だと報告されている [67]．近年では，PUF に対する攻撃に深層学習を用いることが多くなっている．単純な構造である Arbiter PUF はモデリングにより表せるため，サポートベクターマシン (Support Vector Machine, SVM) やロジスティック回帰 (Logistic Regression, LR) を用いた攻撃が行われてきた．その対策として，PUF を複雑な構造とし，入出力の関係性を線形関係で表せなくする手法が提案されてきた．しかし，深層学習によってさらに細かい分類が可能となり，これまで安全とされていた複雑な構造をした PUF に対してさえも攻撃が可能であると

する報告がある [6, 23, 67, 74].

深層学習と根本的に同じであるニューラルネットワークを使用した PUF への攻撃は，2016 年以前にも報告されている [52]．ニューラルネットワークは入力層と隠れ層，そして出力層からなる学習器である．論文 [52] では SVM，LR，ニューラルネットおよび ES を PUF の安全性評価に利用し，LR と ES が有効だったことが述べられているが，ニューラルネットに関しては明らかにされていない．

一般的に，隠れ層が 3 層以上であるニューラルネットワークを深層学習と呼ぶ．隠れ層を増やすことにより，より複雑な表現を可能とするネットワークが作製できるため，複雑な分類問題を解くことができるとされている．一方で，深層学習は学習の結果得られる学習モデルが複雑であるため，学習結果がブラックボックスとなってしまうことが問題としてあげられている．そのため，作製されたモデルの分析は難しいが，より強力な攻撃ツールとして深層学習がクローニング攻撃へ利用されている．

1.3 研究目的

本研究の目的は，PUF の性質と攻撃シナリオについて整理し，異なる PUF や認証方式に対して深層学習を用いて安全性評価を行うことである．そのために，まずは PUF の性能ごとに安全性評価を行う条件となる攻撃シナリオを明確化する．PUF はユニーク性，ランダム性，耐タンパー性，物理的複製困難性，数学的複製困難性といった性質を有する．PUF の使用目的の想定およびそれぞれの性質に対して優先順位の設定が必要である．

本研究では，5 つある性質のうち，数学的複製困難性に注目する．数学的複製困難性は，近年の攻撃手法の発展により評価結果が大きく変遷している．例えば，近年盛んに行われている深層学習を用いた攻撃研究では，これまで数学的複製困難性が高いとされていた PUF が深層

学習を用いることにより容易に攻撃可能であることが報告されている．ただし，これまで数学的複製困難性に注目している論文では攻撃可能か否かを中心に議論している研究が多い．つまり，攻撃に使用する CRP をどうやって入手するのかなどの攻撃シナリオや数学的複製困難性と他の安全性の関係性などを議論している論文は少ない．そこで数学的複製困難性が低くなった PUF は，実際の認証システムでは使用できないのか，安全性はどこまで担保可能かといった議論の必要がある．

1.4 本研究の貢献

本論文の貢献は以下の 3 点である．

- Extensive PUF に対する攻撃シナリオの明確化
- 安全性評価環境の構築
- 深層学習を用いた安全性評価に対する PUF 実装方法による違いの明示

■Extensive PUF に対する攻撃シナリオの明確化　Extensive PUF はモデリング攻撃によってクローンが作製可能なことが知られている．攻撃者が作製するクローンは，攻撃者の能力や PUF の性質によって精度や攻撃シナリオや攻撃コストに違いが生じる．そこで本論文では攻撃者にとって有利になりやすい状況について整理し，それに対する攻撃コストの違いについて明らかにする．

■安全性評価環境の構築　近年，Extensive PUF に対する安全性評価に深層学習を用いる手法が報告されている．しかし，それぞれの手法はパラメータ等が独自に設定されている．そこで本論文では深層学習のライブラリおよび活性化関数について調査し，有効な PUF の安全性評

価環境を構築する．

■深層学習を用いた安全性評価に対する PUF 実装方法による違いの明示　PUF の性質によって，Extensive PUF のクローンを作製するための攻撃コストは変化する．そこで本論文では実装方法の異なる PUF に対して安全性評価を行う．実装方法の異なる PUF は，当然 PUF の性質が異なるため攻撃コストが異なる．既存の PUF の構成は変更せずに，安全性評価を改善する手法について検討を行う．

1.5　論文の構成

本論文の構成を図 1.2 に示す．第 2 章では，先行研究である PUF や深層学習について説明する．第 3 章では，数学的複製困難性に対して PUF の性質や攻撃シナリオを整理し，それぞれの条件化において数学的複製困難性がどうなるかを明確にする．第 4 章では，本論文で行う安全性評価の環境構築について説明する．第 5 章では，1 つのハードウェアインスタンスに対して時系列処理を行った PUF に対して安全性評価を行う．第 6 章では，1 つの回路を並列実装させ数学的複製困難性を改善させる PUF に対して安全性評価を行う．第 7 章では，認証者が認証に用いるレスポンスに意図的なエラーを注入した PUF に対して安全性評価を行う．第 8 章では，本論文をまとめ，今後の展望を述べる．

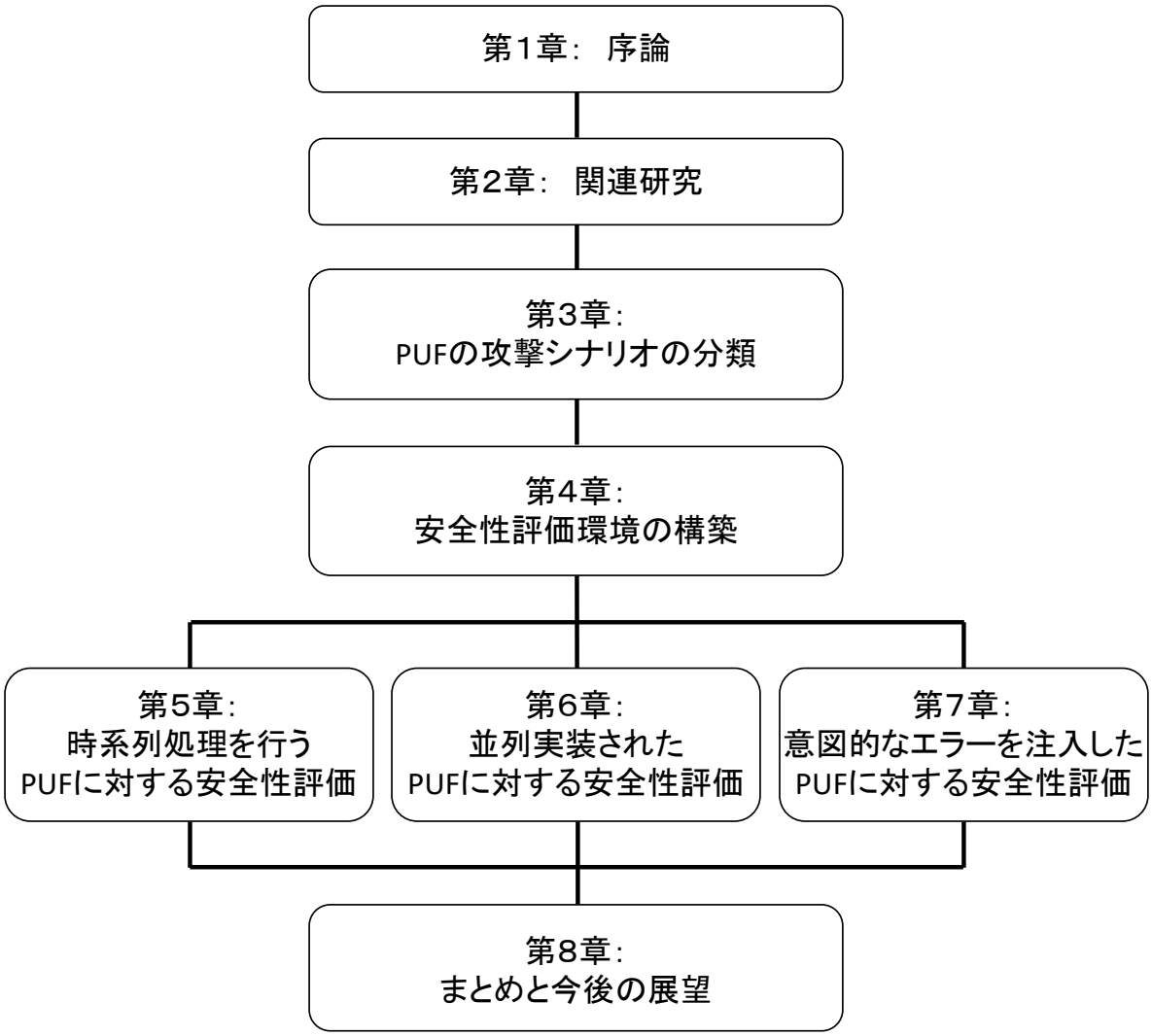
第1章: 序論
第2章: 関連研究
第3章:
PUFの攻撃シナリオの分類
第4章:
安全性評価環境の構築
第5章:
時系列処理を行う
PUFに対する安全性評価
第6章:
並列実装された
PUFに対する安全性評価
第7章:
意図的なエラーを注入した
PUFに対する安全性評価
第8章:
まとめと今後の展望

図. 1.2　本論文の構成

第 2 章

関連研究

2.1 Extensive PUF

Physically Unclonable Function (PUF) [46, 47] は半導体製造時に生じる物理的特徴のばらつきを用いて出力 (レスポンス) を得る技術である．PUF には大きく分けて 2 種類あり，入力可能なチャレンジの数が少ない PUF を Confined PUF，チャレンジの数が多い PUF を Extensive PUF と呼ぶ.

2.1.1 Arbiter PUF

Extensive PUF の代表例として，Arbiter PUF [16] がある．図 2.1 に示すように，Arbiter PUF はセレクタと Arbiter 回路から構成される．同じ入力を与えられる 2 つのセレクタをペアとし，上下で同じ配線となるように n 個のペアから (n 段) 構成される．Arbiter PUF は同じ構成の 2 つの信号経路の伝搬時間が配線長の違いや閾値電圧の違いなどによって異なることを利用し，信号伝搬時間によってレスポンスを決定する．Arbiter PUF のチャレンジは n ビットのバ

イナリ (C={C_1,$C_2$$\cdots$,$C_n$}) で，図のように 1 ビットずつセレクタのペアに入力される．チャレンジビットはセレクトペアに入力され信号伝搬経路の決定に利用される．チャレンジビットの値が 0 の場合には信号伝搬経路は直進，1 の場合には交差となる．チャレンジによって経路が決定されたのちに，立ち上がり信号が入力される．立ち上がり信号はチャレンジによって決定された経路を伝搬していくが，この時，配線長やゲート電圧の違いにより，同じ構成をした伝搬経路を流れる信号でも遅延時間差が生じる．その遅延時間差を基にどちらの信号が早く伝搬されたかを Arbiter 回路，Set-Reset (SR) ラッチなどによって判定し，0 か 1 を出力する．

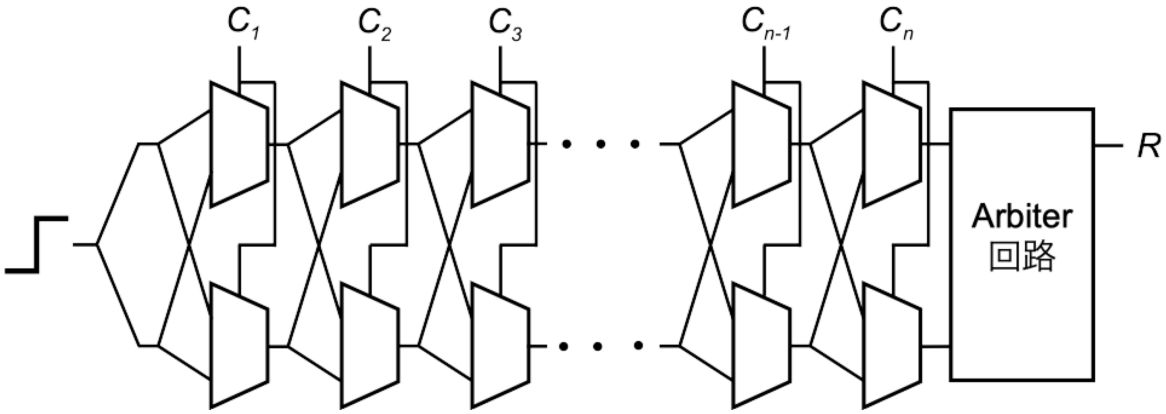

図. 2.1　Arbiter PUF の回路図

2.1.2　*n*-XOR PUF

n-XOR PUF はモデリング攻撃への対策として提案された PUF [59] である．図 2.2 に *n*-XOR PUF の回路図を示す．*n*-XOR PUF は Arbiter PUF を *n* 個並行に実装し，得られる *n* ビットのレスポンスを XOR することで 1 ビットの出力を得る PUF である．最終的にレスポンスを XOR するため，Arbiter PUF に比べて環境ノイズによりビット反転を起こしたレスポンスが多くなる．そのため，*n* の数に従って指数関数的にモデリング攻撃が困難になると報告され

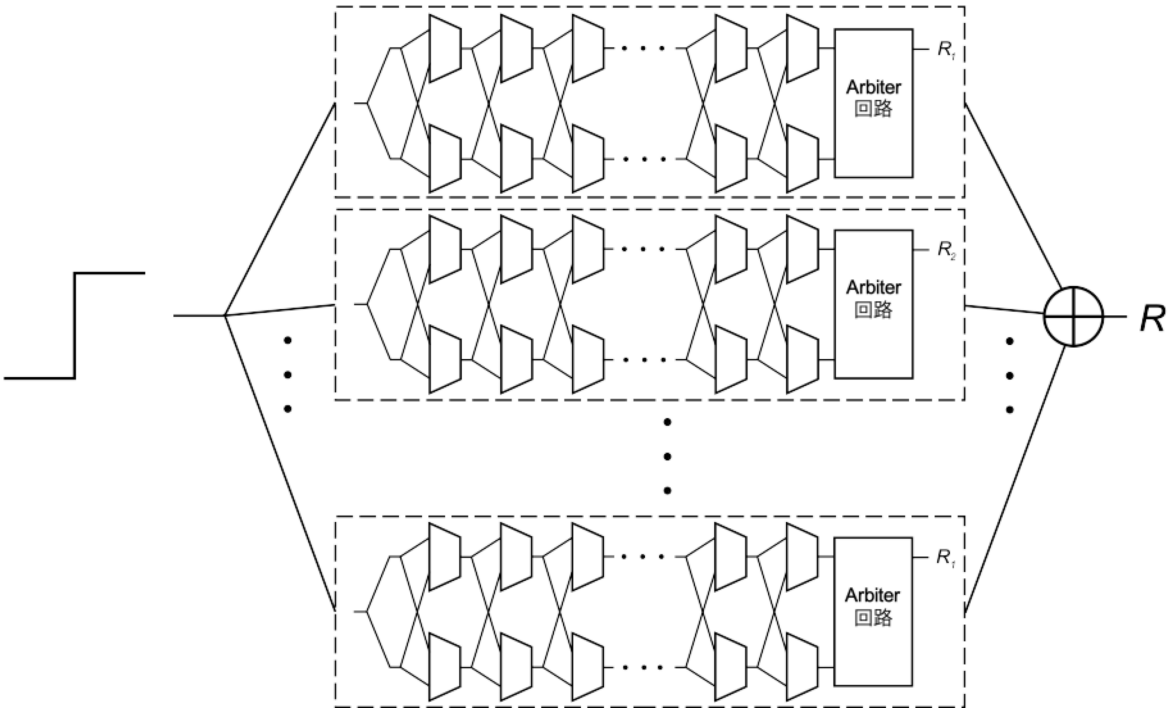

図. 2.2　*n*-XOR PUF の回路図

ている．

2.1.3　Double Arbiter PUF

図 2.3 に Double Arbiter PUF (DAPUF) の回路図を示す．DAPUF [32, 33, 34] は Field-Programmable Gate Array (FPGA) 上に実装した Arbiter PUF のユニーク性が低いことを改善するために提案された PUF である．DAPUF は *n*-XOR PUF と構造は似ているが，Arbiter 回路で計測する遅延時間差の扱い方が Arbiter PUF や *n*-XOR PUF と大きく異なっている．FPGA はすでに実装されている配線やブロックから回路に必要なパーツを選択することで，構成の異なる回路を何度も実装することが可能である．しかし，すでに実装されている配線やブロックを使うことから配置に対する制約が多い．特に Arbiter PUF の実装において配線長を揃えるこ

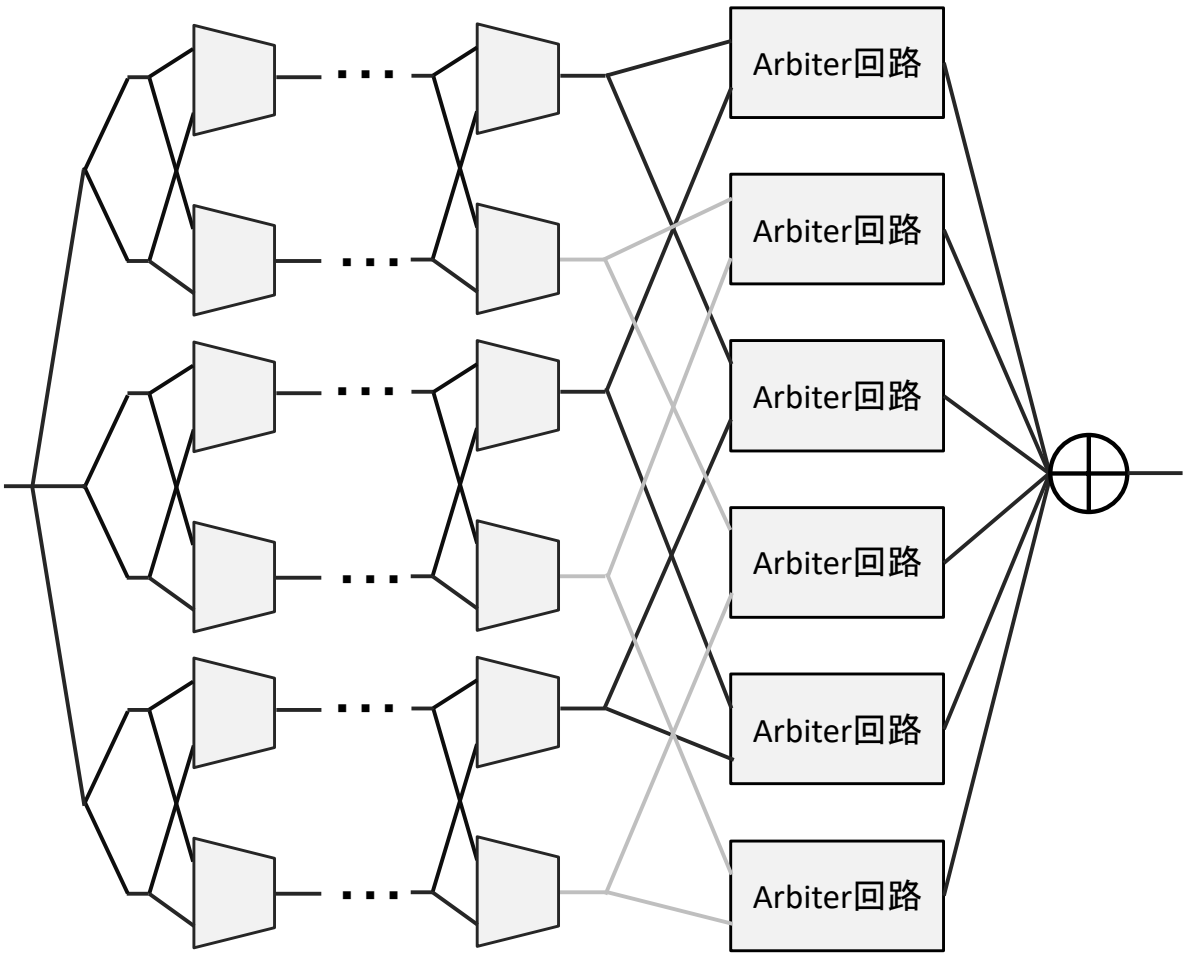

図. 2.3　Double Arbtier PUF の回路図

とが難しい点から配線長の違いの影響を受けやすく，複数の Arbiter PUF を実装した時に同じチャレンジを入力すると同じレスポンスが出力されやすい．Arbiter PUF において，特定の箇所で大きな遅延時間差が発生した場合，他の箇所で発生した遅延時間差の影響を打ち消してしまいレスポンスの値に直接反映される．特に 2-XOR PUF を FPGA ボードに実装した時，その影響は顕著に現れる．各 Arbiter PUF でチャレンジに対して同じレスポンスが出やすい傾向があると，2-XOR PUF ではほとんどのレスポンスが 0 になる．DAPUF は Arbiter PUF の内の同じ伝搬経路の信号同士に生じる遅延時間差を計測するためレスポンスの偏りを改善するこ

とが可能である．

2.1.4 RG-DTM PUF

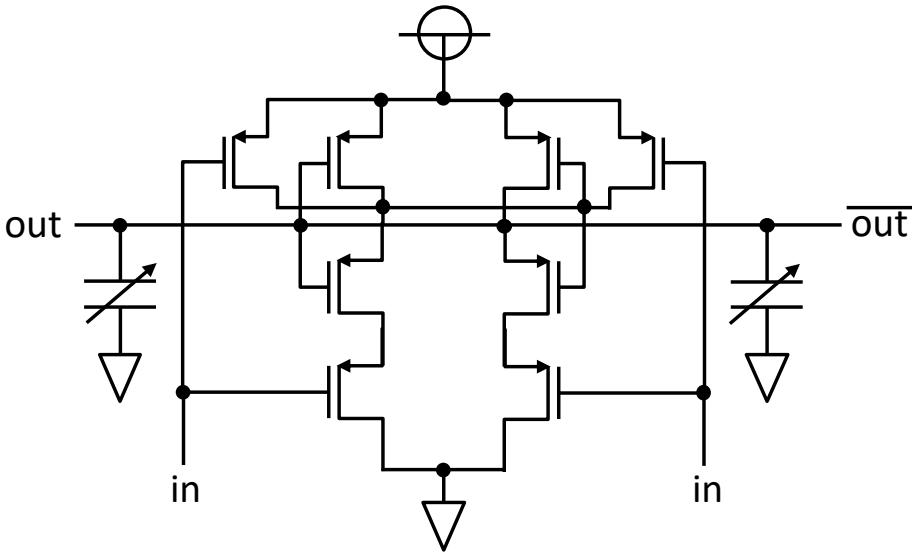

図. 2.4 RG-DTM PUF の Arbiter 回路

Response Generate Delay Time Measurement PUF (RG-DTM PUF) は Arbiter PUF のユニーク性を向上させるために提案された PUF である [14]. Arbiter PUF と基本的には同じ構成だが，図 2.4 のような Arbiter 回路を有している．この Arbiter 回路は回路容量を変更可能な回路であり，Arbiter PUF のレスポンス 0 と 1 を決定する閾値を変更することが可能である．通常の Arbiter 回路ではある遅延時間差に対して 1 つだけ閾値が存在し，その閾値以上であれば 1，以下であれば 0 というようにレスポンスを決定する．一方，RG-DTM PUF では Arbiter 回路の回路容量を変更することで，閾値を変更することが可能となり，図 2.5 に示すように複雑な閾値を設定することが可能である．

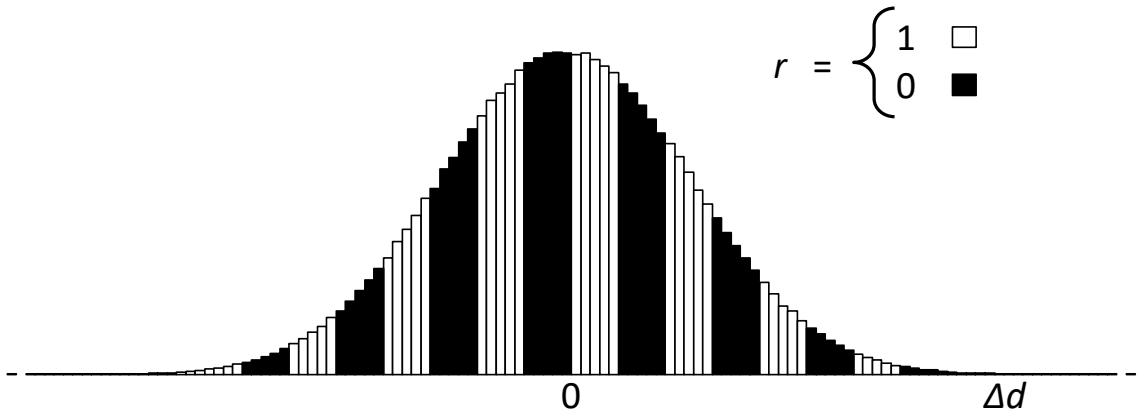

図. 2.5　RG-DTM PUF のレスポンス (16 分割)

2.2　深層学習を用いた攻撃

2.2.1　Arbiter PUF のモデリング

Arbiter PUF の遅延時間差は数式によってモデル化することが可能であると報告されている [28, 50]. モデリング式は次のようになる.

x 段 Arbiter PUF の入力である x ビットのチャレンジの立ち上がり信号の入力側から l 番目のチャレンジビットを $c_l \in \{0, 1\}$ とする．l 段目で生じる遅延時間差は，$c_l = 0$ の時 δ_l^0，$c_l = 1$ の時 δ_l^1 とする．伝搬経路を表すパリティベクトル $(\vec{\Phi})$ は

$$\vec{\Phi}(\vec{C}) = (\Phi^1(\vec{C}), \ldots, \Phi^x(\vec{C}), \Phi^{x+1}(\vec{C}))^T \tag{2.1}$$

と表される．ここで $\Phi^l(\vec{C})$ は

$$\Phi^l(\vec{C}) = \begin{cases} \prod_{i=l}^{x}(1 - 2c_i) & (l = 1, \ldots, x) \\ 1 & (l = x + 1) \end{cases} \tag{2.2}$$

である．各パリティベクトルに対応する遅延時間差 ($\vec{w}$) を

$$\vec{w} = (w^1, w^2, \ldots, w^x, w^{x+1})^T \tag{2.3}$$

と定義する場合，w^i は

$$w^i = \begin{cases} (\delta_1^0 - \delta_1^1)/2 & (i = 1) \\ (\delta_{i-1}^0 + \delta_{i-1}^1 + \delta_i^0 - \delta_i^1)/2 & (i = 2, \ldots, x) \\ (\delta_x^0 - \delta_x^1)/2 & (i = x + 1) \end{cases} \tag{2.4}$$

となる. 全ての段のセレクタペアで発生した遅延時間差 (Δ) は

$$\Delta = \vec{w}^T \vec{\Phi}. \tag{2.5}$$

と表すことができる． 結果として, Arbiter PUF のレスポンス (r) は

$$r = \mathrm{sgn}(\Delta) = \begin{cases} 0 & (\Delta \leq 0) \\ 1 & (\Delta > 0) \end{cases} \tag{2.6}$$

となる．

2.2.2 深層学習と機械学習の違い

深層学習は，ニューラルネットワークを拡張させた構造を有する学習器である．ニューラルネットワークの歴史は 1943 年に形式ニューロン [39] という数理モデルが提案されたことから始まる．図 2.6 に形式ニューロンおよび単純パーセプトロンと多層パーセプトロン (ニューラルネットワーク) の例を示す．形式ニューロンは脳の神経細胞を模倣することで高度な計算処理を有する可能性があることから提案された．形式ニューロンの仕組みは入力に対して重みをかけ，その値が閾値以上になった時に 1 を出力する．図を例とすると，x_1, x_2, x_3 という入力があった時，それぞれに重み w_1, w_2, w_3 が乗算される．得られた 3 つの値の加算結果が閾値 Θ

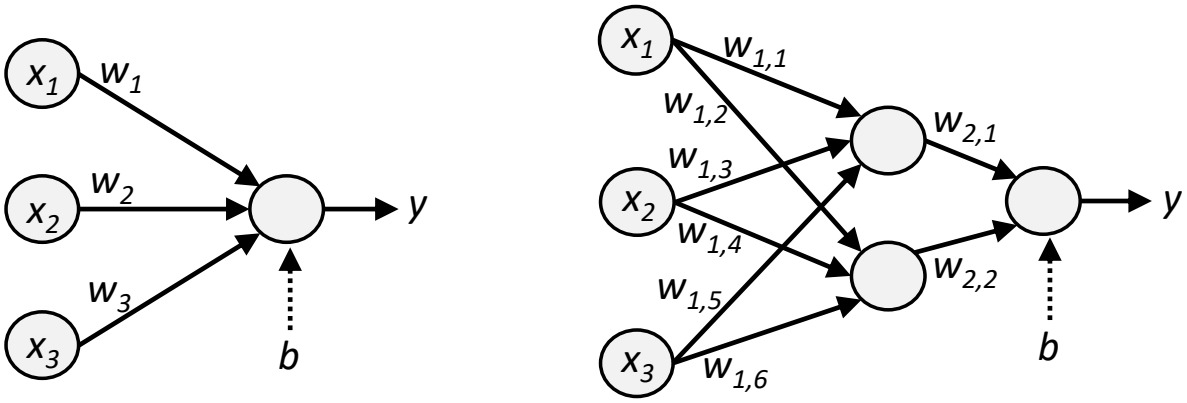

図. 2.6　単純パーセプトロンと多層パーセプトロン

以下の場合には y は 0 となり，Θ より大きい場合に y は 1 となる．つまり，これまでの説明を式化すると y について以下のようになる．

$$y = \begin{cases} 0 & (x_1 w_1 + x_2 w_2 + x_3 w_3 \leq \Theta) \\ 1 & (x_1 w_1 + x_2 w_2 + x_3 w_3 > \Theta) \end{cases} \tag{2.7}$$

形式ニューロンを基に 1958 年に提案されたパーセプトロン [49] と呼ばれるアルゴリズムが提案された．まず最も簡単なパーセプトロンの構造である単純パーセプトロンは．形式ニューロンと同様の構造をしている．ただし，単純パーセプトロンには，y が 0 または 1 を出力しやすくするためのバイアス (b) が加えられる．そのため，y について

$$y = \begin{cases} 0 & (x_1 w_1 + x_2 w_2 + x_3 w_3 + b \leq \Theta) \\ 1 & (x_1 w_1 + x_2 w_2 + x_3 w_3 + b > \Theta) \end{cases} \tag{2.8}$$

となる．そして形式ニューロンとパーセプトロンの最も大きな違いとしては，パーセプトロンでは入力データとして $X = \{x_1, x_2, x_3\}$ と y が与えられ，重みである $W = \{w_1, w_2, w_3\}$ を調整する学習できる点があげられる．単純パーセプトロンは，形式ニューロンを複数並列にすることが可能なため，より複雑な学習モデルが構築可能であると期待されていた．しかし，式 (2.8) からもわかるように単純パーセプトロンは線形表現しかできない [40]．線形表現以外のより複

雑な表現を可能とする手法として多層パーセプトロンがある．単純パーセプトロンは構造上何層も重ねることが可能である．この多層にしたパーセプトロンを後々にニューラルネットワークと呼ぶことになる．単層パーセプトロンと同様に図中の多層パーセプトロンを式化すると以下のようになる．

$$y = \begin{cases} 0 & (d_1 w_{2,1} + d_2 w_{2,2} + b \leq \Theta) \\ 1 & (d_1 w_{2,1} + d_2 w_{2,2} + b > \Theta) \end{cases} \tag{2.9}$$

ここで，

$$d_1 = x_1 w_{1,1} + x_2 w_{1,3} + x_3 w_{1,5} \qquad d_2 = x_1 w_{1,2} + x_2 w_{1,4} + x_3 w_{1,6} \tag{2.10}$$

である．

多層パーセプトロンでは，何層にも層を重ねることが可能だが，解けない問題が多かった．これは式 (2.9)，(2.10) からわかるようにパーセプトロンは線形式で表すため，線形問題しか解くことができない．そこで提案されたのが誤差逆伝播法 (バックプロパゲーション) である [53]．誤差逆伝播法を用いるにあたり，出力の値 (y) を求めるのにシグモイド関数を用いるようになり，0 から 1 の連続値を扱えるようになった．

ニューラルネットを深層にする手法は，1980 年代に提案されていたが，過学習などの問題が起こりやすかった．これは多層になればなるほど，重み w_i を学習することが困難になるためである．重みの数が増えると，重みの最適化が難しくなったり，初期値への依存が大きくなったりした．ここに誤差逆伝播法を用いることで，最適な重みを見つけやすくなる．誤差逆伝播法では誤差関数を偏微分し勾配降下法を用いることで，最適な値へ収束する重みの更新を容易にする．

第 3 章

PUF の攻撃シナリオの分類

3.1　PUF への攻撃シナリオ

PUF に対する想定される攻撃者の能力と攻撃シナリオについて検討する．まず，Confined PUF と Extensive PUF では攻撃に対するシナリオの考え方が基本的に異なる．Confined PUF はチャレンジ空間が小さいことから，攻撃者が全てのレスポンスを 1 回取得するだけで攻撃シナリオが成立する．そのため，攻撃者に情報を取得させないためにレスポンスを出力しない設計を行うことが一般的である．Extensive PUF はチャレンジ空間が大きいことから，利用方法に同じチャレンジを再利用する場合とワンタイムパッドのようにチャレンジを使い捨てする場合の 2 種類がある．チャレンジを再利用する場合は Confined PUF と同様にレスポンスを 1 回取得すると攻撃が成立するが，チャレンジを使い捨てする場合では未知のチャレンジに対するレスポンス予測が必要となる．

今回，第 1.2.3 章で説明したような攻撃手法を想定し，前提条件として以下の 4 点を想定する．

- 攻撃対象の PUF は Extensive PUF
- 攻撃者は CRP を取得可能
- 攻撃者は事前にクローンを作製する
- 攻撃時，PUF は攻撃者の手元にはない

まず，攻撃対象の PUF は Extensive PUF とする．攻撃時に実際の PUF は攻撃者の手元になく，事前にクローンを作製しておき，チャレンジに対するレスポンスを予測することを想定する．また，これから想定する攻撃シナリオでは，攻撃者は複数の CRP を取得可能とする．チャレンジとレスポンスの対応が不明な場合，クローニング攻撃が困難になる．そのため，マスターチャレンジと呼ばれる 1 個のチャレンジを渡し，各 Arbiter PUF にビットシフトなどを行い，異なるチャレンジを Arbiter PUF に与えた方が認証プリミティブとしては有用である．攻撃者にとっては，通信路に流れるマスターチャレンジしか取得できず，チップ上で行われるチャレンジ変換の情報を取得するには物理攻撃など追加の攻撃能力が必要になるためである．また同様に，レスポンスを攻撃者が取得できない場合，攻撃不可能となるため今回は考慮しない．しかし，安全性評価としては攻撃者に有利な攻撃シナリオの想定を行う方が有意のため，チャレンジとレスポンスの対応を攻撃者が知ることができるシナリオを想定する．

本章では次の 3 つに基づいて攻撃シナリオの分類を行う．

1. 攻撃者の測定能力
2. 環境ノイズの有無
3. PUF の遅延時間差パラメータ

3.2　攻撃者の測定能力に基づく攻撃シナリオの分類

攻撃者がクローンを作製するにあたり，最も影響を与えやすいのが攻撃者の測定能力である．攻撃者が高性能なクローンを作製するためには，攻撃対象の PUF に関する情報をより多く取得する必要がある．そこで攻撃者が取得可能な情報について，攻撃者の測定能力を基に整理し攻撃シナリオの分類を行う．

3.2.1　Arbiter PUF に対する攻撃シナリオ

初めに攻撃シナリオを想定する対象は，Extensive PUF の代表例である Arbiter PUF [16] とし，そして未知のチャレンジに対してレスポンスを予測する場合のみを想定する．

Arbiter PUF は，遅延時間差を基にレスポンスを決定する PUF である．遅延時間差は，対称となる回路の信号伝搬遅延時間の差分であり，各段で生じた遅延時間差の総和によってレスポンスが決定される．そのため，各段で生じる遅延時間差が攻撃対象の PUF と同じクローンを最も精度の高いクローンとする．

攻撃者の有する測定能力によって，クローニング攻撃に最も有効となる攻撃シナリオは異なる．そこで，攻撃者が測定できる情報によって条件を図 3.1 に示すように次の 3 つに分類した．

1. PUF 回路の内部の遅延時間が細かく測定可能である場合
2. Arbiter 回路に到達した際の遅延時間差を測定可能な場合
3. レスポンスを取得することができる場合

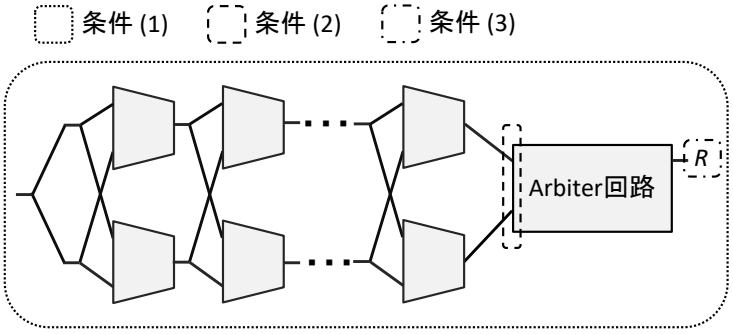

図. 3.1　攻撃者が測定可能な情報の分類

■条件 (1)–PUF 回路の内部の遅延時間が細かく測定可能である場合　攻撃者にとって最も有利な攻撃シナリオは条件 (1) である．配線上の伝播遅延やゲート遅延などを細かく測定できる場合，攻撃者は全ての遅延時間を 1 回測定し，それぞれの遅延をコピーすることでクローンを作製することができる．クローンを用いることにより，攻撃者はチャレンジからレスポンスを予測することが容易であり，クローンのもつパラメータの多さから精密なレスポンス予測が可能になる．例えばクローン作製後に認証者が付加電圧による遅延調整などをした場合でも，その情報が手に入れば攻撃者はクローンのそれぞれの遅延にパラメータを付加することによって対応が可能である．

■条件 (2)–攻撃者が Arbiter 回路に到達した際の遅延時間差を測定可能な場合　次に攻撃者が有利なシナリオは，条件 (2) である．攻撃者は取得した遅延時間差とチャレンジから線形計画法を解くことによって各段で生じる遅延時間差を得ることができ，クローンとして利用できる．この時得られるクローンは前述の条件と比較してパラメータ数が大幅に少なくなる．しかし，攻撃者がレスポンスを予測する際に取得が可能なのはチャレンジしかないので，各チャレンジビットに対する遅延時間差を取得できればレスポンス予測の精度は高い．

■条件 (3)–攻撃者がレスポンスを取得することができる場合　最後に，条件 (3) を想定する．ただし，レスポンスの取得方法にも複数の条件がある．攻撃者が複数レスポンスを取得できる場合 (条件 (3-a)) と 1 回しかレスポンスを取得できない場合 (条件 (3-b)) である．まず，条件 (3-a) の場合，レスポンスの信頼性を用いたサイドチャネル攻撃 [13] が可能になる．PUF の特徴の 1 つに，ある PUF に対して同じチャレンジを複数回入力すると出力するレスポンスは同じであるという特徴がある．信頼性はどのくらい安定して同じチャレンジを出力するかを表す指標で，どのくらい環境ノイズの影響を受けたか測定することができる．環境ノイズの影響は遅延時間差に影響すると考えられており，つまり信頼性を用いることでおおよその遅延時間差を推測することが可能になる．そこで信頼性の値とチャレンジから線形計画法を解くことにより各段で生じる遅延時間差の推定ができ，クローンとして利用できる．次に，条件 (3-b) の場合を想定する．レスポンスは遅延時間差から導出されるが，遅延時間差と比べて情報量が少ない．そのため，Rührmair ら [50] が提案したモデリング攻撃の手法などを用いることで，遅延時間差の推測は可能だが，精度はこれまでにあげた手法の中では低くなりやすい．

3.2.2　Arbiter PUF のバリアントに対する攻撃シナリオ

Arbiter PUF には様々なバリアントが存在する．今回は，バリアントとして *n*-XOR Arbiter PUF (*n*-XOR PUF) [59], *n*-1 Double Arbiter PUF (*n*-1 DAPUF) [33, 34], Response Generator according to Delay Time Measurement PUF (RG-DTM PUF) [14] を想定する．*n*-XOR PUF は *n* 個の Arbiter PUF から構成され，これらの Arbiter PUF から得られる *n* ビットのレスポンスを XOR することで 1 ビットのレスポンスを出力する PUF である．*n*-1 DAPUF は各段の構成は *n*-XOR PUF と同様だが，Arbiter 回路が同経路を伝搬した遅延時間の差分を測定する．2 経

路それぞれに対して同経路を伝搬した ${}_nC_2$ の遅延時間差を測定し，得られた ${}_nC_2 \times 2$ ビットを XOR することで 1 ビットのレスポンスを出力する．RG-DTM PUF は Arbiter PUF と同様の構成をしているが，Arbiter 回路に回路容量を変更するトランジスタが組み込まれており，回路容量の変更によってレスポンスのクラスタリングを複雑にする手法を用いた PUF である．これらの PUF に対する攻撃シナリオも Arbiter PUF と同様になることから第 3.2.1 章で用いた条件番号 (1)–(3) を用いる．

3.2.2.1 *n*-XOR PUF に対する攻撃シナリオ

まず，想定できる *n*-XOR PUF に対する攻撃シナリオについて攻撃者が取得できる情報量が多い順に説明する．

■条件 (1)–PUF 回路の内部の遅延時間が細かく測定可能である場合 Arbiter PUF と同様に，条件 (1) が攻撃者にとっても最も有利な攻撃シナリオになる．攻撃者は，全ての遅延時間を 1 回測定するだけで精度の高い *n*-XOR PUF のクローン作製が可能である．ただし，*n* の数により測定箇所が増えるため Arbiter PUF 単体を攻撃する時に比べて攻撃コストは増加する．

■条件 (2)–攻撃者が Arbiter 回路に到達した際の遅延時間差を測定可能な場合 条件 (2) が次に有利な攻撃シナリオとなる．遅延時間差をそれぞれ測定できる場合，Arbiter PUF の場合と同様に攻撃することが可能なため，同様のクローンの精度を保つことができる．ただし，条件 (2) の場合，*n* の数により測定箇所が増えたり，Arbiter PUF のクローンを *n* 個作製したりしなくてはならないため，Arbiter PUF 単体を攻撃する時に比べて攻撃コストは増加する．

■条件 (3)–攻撃者がレスポンスを取得することができる場合 次に条件 (3) について想定する．*n*-XOR PUF の場合，取得可能なレスポンスによって大きく 2 つに分けることができる．

まず初めのケースが XOR する前の Arbiter PUF のレスポンスが取得可能な場合である．この時，攻撃対象が n 個に増えるが，Arbiter PUF と同様に条件 (3-a) および条件 (3-b) を想定することができる．最後に想定されるのが，n-XOR PUF のレスポンスしか取得できない場合である．まず，レスポンスが複数回取得でき，n-XOR PUF の構成が既知である場合，前述のサイドチャネル攻撃 [13] を拡張した信頼性攻撃 [7] が可能になる．信頼性攻撃は n-XOR PUF の信頼性から同じような信頼性をもつ Arbiter PUF を n 個作製する攻撃手法である．この時，それぞれの Arbiter PUF の遅延時間差推定には非決定的なアルゴリズムを使用することで，異なる遅延時間差をもつ Arbiter PUF を作製する．

次に想定される条件として，レスポンスは 1 回しか取得できないが，n-XOR PUF の構成が既知である場合である．この時，Rührmair ら [50] が提案した n-XOR PUF のモデリング攻撃が可能になる．ただし，この手法では n の数に従い，学習時間が線形的に上昇することが問題点としてあげられるため，攻撃者が選択する攻撃手法としては現実的ではない．

最後に，レスポンスは 1 回しか取得できなく，n-XOR PUF の構成が未知の場合である．この時，攻撃者は得られる情報から攻撃対象の PUF が Arbiter PUF か n-XOR PUF かを判断することができない．しかし，Arbiter PUF のモデリング攻撃に深層学習を用いることで攻撃する手法が報告されている [55, 67].

3.2.2.2 n-1 DAPUF に対する攻撃シナリオ

n-1 DAPUF に対して想定できる攻撃シナリオについて説明する．DAPUF は基本的には n-XOR PUF と同じ構成をしているため，同様の攻撃シナリオの想定ができる．ただし，遅延時間差を推定する攻撃手法において，n-XOR PUF ではなく，$(2 \times {}_nC_2)$-XOR PUF と想定する必要がある点が異なる．条件 (3) に関して想定すると，DAPUF の構成は n-XOR PUF とほと

んど同じことから条件 (3-a) かつ PUF の構成が既知の場合，$(2\times{}_nC_2)$-XOR PUF と想定することで信頼性攻撃が可能である．最後に条件 (3-b) かつ，PUF の構成が未知の時，n=2，3，4 の時に n-1 DAPUF に対して Arbiter PUF のモデリング式と深層学習を用いた攻撃が報告されている [6, 23, 74].

3.2.2.3　RG-DTM PUF に対する攻撃シナリオ

最後に RG-DTM PUF に対する攻撃シナリオについて説明する．RG-DTM PUF はレスポンスのクラスタリングの数および閾値が既知であるかが攻撃シナリオに最も影響する．条件 (1) および条件 (2) に関して想定すると，クラスタリングの設定が既知の場合，RG-DTM PUF の構成は Arbiter PUF と同じのため，クローンの精度は Arbiter PUF と同様になる．クラスタリングの設定が未知の場合，各段の遅延時間差の予測精度が高いクローンは作製可能なため，通常出力であるレスポンスを複数集めて Arbiter 回路に到達した時の遅延時間差と比較することでクラスタリングの設定を推定可能である．この時，条件 (1) であっても複数の CRP を取得しなくてはならなくなるため攻撃コストが増加する．最後に，条件 (3) では，クラスタリングの閾値が既知であったとしても，攻撃者は得られたレスポンスがどの閾値の遅延時間差で揺らいでいるか特定が困難なため，クラスタリングの設定情報をうまく利用できない可能性が高い．ただし，分割数は汐崎ら [56] がモデリングに利用できることを報告している．そして条件 (3-b) の時，クラスタリングの設定および PUF の構成が未知であったとしても Arbiter PUF のモデリング式と深層学習を用いた攻撃が報告されている [66].

3.3 環境ノイズに基づく攻撃シナリオの分類

精度の高いクローンを作製したとしてもクローンと実際の PUF でレスポンスが異なる場合がある．その原因が環境ノイズである．環境ノイズの影響は，攻撃者が確率的に推定することはできても正確に予測することが困難である．一方で，認証者であっても同様に環境ノイズを正確に予測することはできないため，使用時には環境ノイズの影響を取り除いてから鍵生成や認証に利用するのが一般的である．

環境ノイズはクローンの性能や攻撃者が実際の PUF から取得する情報量に大きな影響を与える．例えば，取得した CRP に環境ノイズの影響を受けビットフリップしたレスポンスが含まれている場合，クローン作製にエラー訂正せずに CRP を利用するとクローンの性能が下がる [50, 63]．そのため，攻撃者は同じチャレンジのレスポンスを複数取得しエラー訂正する必要がある．

一方で攻撃者にとって優位になる攻撃シナリオも存在する．クローンは環境ノイズの影響を受けない決定的なレスポンスを出力する．そのため，クローンの予測誤りレスポンス数が環境ノイズの影響を受けるレスポンス数より少ない場合，認証通過やレスポンスのエラー訂正が可能となる．つまり，クローンが目指すレスポンス予測精度の最低ラインは環境ノイズの影響より低ければ良い．また，攻撃シナリオにあるように環境ノイズは攻撃に利用できることも報告されている [7, 13]．攻撃者は同じチャレンジのレスポンスを複数取得して信頼性を取得する必要があるため必要な情報量は増えるが，攻撃精度は高くなる．文献 [13] では，環境ノイズの影響を受けるレスポンスが多いほど攻撃に使用可能な CRP が増えるため攻撃が容易になると報告されている．これらのことから，環境ノイズの影響は PUF を利用したシステムを想定する時に，攻撃シナリオによっては攻撃者の優位性を高める要因となりうる．

3.4　PUF の遅延時間差パラメータに基づく攻撃シナリオの分類

3.4.1　Arbiter PUF の遅延時間差パラメータ

この節では Arbiter PUF がもつ遅延時間差パラメータと攻撃シナリオについて議論する．Arbiter PUF のもつ遅延時間差パラメータは l 段セレクタと Arbiter 回路の 2 つに大きく分けられる．そこで，それぞれの場合に焦点を当てて攻撃シナリオを議論する．また，攻撃に必要なパラメータ数としてチャレンジのビット数がある．通常攻撃には l ビットのチャレンジが必要になるが，これから議論する条件によってはレスポンス導出に必要なチャレンジビット数が少なくなる．そこで，各条件と共にチャレンジビット数が変化する場合についても議論する．なお，ここでは細かく遅延時間差を測定できる条件に関しては議論せず，攻撃者が推定に必要なパラメータ数のみ言及する．

3.4.1.1　l 段セレクタの遅延時間差パラメータに基づく攻撃シナリオの分類

まず，l 段セレクタの遅延時間差パラメータについて議論する．i 段目に注目した時の l 段セレクタの遅延時間差パラメータを図 3.2 に示す．i 段目より前の遅延時間差 $(\sum_{k=1}^{i-1} \delta_k)$ を Δ_F，i 段目より後の遅延時間差 $(\sum_{k=1}^{i-1} \delta_k)$ を Δ_B，i 段目の遅延時間差は δ_i とする．l 段セレクタの i 段目で発生した遅延時間差 δ_i は，チャレンジビットの値によって $\delta_{i,0}$ と $\delta_{i,1}$ が存在する．

通常，攻撃者は l 段セレクタに対して $2 \times l$ パラメータの推定が必要になる．そこで，攻撃者の推定するパラメータ数が $2 \times l$ より少なくなる条件について議論する．

■i 段目で発生した遅延時間差 ($\delta_{i,0}$ と $\delta_{i,1}$) が等しい場合　最初に $\delta_{i,0}$ と $\delta_{i,1}$ について考えると，$\delta_{i,0}$ と $\delta_{i,1}$ について，$\delta_{i,0} < \delta_{i,1}$，$\delta_{i,0} = \delta_{i,1}$，または $\delta_{i,0} > \delta_{i,1}$ という不等式が成り立つ．各段は

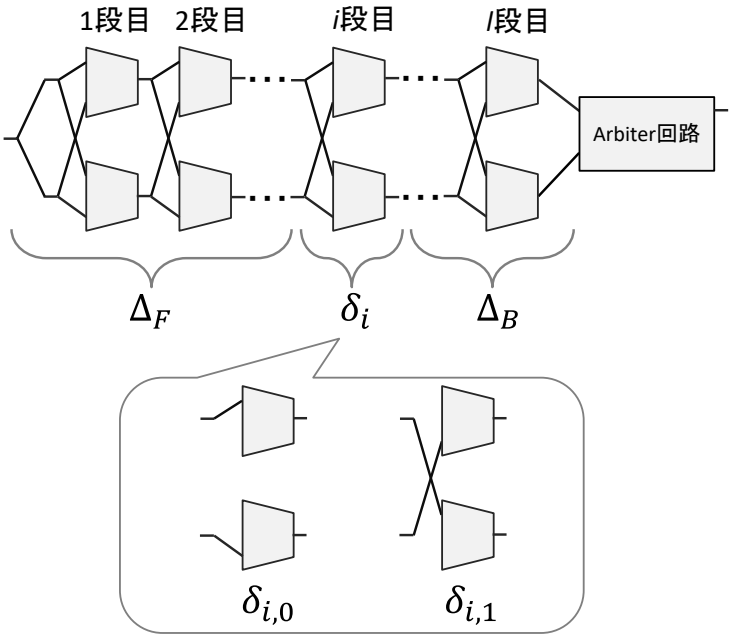

図. 3.2　i 段目に注目した際の l 段セレクタのパラメータ

いずれかの不等式が当てはまり，PUF の性質が決定されている．ただし，不等式の組み合わせによっては特殊な場合が存在する．全ての段で等式 ($\delta_{i,0} = \delta_{i,1}$) が成り立っている場合である．全ての段で等式が成り立つ時，チャレンジビットによる遅延時間の差が存在しない．攻撃者がクローン作製する時，従来は 2 パラメータ ($\delta_{i,0}$) および ($\delta_{i,1}$) を推定しなくてはならない．しかし，全ての段で等式 ($\delta_{i,0} = \delta_{i,1}$) が成り立っている時は，各段で 1 パラメータ (δ_i) を推定することで攻撃が可能となる．つまり，推定する遅延時間差パラメータが $2 \times l$ から l と少なくなる．

■i 段目の遅延時間差 (δ_i) が支配的な場合　i 段目の遅延時間差 (δ_i) が大きく，レスポンスの出力に与える影響が支配的な場合がある．i 段目の遅延時間差 (δ_i) が支配的な場合に，以下の 3 パターンが考えられる．

1. i 段目より前の遅延時間差 (Δ_F) が無視される場合

2. i 段目より後の遅延時間差 (Δ_B) が無視される場合
3. i 段目より前の遅延時間差 (Δ_F) および i 段目より後の遅延時間差 (Δ_B) が無視される場合

Δ_F が無視される場合　$\min(|\delta_{\mathrm{i},0}|, |\delta_{\mathrm{i},1}|) > \max(|\Delta_{\mathrm{F}}|)$ の場合，i 段目より前で発生した遅延時間差がレスポンスに与える影響はなくなる．つまり，l ビットのチャレンジを与えた時の遅延時間差の総和 ($\sum_{k=1}^{l} \delta_k$) と i 段目から l 段目で生じる遅延時間差の総和 ($\delta_i + \Delta_B$) について $\mathrm{sign}(\sum_{\mathrm{k}=1}^{\mathrm{l}} \delta_{\mathrm{k}}) = \mathrm{sign}(\delta_{\mathrm{i}} + \Delta_{\mathrm{B}})$ が成り立つ．この時，攻撃者は l 段全ての遅延時間差を推定する必要はなく，i 段目から l 段目で生じる遅延時間差を推定すればクローンを作製することができる．つまり，チャレンジビットによって遅延時間差が異なる場合では $2 \times l$ から $2 \times (l - (i - 1))$ に，同じ場合に l から $l - (i - 1)$ に推定する遅延時間差パラメータが少なくなる．この時，レスポンス導出に必要なチャレンジビット数も l から変化する．レスポンス導出にあたり，i 段目以前の遅延時間差はレスポンスに影響を与えないため，チャレンジビット数も l から $l - (i - 1)$ まで少なくなる．

Δ_B が無視される場合　$\min(|\delta_{\mathrm{i},0}|, |\delta_{\mathrm{i},1}|) > \max(|\Delta_{\mathrm{B}}|)$ の場合，i 段目より後に発生した遅延時間差がレスポンスに与える影響はなくなる．つまり，l ビットのチャレンジを与えた時の遅延時間差の総和 ($\sum_{k=1}^{l} \delta_k$) と 1 段目から i 段目で生じる遅延時間差の総和 ($\Delta_F + \delta_i$) について $\mathrm{sign}(\sum_{\mathrm{k}=1}^{\mathrm{l}} \delta_{\mathrm{k}}) = \mathrm{sign}(\Delta_{\mathrm{F}} + \delta_{\mathrm{i}})$ が成り立つ．この時，攻撃者は l 段全ての遅延時間差を推定する必要はなく，1 段目から i 段目で生じる遅延時間差を推定すればクローンを作製することができる．つまり，チャレンジビットによって遅延時間差が異なる場合では $2 \times l$ から $2 \times i$ に，同じ場合に l から i に推定するパラメータが少なくなる．ただし，レスポンス導出に必要なチャレンジビットは削減できない．

Δ_F および Δ_B が無視される場合　$\min(|\delta_{i,0}|, |\delta_{i,1}|) > \max(|\Delta_F|)$ かつ $\min(|\delta_{i,0}|, |\delta_{i,1}|) - \max(|\Delta_F|)$ が $i+1$ 段目以降の遅延時間差の総和がとりうる値の最大値 ($\max(|\Delta_B|)$ より大きな値であった場合，推定するパラメータはさらに少なくなる．攻撃者は i 段目の遅延時間差を推定すれば良いため，遅延時間差が異なる場合では 2，同じ場合では 1 パラメータ推定すれば攻撃が可能である．またレスポンス導出には，遅延時間差の正負を判定するために i 段目以降のチャレンジビットが必要となる．そのため，必要なチャレンジビットは $l-(i-1)$ となる．

■全ての段において遅延時間差 ($\delta_{i,0}$ と $\delta_{i,1}$) のどちらかが大きい場合　全ての段において，遅延時間差 ($\delta_{i,0}$ と $\delta_{i,1}$) のどちらかが非常に大きい場合，チャレンジビットがレスポンスに与える影響が大きくなる．特に影響が大きくなるのが，実装上の制約でセレクタペアの同じ箇所で配線長が長くなり，遅延時間差が大きくなる場合である．仮に，チャレンジビットが 1 の時には伝搬経路は交差し，チャレンジビットが 0 の時には伝搬経路は直進とする．全ての段で $\delta_{i,1} > \delta_{i,0}$ が成り立ち，$\min(|\delta_{i,1}|) > \sum_{k=1}^{l} |\delta_{k,0}|$ の場合，1 であるチャレンジビットが奇数ならレスポンスは 0，偶数なら 1 になる．反対に $\delta_{i,0} > \delta_{i,1}$ が成り立ち，$\min(|\delta_{i,0}|) > \sum_{k=1}^{l} |\delta_{k,1}|$ の場合にも，1 であるチャレンジビットが奇数ならレスポンスは 0，偶数なら 1 になる．すなわち，攻撃者はこの条件下では遅延時間差パラメータを推定する必要がなく，チャレンジビットのみでレスポンスを導出することができる．

■全ての段で生じる遅延時間差 (δ_i) が等しい場合　最後に，全ての段で生じる遅延時間差 (δ_i) が同じ状況である．この時の遅延時間差を，チャレンジビットが 0 の時に δ^0，チャレンジビットが 1 の時に δ^1 とする．まず 1 段で発生する遅延時間差がチャレンジビットにより異なる場合 ($\delta^0 \neq \delta^1$) について考える．チャレンジビットが 1 の時に生じる遅延時間差は，$\mathrm{mod}(\sum_{k=1}^{l} C_k, 2) \times \delta^1$ となる．チャレンジビットが 0 の時に生じる遅延時間差

は，$(l - \mathrm{mod}(\sum_{\mathrm{k=1}}^{\mathrm{l}} \mathrm{C_k}, 2)) \times \delta^0$ となる．これら 2 つの式をまとめると，全体の遅延時間差は $\mathrm{mod}(\sum_{\mathrm{k=1}}^{\mathrm{l}} \mathrm{C_k}, 2) \times \delta_{\mathrm{i},1} + (\mathrm{l} - \mathrm{mod}(\sum_{\mathrm{k=1}}^{\mathrm{l}} \mathrm{C_k}, 2)) \times \delta_{\mathrm{i},0}$ となる．この時，攻撃者はある 1 段で発生する遅延時間差 δ^1 と δ^0 の 2 パラメータを推定するだけでクローンを作製することができる．

そして，チャレンジビットによる遅延時間差が同じ場合には，Arbiter PUF はどんなチャレンジを与えたかにかかわらず同じレスポンスを出力する．この時の遅延時間差は $l \times \delta^0$，または $l \times \delta^1$ で導出できる．つまり，攻撃者は δ_0，または $l \times \delta^1$ の 1 パラメータを推定することでクローンを作製できる．また，レスポンスは 1 値しかないため，クローンを作製せずともレスポンスの予測が可能である．ただし，このような性質をもつ PUF は現実システムにおいて使用されるとは考えにくいため，攻撃シナリオに当てはまる機会が存在しない可能性が高い．

3.4.1.2 Arbiter 回路の遅延時間差パラメータに基づく攻撃シナリオの分類

Arbiter 回路で発生する遅延時間と攻撃シナリオについて議論する．ここで Arbiter 回路に到達した際の遅延時間差 $(\sum_{k=1}^{l} \delta_k)$ を Δ_s，Arbiter 回路で発生する遅延時間差を Δ_a とする．

Δ_s は 0 を平均にとるガウス分布に従うことが知られている．そのガウス分布に対して，レスポンスを決定するの閾値 (Δ_a) を設定する．Δ_s が Δ_a より大きい場合にレスポンスは 1 (または 0)，小さい場合には 0 (または 1) になる．また，Δ_a にはチャレンジが入力されることはなく，遅延時間差が決定的なため，レスポンスの決定に大きな影響を与えやすい．

Δ_a がとる状態は $\Delta_a > 0$，$\Delta_a < 0$，$\Delta_a = 0$ の 3 つがある．

■$\Delta_a = 0$ の場合　$\Delta_a = 0$ の場合，Δ_s の値がそのまま出力に影響する．つまり攻撃者に影響を与えるのは，第 3.4.1.1 章で説明した条件のみとなる．

■$\Delta_a > 0$ の場合　$\Delta_a > 0$ の場合には，$\min(|\delta_{\mathrm{i},0}|, |\delta_{\mathrm{i},1}|) > \max(|\Delta_{\mathrm{F}}|)$ に類似した $\max(|\Delta_{\mathrm{F}}|) < \min(|\delta_{\mathrm{i}} + \Delta_{\mathrm{B}}|)$ が成り立ちやすくなる．この式において Δ_a を考慮した場合，$\max(|\Delta_{\mathrm{F}}|) <$

$\min(|\delta_{\mathrm{i}} + \Delta_{\mathrm{B}}|) + \Delta_{\mathrm{a}}$ となる．すなわち，Δ_a の分，右辺の値が大きくなり，条件が成り立ちやすくなる．この条件化では攻撃者が Δ_F を無視できるため，推定する遅延時間差パラメータが $2\times(l-i)$ または $l-i$ まで少なくなる．また，レスポンス導出に必要なチャレンジビットも $l-i$ となる．

■$\Delta_a < 0$ の場合　$\Delta_a < 0$ の場合には，$\min(|\delta_{\mathrm{i},0}|, |\delta_{\mathrm{i},1}|) < \max(|\Delta_{\mathrm{F}}|)$ に類似した $\max(|\Delta_{\mathrm{F}} + \delta_{\mathrm{i}}|) > \min(|\Delta_{\mathrm{B}}|)$ が成り立ちやすくなる．この式において Δ_a を考慮した場合，$\max(|\Delta_{\mathrm{F}} + \delta_{\mathrm{i}}|) > \min(|\Delta_{\mathrm{B}}|) + \Delta_{\mathrm{a}}$ となる．この時，Δ_a は負の値のため，右辺の値が小さくなり，条件が成り立ちやすくなる．この条件化では攻撃者が Δ_B を無視できるため，推定する遅延時間差パラメータが $2\times i$ または i まで少なくなる．ただし，レスポンス導出に必要なチャレンジビットは削減できない．もしも，$\min(\Delta_{\mathrm{s}}) > \Delta_{\mathrm{a}}$ または，$\max(\Delta_{\mathrm{s}}) < \Delta_{\mathrm{a}}$ となる場合，レスポンスは 1 値しか出力されない．

3.4.2　Arbiter PUF のバリアントとパラメータ

第 3.4.1 章と同様に Arbiter PUF のバリアント (*n*-XOR PUF，*n*-1 DAPUF，RG-DTM PUF) について遅延時間差パラメータによる攻撃シナリオについて考察する．

3.4.2.1　*n*-XOR PUF のパラメータ

まず，全ての Arbiter PUF で $\Delta_a = 0$ とした時の *n*-XOR PUF のパラメータについて想定する．通常，*n*-XOR PUF は *n* 個の Arbiter PUF から構成されるため，攻撃者は $n\times(2\times l)$ パラメータを予測する必要がある．予測するパラメータ数が最も少なくなる場合は，*n* 個の Arbiter PUF 全てが同じ遅延時間差を有する場合である．もしも，*n* 個の Arbiter PUF 全てが同じ遅延時間差を有し *n* の数が偶数ならば，レスポンスは全て 0 になる．そのため，チャレンジに関わ

らず全てのレスポンスを 0 と予測することで，クローンが作製可能である．つまり，遅延時間のパラメータ推定ができなくてもレスポンスが予測できる．n が奇数の場合，$(n-1)$-XOR PUF のレスポンスは全て 0 となる，つまり n-XOR PUF を 1 つの Arbiter PUF として捉えることができるため，予測するパラメータ数は前項の Arbiter PUF のパラメータ数と同様になる．次に，n 個の Arbiter PUF のうち，m 個の Arbiter PUF が同じパラメータを有することを想定する．この時，推定が必要なパラメータ数は m が偶数であれば，最大で $(n-m)\times(2\times l)$，m が奇数であれば最大で $(n-(m-1))\times(2\times l)$ となる．もし，$n-m$ または $n-(m-1)$ 個の Arbiter PUF として攻撃するシナリオが有効な場合，前項の条件が当てはまっていれば，予測するパラメータ数を減らすことが可能である．

3.4.2.2 n-1 DAPUF のパラメータ

n-1 DAPUF について想定する．第 3.2.2.2 章で述べたように，n-1 DAPUF は，$(2\times {}_nC_2)$-XOR PUF と考えることができる．つまり予測が必要なパラメータ数は $2\times {}_nC_2\times(2\times l)$ となる．$2\times {}_nC_2$ は偶数であるため，n-XOR PUF の条件のうち，n が偶数の場合の条件に当てはまる．すなわち，$2\times {}_nC_2$ 個の Arbiter PUF 全てが同じ遅延時間差を有する場合にはレスポンスは全て 0 になり，$2\times {}_nC_2$ 個の Arbiter PUF のうち，m 個の Arbiter PUF が同じパラメータを有する場合には最大で $2\times(n-m)\times(2\times l)$，$m$ のパラメータ推定が必要となる．つまり，同じ遅延差をもつ Arbiter PUF が存在する場合，推定が必要なパラメータ数が減る可能性がある．

3.4.2.3 RG-DTM のパラメータ

RG-DTM PUF は Arbiter PUF と同様の構成をしているため，推定する l 段セレクタのパラメータ数は Arbiter PUF と同条件となる．Arbiter 回路の遅延時間パラメータ (Δ_a) は，RG-DTM PUF の場合，推定するパラメータ数が増える．これは，RG-DTM PUF では分割数によ

るクラスタリングをするためである．例えば 8 分割の時には Δ_a は，$\Delta_{a,1}$ から $\Delta_{a,7}$ まで値を変更する．つまり分割数によってパラメータ数が増大する．

第 4 章

安全性評価環境の構築

本章では，PUF の安全性評価に使用する深層学習のライブラリと深層学習のパラメータについて検討を行う．

4.1　深層学習のライブラリ

深層学習の実装は，一般的には実装を容易にするために既存のライブラリを使用することが多い．

表. 4.1 に本論文で安全性評価に利用した PC のスペックを示す．

表 4.1　安全性評価に利用する PC のスペック

OS	CPU	GPU
Windows 10 Pro	Intel(R) Core(TM) i9-9900K	GPU Nvidia GeForce RTX 2080 SUPER

今回検討したライブラリは，Pylearn2 [17], Keras [10], Pytorch [48] である．

Pylearn2 は，2016 年に PUF に対する深層学習を用いた安全性評価を初めて行った論文 [67]

で利用されたライブラリである．2019 年に，文献 [67] では攻撃ができないと報告されていた 3-1 DAPUF への攻撃が可能だと報告された [6]．文献 [6] で安全性評価に利用されていたのが Keras である．Pytorch は近年，画像認識分野で注目を集めているライブラリのひとつである．

表. 4.2 に 3 つのライブラリの特徴をまとめる．

表 4.2　深層学習の 3 つのライブラリの特徴

	Pylearn2	Keras	Pytorch
発表年	2013	2015	2019
使用言語		Python	
主な機械学習ライブラリ	Theano	Theano Tensorflow	Torch
記述	△	○	○
動作	○	○	×
パラメータ変更	○	△	–

■Pylearn2　Pylearn2 は深層学習で用いる各層の設定を yaml 形式で 1 層ずつ記述する．事前学習時には 1 層ごとに学習した結果を pkl 形式で保存し，その前の層の結果を読み込んで学習を行う．最後の層では事前学習で保存したすべての結果を読み込み，学習を行う．Pylearn2 の大きな特徴としては各層ごとにファイルを分けて記述を行う点である．そのため，学習モデルのパラメータを変更した際，途中の層から学習を行うことが容易である．また，同様の理由で CPU や GPU への負荷がそこまで大きくならず，低リソースの PC でも動かすことができた．ただし，Pylearn2 は 2015 年に開発を停止しており，最近の研究動向などが反映されていない．

■Keras　Keras は Tensorflow や Theano を利用するライブラリである．Keras はドキュメントが作りこまれており，利用者も多いことからコミュニティも充実しておりコードの記述やエラーの排除が初心者でも比較的簡単に行える．Tensorflow を利用する際に一定数の GPU を確保されるため，並列で処理等は厳しい時があるが今回用意した GPU ではそこまで時間もかからずに学習が可能である．Keras では途中の学習結果をコールバックにより保存可能である．また，コールバックで保存された結果を読み込んで学習することも可能である．ただし，パラメータに変更を加えた場合には動かなくなることが多く，パラメータ変更の際にはすべて学習しなおす必要があった．

■Pytorch　Pytorch は Torch を利用するライブラリである．Pytorch の記述方法は Python で多次元配列を取り扱う Numpy に近く，Python のプログラムを記述したことがあれば比較的容易に記述ができる．今回 Pytorch を試したところ，小さな構造のニューラルネットワークでは動かせたが，パラメータが大きくなるとメモリ不足のエラーが出たため，今回用意した PC で使用するのが困難と判断した．

4.2　活性化関数

深層学習を行う際のパラメータのひとつに活性化関数がある．活性化関数はニューラルネットで得られた出力に対して変換を行う関数のことを指す．この活性化関数による変換によってニューラルネットの表現力は複雑にすることができる．今回学習モデルを作製するのには検討した活性化関数 3 つについて説明する．

■シグモイド関数　シグモイド関数は，以下の式の関数であり，図 4.1 のようになる．

$$y = \frac{1}{1 + \exp(-x)}$$

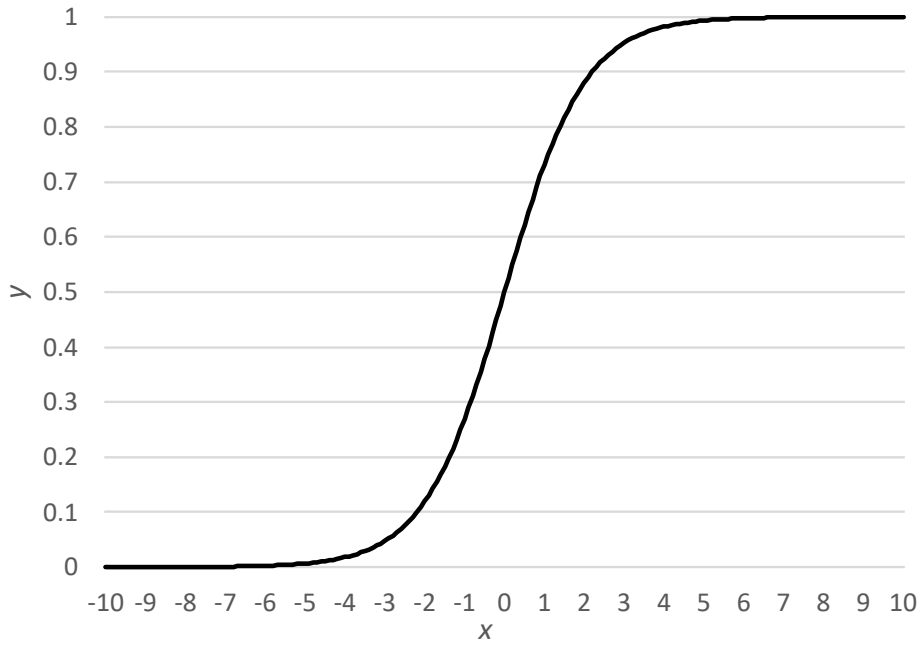

図. 4.1　シグモイド関数

シグモイド関数は微分計算が容易のため，計算量を減らすことが可能であり，高速な計算が行いやすい．シグモイド関数は 2 値分類に強く，出力値を 0 から 1 の範囲でマッピングしなおす．しかし，勾配消失が問題となる．ニューラルネットでは，勾配を基に最適解を探索する．シグモイド関数では微分係数の最大値が 0.25 のため，層が増えれば増えるほど 0 に近づく．勾配がなくなるとニューラルネットは値を更新できなくなるため，学習ができなくなる．また 0 にならなくても，勾配が小さい場合には値の更新幅が小さくなるため最適解にたどり着くまでに更新が増え，学習時間がかかるという問題がある．

■Hyperbolic Tangent (tanh) **関数**　Hyperbolic Tangent (tanh) 関数は，以下の式の関数であり，グラフは図 4.2 のようになる．

$$y = \frac{\exp(\mathrm{x}) - \exp(-\mathrm{x})}{\exp(\mathrm{x}) + \exp(-\mathrm{x})}$$

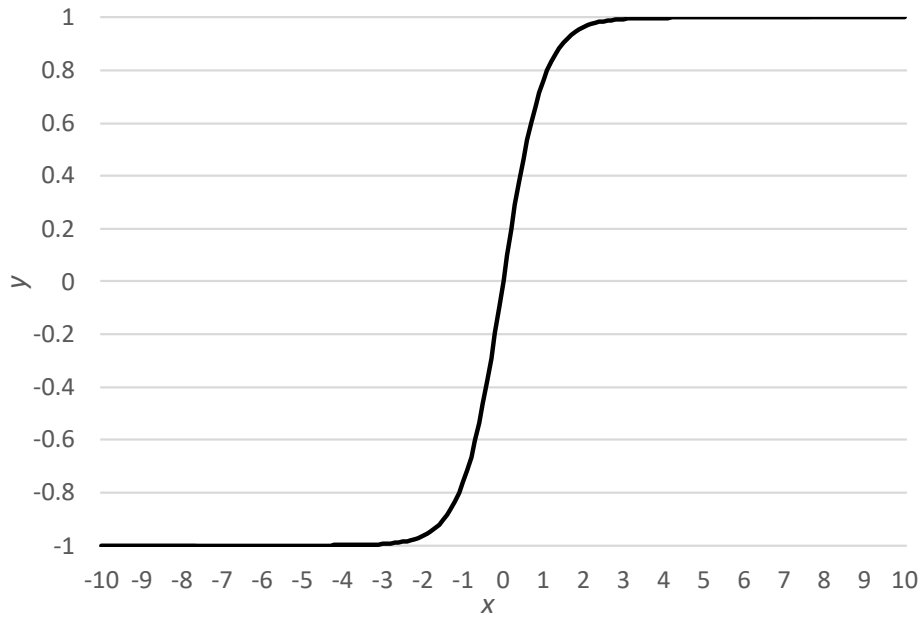

図. 4.2　tanh 関数

tanh 関数は 2 値分類などに強く，出力値を-1 から 1 の範囲でマッピングしなおす．tanh 関数の微分係数は最大値が 1 である．そのため，シグモイド関数に比べて微分係数を大きく，勾配消失問題を緩和させることが可能である．しかし，微分値が 1 になるのは局所的であり，ほとんどの出力で微分値が 1 以下になるため，勾配消失問題を解決はできない．

■Rectified Linear Unit (ReLU) **関数**　Rectified Linear Unit (ReLU) 関数 [41] は，以下の式の関数であり，グラフは図 4.3 のようになる．

$$y = \max(0, \mathrm{x})$$

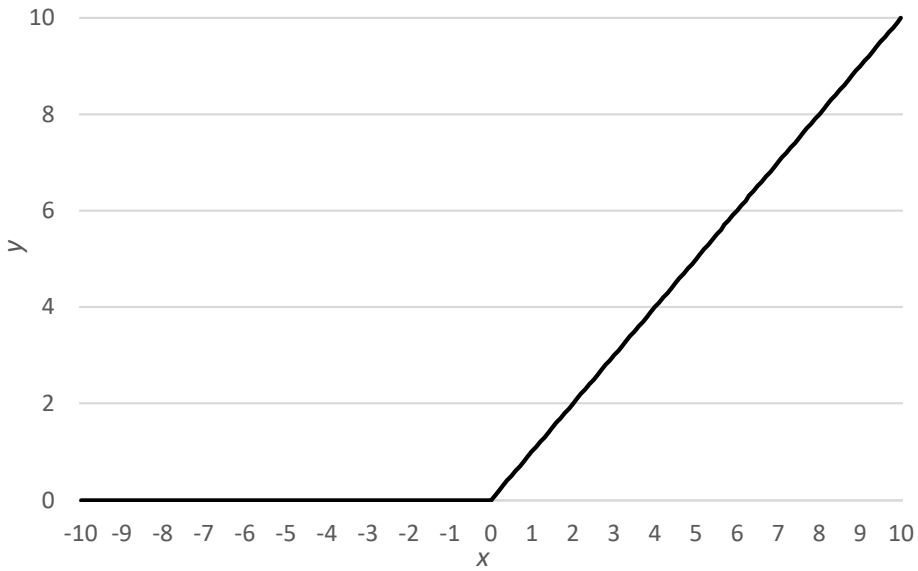

図. 4.3　ReLU 関数

ReLU は特に画像認識分野で有効とされている活性化関数である．ReLU 関数はノイズに強く，出力値がマイナスであれば 0 にし，プラスであれば無限大までその値を維持する．ReLU は勾配消失問題を解決する方法のひとつである．しかし，マイナスの出力はノイズとして無視することになるため，定常的にマイナスが出る測定値などには向かない．

4.3　遅延時間差の分析および安全性評価への影響の調査

前述の特徴を基に，安全性評価環境の構築を行った．本論文で使用するライブラリは記述の簡単さやドキュメントの充実さから Keras とした．Keras による安全性評価環境の確認のため，実際に安全性評価を行った．安全性評価の対象として，Arbiter PUF をシミュレーション実装し，遅延時間差について分析を行う．また，単純な遅延時間差の影響を見るために，遅延時間差を用いて安全性評価を行う．

4.3.1 シミュレーション PUF

128 段 Arbiter PUF を実際の PUF から計測した値を参考に，平均が 0 で，標準偏差が 10.75 のガウス分布から各段の遅延時間差を決定してシミュレーション PUF を実装した．今回は遅延時間差に対する予測成功率の影響を観察するために環境ノイズの影響は考慮しない．

4.3.2 実験環境

前述の活性化関数およびパラメータを変更し複数回試行を行い, 最適な値を導出した．表 4.3 に Keras で用いたパラメータを示す．

表 4.3　安全性評価に使用した Keras のパラメータ

隠れ層	$h1$	$h2$	$h3$	$h4$
ユニット数	5000	1000	200	100
活性化関数	tanh	sigmoid	tanh	sigmoid
Dropout	0.1	0.1	0.1	–

使用する CRP は 2^{20} = 1048576 ペア用意し，図 4.4 のように遅延時間差による分類を行った．分類は各 PUF で確率分布を求め，累積確率分布を計算し，遅延時間差の絶対値が大きい値から 10% に当たる閾値となる遅延時間差を求め分類した．トレーニングデータセットは，それぞれの分類に対して 10,000 CRP を利用した．テストデータセットには，全区間に対してとそれぞれの分類に対して 10,000 CRP を利用した．

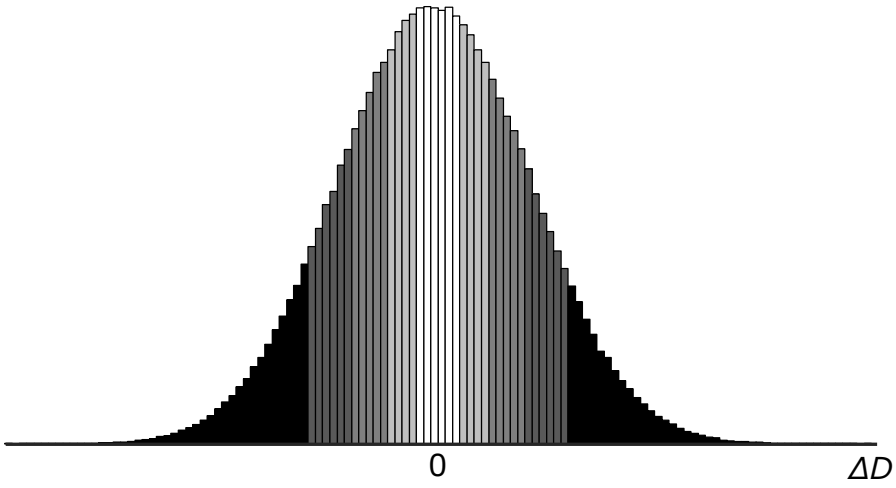

図. 4.4　遅延時間差による分類

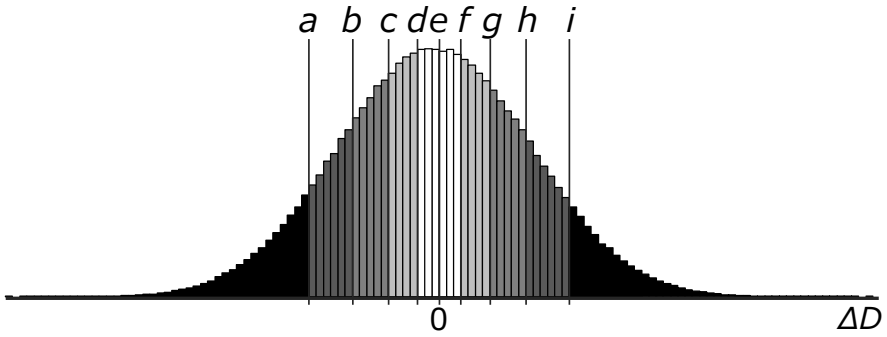

図. 4.5　遅延時間差による分類の閾値

4.3.3　実験結果

図 4.5 のように遅延時間差を分割した際の閾値を表 4.4 に示す．今回の閾値は端から 10% ごとに決定しているため，各分類には大体 104,858 CRP が属する．そのため，レスポンス，つまり，0/1 頻度に偏りが生まれるのは e の閾値の影響である．図 4.5 において白い区間に分類された CRP の偏りおよび全体の偏りを表 4.5 に示す．表からもわかるように，平均が 0 で，

表 4.4　各ＰＵＦの分類の閾値

	a	*b*	*c*	*d*
PUF 1	-154.829	-101.786	-63.1921	-30.1628
PUF 2	-156.194	-103.73	-65.4968	-32.8137
PUF 3	-168.197	-117.695	-81.0843	-49.5857
PUF 4	-157.58	-104.301	-65.1489	-32.7225
PUF 5	-142.666	-90.1974	-52.1533	-19.7961

e	*f*	*g*	*h*	i
0.564994	31.5274	64.35704	103.0508	156.2058
-2.21425	28.32188	61.60769	99.15487	151.8851
-20.0959	9.19337	40.45336	77.21276	128.0121
-1.7549	29.16434	62.41332	101.0991	154.5789
10.6105	40.92271	73.38959	111.4651	163.6041

標準偏差が 10.75 のガウス分布から遅延時間を決定していたとしても，128 段 Arbiter PUF では標準偏差が $\sigma = \sqrt{128 \times 10.75^2} = 121.6223664$ となり，PUF によってばらつきが大きくなる．今回，特に PUF3 と PUF5 で偏りが大きくなった．

次に，各トレーニングデータに対して安全性評価を行った結果を表 4.6 に示す．全体的に，分割されたトレーニングデータから作製したクローンは，全範囲のトレーニングデータから作製されたクローンと比べて予測成功率が低くなった．ただし，PUF1 と PUF5 では遅延時間差が 0 に近いトレーニングデータの方が全範囲のトレーニングを用いるよりも予測成功率が高くなった．

表 4.5　白い区間におけるレスポンス 0/1 の頻度

	PUF1	PUF2	PUF3	PUF4	PUF5
0	102915	112635	177235	110890	67855
1	106800	97080	32480	98825	141860
freq. [%]	50.19	49.26	43.10	49.42	53.53

表 4.6　遅延時間差に対する予測成功率 (%)

	PUF1	PUF2	PUF3	PUF4	PUF5
全	98.23	98.52	97.97	98.70	98.20
$\Delta D < a, i < \Delta D$	92.68	50.13	57.22	50.31	46.58
$a \leq \Delta D < b, h < \Delta D \leq i$	94.60	50.13	57.22	50.31	46.58
$b \leq \Delta D < c, g < \Delta D \leq h$	95.15	50.13	57.22	50.31	46.58
$c \leq \Delta D < d, f < \Delta D \leq g$	97.95	50.13	57.22	50.31	46.58
$d \leq \Delta \leq f$	99.46	50.13	57.22	50.31	99.02

4.4　まとめ

本章では安全性評価環境の構築について検討し，遅延時間差が安全性評価に対して与える影響を調査した．評価環境としては Keras ライブラリを利用し，活性化関数はパラメータとして各データセットに対して検討した．また，シミュレーション PUF を 5 つ作製し，遅延時間差の分析および安全性評価を行った．遅延時間差の分析の結果，段数を増やすことで遅延時間差の標準偏差が大きくなり，レスポンスに偏りが生じやすくなることを示した．また，遅延時間

差による安全性評価の結果では，トレーニングデータに含まれる CRP の遅延時間差によって結果に差が生じる可能性を示した.

第 5 章

安全性評価 1: 時系列処理を行った PUF に対する安全性評価

5.1 はじめに

本章では，時系列処理を行った PUF に対して安全性評価を行う．本章において Arbiter PUF の構成を変更せず，遅延時間差を元に Arbiter 回路やレスポンスに加工を行うことを時系列処理とする．具体的には第 2.1.4 章で説明した RG-DTM PUF [14] と Q-class 認証 [68] を時系列処理を行った PUF とする．

本章では，RG-DTM PUF に対して深層学習を用いた安全性評価を行い，深層学習が時系列処理を行った PUF に対して LR より安全性評価において有効であることを示す．次に，Q-class 認証に対して安全性評価を行い，これまで深層学習を用いた攻撃への対策手法として有効だとされていた Q-class 認証はクローニング攻撃への耐性向上には適さないということを示す．結果として，時系列処理は深層学習を用いたクローニング攻撃への耐性を向上するには適さない

という知見を得た.

5.2 RG-DTM PUF に対する安全性評価

Extensive PUF の 1 つである RG-DTM PUF [14] は，複雑な構造の Arbiter 回路を持ち，機械学習攻撃を用いた複製が困難なことが報告されている [56]．先行研究では，Support Vector Machine (SVM) や Logistic Regression (LR) といった学習器を安全性評価へ利用しており，本論文で取り扱ったような深層学習による学習は行われていない．そこで本章では RG-DTM PUF に対して深層学習を用いたクローニング攻撃を行い安全性評価を行う．

5.2.1 安全性評価

5.2.1.1 シミュレーション PUF

安全性評価を行う RG-DTM PUF は，MATLAB を用いてシミュレートした．RG-DTM PUF の構成は基本的には Arbiter PUF と同様であり，Arbiter 回路が異なる．まずベースとなる Arbiter PUF のシミュレーションは，第 4 章と同様に実際の PUF から計測した値を参考に，平均が 0 で，標準偏差が 10.75 のガウス分布から遅延時間差 $(\delta_l^0,\ \delta_l^1)$ をランダムに決定した．また，Arbiter 回路上で発生する遅延はなく，環境ノイズの影響はないとした．セレクターチェーン部分で発生した遅延時間差は Arbiter 回路に到達した際に 0 を中心としたガウス分布に従う．RG-DTM PUF はその遅延時間差によって分割し，レスポンスを決定する．そこで，遅延時間差の分割数を 2，4，6，8，12，16 とし，遅延時間差を元に等間隔で分割した．このとき，今回の RG-DTM PUF は遅延時間差で分割したため，各分割に含まれる CRP 数には偏りが生じる．シミュレーション PUF は，段数を 32，64，96，128 として，各条件に対して 5 つ作製

した.

5.2.1.2 実験環境

表 5.1 に Keras で用いたパラメータを示す.

表 5.1 安全性評価に使用した Keras のパラメータ

隠れ層	$h1$	$h2$	$h3$	$h4$
ユニット数	5000	1000	200	100
活性化関数	tanh	sigmoid	sigmoid	sigmoid
Dropout	0.1	0.1	0.1	–

先行研究 [56] では一定の予測成功率を達成する際に必要なトレーニングデータの数を用いて安全性評価を行っている．まず，先行研究で行った SVM や LR といった学習器との比較のために，トレーニングデータを 1,000 CRP から 500,000 CRP まで変更して学習を行った．テストデータは 10,000 CRP とし，何 % のレスポンスを正しく予測できたかによって予測成功率を計算した.

5.2.1.3 実験結果

■先行研究 [56] との比較　RG-DTM PUF に対する深層学習を用いた安全性評価の有効性を検証するために先行研究との比較を行う．先行研究と同じ条件にするために，この章では 32 段の RG-DTM PUF に対して安全性評価を行う．先行研究では，初めて予測成功率が 95% を達成した際に必要だったトレーニングデータ数を用いて安全性評価を行っている．そこで，深層学習を用いた結果について，95% の予測成功率を最も少ないトレーニングデータ数で達成した場合 (BEST) と 5 つのクローンにおける必要なトレーニングデータ数の平均値 (AVERAGE)

の 2 つを示す．

比較結果を図 5.1 に示す．横軸は RG-DTM PUF の分割数，縦軸は 95% の予測成功率を達成した時のトレーニングデータ数を表す．95% の予測成功率を達成する時に必要なトレーニングデータ数は，少ない方がパラメータ推定が容易である．また，攻撃に必要な CRP 数が少ないため攻撃コストが低くなる．つまり図内のプロットは下にある方がより優れている結果である．

図 5.1 に示すように，RG-DTM PUF の分割数が 4 以上の場合，BEST に必要なデータ数は先行研究に比べて少なかったが，分割数が 12 未満の場合には AVERAGE に必要なデータ数は先行研究より多くなった．一方，分割数が 12 以上になった時，深層学習の結果が先行研究と比較して良くなり，AVERAGE に必要なデータ数も先行研究と比べて少なかった．特に，分割数が 16 の場合，先行研究では 100,000 CRP を用いても 95% の予測成功率を達成しなかったが，深層学習では BEST で 90,000 CRP，AVERAGE で 200,000 CRP で 95% を達成した．

RG-DTM PUF の分割数が少ない場合，深層学習の平均予測精度は LR を用いた先行研究の結果よりも低くなった．一方で，分割数が 16 の場合，LR ではレスポンス予測が困難であったが，深層学習は学習データ数が 200,000 CRP あれば，95% 予測できた．複雑な分割を行う PUF に対してクローニング攻撃をする場合には，LR よりも深層学習の方が有効になる．ただし，取得する CRP 数が極端に少ない場合には深層学習でもクローニング攻撃が困難になる．また，攻撃対象の PUF が単純な構成であり，攻撃者が取得できる CRP 数に限りがある攻撃シナリオでは深層学習より LR の方が有効になる．

■バッチサイズによる学習速度と予測成功率の違い　先行研究 [56] との比較では，深層学習のパラメータとしてバッチサイズを 100 として学習したが，トレーニングデータデータセットの数が増えるに従い学習時間が増大した．そこで学習時間を高速化するために，バッチサ

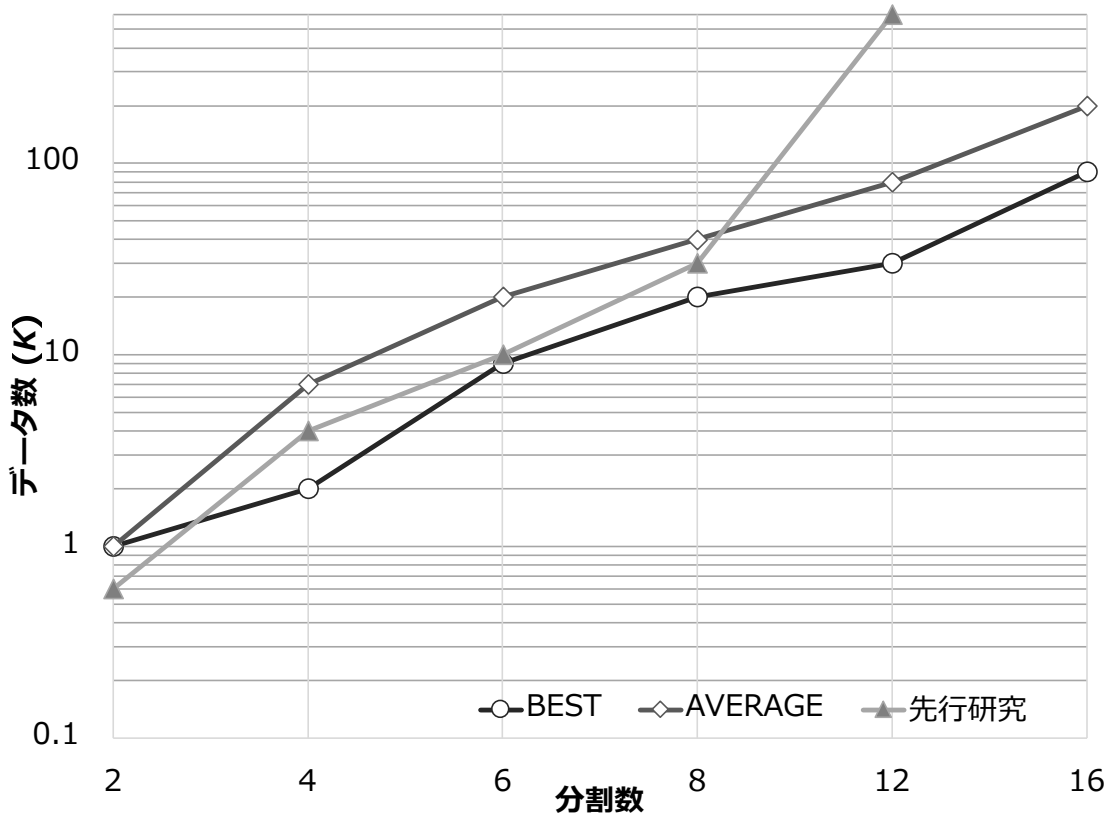

図. 5.1　95% を達成時に必要データ数について先行研究 [56] と最も良い予測成功率 (BEST) と 5 つのクローンにおける予測成功率の平均値 (AVERAGE) の比較

イズを 100 と 1000 にした時の学習時間および予測成功率に関して比較を行う．前章の先行研究 [56] との比較で用いた 5 つの 32 段の 16-分割 RG-DTM PUF に対して安全性評価を実施する．また，予測成功率は 5 つのクローンの予測成功率の平均とする．

表 5.2 にトレーニングデータセットの数による学習時間の変化を示す．

安全性評価の結果，全体的にバッチサイズを 100 から 1000 に変更することで学習時間の高速化が可能である．特にトレーニングデータセットの数が 500,000 CRP まで増えた場合，学習時間が 4200 秒，すなわち 1 時間 10 分の差となった．

次に，図 5.2 にバッチサイズによる予測成功率の差を表す．バッチサイズが 100 または 1000

表 5.2　トレーニングデータセットの数およびバッチ数の違いによる学習時間

	バッチサイズ	
	100	1000
1,000	32.47s	9.91s
5,000	73.56s	18.38s
10,000	123.91s	28.02s
50,000	533.06s	106.44s
100,000	1051.48s	205.19s
500,000	5206.84s	1007.45s

の場合に対して，分割数ごとにトレーニングデータセットの数を増やし，予測成功率が 95% 以上を 5 回連続で達成した時点で学習を終了する．横軸はトレーニングデータセットの数，縦軸は予測成功率の値を表す.

安全性評価の結果，2-分割では，バッチサイズを変更させても予測成功率はほとんど同じ結果となった．4-分割では，トレーニングデータセットの数が 1,000 CRP の時にバッチサイズ 100 のみが予測成功率 90% を達成した．しかし，バッチサイズが 1000 の時，予測成功率が 95% 以上を 5 連続で達成するトレーニングデータセットの数が 10,000 CRP となり，バッチサイズ 100 の時の 40,000 CRP よりも少ない結果となった．6-分割，8-分割ではバッチサイズ 100 の方が少ないトレーニングデータセットの数で予測成功率が上がる傾向があったが，予測成功率が 95% 以上になるトレーニングデータセットの数はバッチサイズに関係なく同じであった．12-分割と 16-分割では全体的にバッチサイズ 100 に比べてバッチサイズ 1000 の方が予測成功率が下がる結果となった．12-分割と，16-分割共に，バッチサイズに関係なく，95%

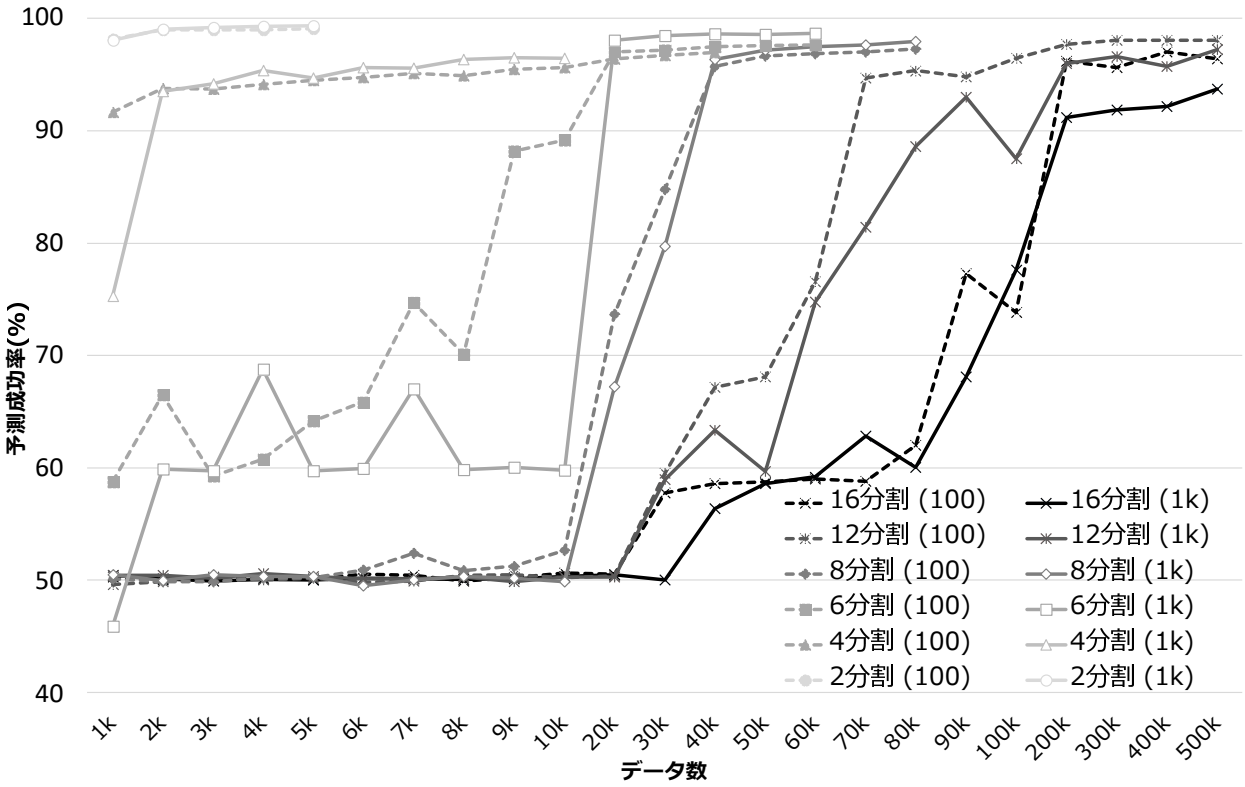

図. 5.2　バッチサイズおよび分割数による予測成功率の違い

を 5 回達成はしなかったが，500,000 CRP では 90% 以上を達成した．そのため，学習時間という攻撃コストを下げるためにバッチサイズを変更した場合でも，安全性評価の結果が大きく変化しないことがわかった．

■段数による予測成功率の変化　前節の結果から深層学習を安全性評価に用いた時，32 段 RG-DTM PUF で 16-分割まで増やしても予測成功率は 90% を達成した．そこで，クローニング攻撃に必要なパラメータ数を増やすために段数を変化させた時の予測成功率について評価する．段数は 32 段から 32 段ごと増加させ，64，96，128 に変更する．この時，学習時間の高速化のためにバッチ数は 1000 とし，学習クローニング攻撃に用いるトレーニングデータセットは 500,000 CRP とする．安全性評価の結果は 5 つのクローンの予測成功率の平均値とする．

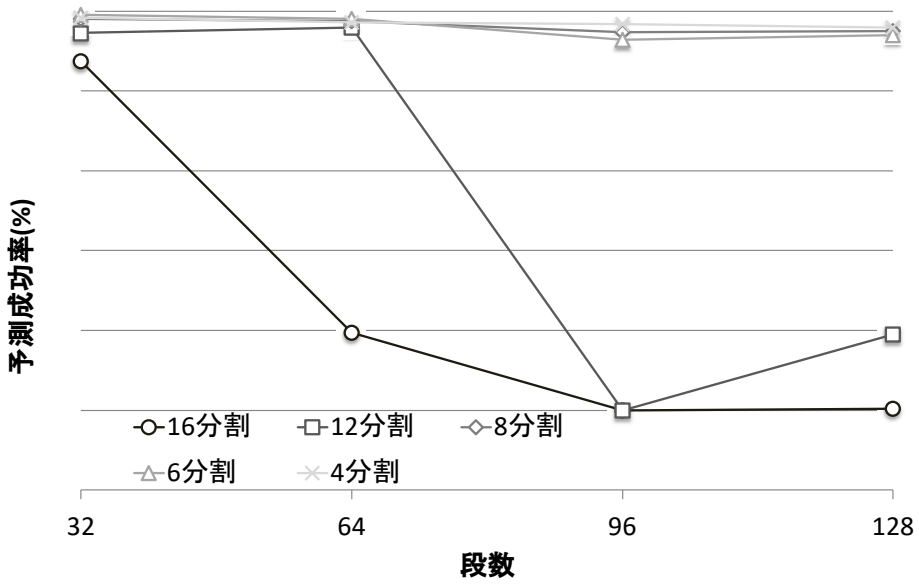

図. 5.3　段数の変化による予測成功率の変化

図 5.3 に段数による予測成功率の変化を示す．64 段にした場合，16-分割では，4 つの PUF で予測成功率が約 50% まで低下した．しかし，1 つの PUF では予測成功率が 90% 以上だったため，平均予測成功率が 60% となった．96 段にした場合，16-分割，12-分割で全てのクローンの予測成功率が 50% となった．128 段にした場合，16-分割では全てのクローンの予測成功率が 50% となった．しかし，12-分割では 1 つのクローンの予測成功率が 90% 以上だったため平均の予測成功率が 60% となり，96 段よりも上昇した．一方で，4-，6-，8-分割では段数を増やしても予測成功率に変化がほとんど見られなかった．

128 段 12-分割 RG-DTM PUF における予測成功率の差を見るために遅延時間差の分析をした結果を表 5.3 に示す．図 5.4 に示すように，分析のために第 4 章のように遅延時間差ごとに分割し，遅延時間差がマイナスから範囲ごとに *a–l* とした．そして PUF ごとにランダムに 10,000 CRP 取得し，各 CRP の遅延時間差がどの範囲に分類されるか分析し，各範囲に含まれる CRP の割合を示す．

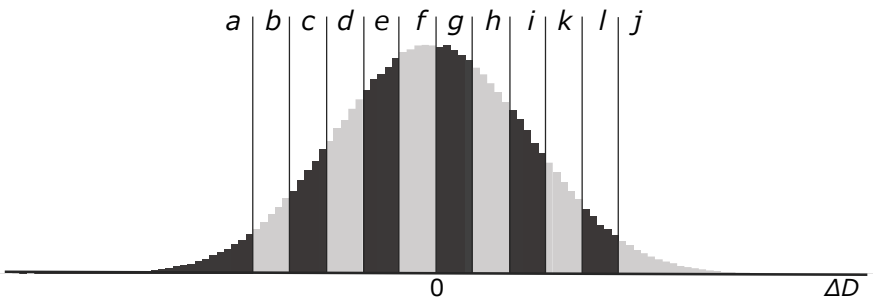

図. 5.4　遅延時間差による 12-分割 RG-DTM PUF の分割

表に示されたように，段数が多くなることで PUF ごとの遅延時間差の分散が大きくなり，各値域に属する CRP 数に偏りが生じた．予測精度が 95% まで上がった PUF3 では，他の PUF に比べて分散が大きく a および l に属する CRP の割合が高かったことがわかる．深層学習が学習するにあたり，該当するデータ数が少なすぎると学習をうまく行うことができない傾向がある．そのため，偏りが大きい PUF では，CRP の少ない区間があったため学習がうまく行えなかったと考えられる．

5.3　n-XOR PUF を用いた Q-class 認証に対する安全性評価

文献 [68] では，Q-class 認証を DAPUF に対して適用し，Q-class が通常より深層学習に対する耐性を向上させたと報告している．そこで Q-class 認証を n-XOR PUF に用いて安全性評価を行う．DAPUF は FPGA 上に実装された Arbiter PUF のユニーク性の低さを改善するために提案された PUF である．今回は DAPUF に構造が似ているシミュレーション実装した n-XOR PUF に対して Q-class 認証を用いた場合の安全性評価を行う．

表 5.3　各 128 段 12-分割 RG-DTM PUF における遅延時間差によって所属する CRP の確率 (%)

	a	*b*	*c*	*d*	*e*	*f*
PUF 1	0.02	0.25	1.9	7.08	15.84	24.16
PUF 2	0.09	0.63	3.14	8.48	15.83	21.91
PUF 3	0.24	1.19	4.32	9.49	15.65	20.11
PUF 4	0.10	0.73	3.38	8.83	16.04	21.77
PUF 5	0.14	0.85	3.67	8.88	15.78	21.12

g	*h*	*i*	*j*	*k*	*l*	予測成功率
24.48	16.47	7.4	2.1	0.28	0.02	50.27
21.90	15.81	8.37	3.11	0.64	0.09	50.89
19.90	15.10	8.84	3.91	1.03	0.22	97.21
21.52	15.51	8.28	3.11	0.65	0.08	50.22
21.01	15.47	8.65	3.5	0.8	0.13	48.79

5.3.1　*Q*-class 認証システム

本節では，*Q*-class 認証のコンセプトおよび認証フローについて説明する．PUF のレスポンスはハードウェア実装された時には環境ノイズの影響をうけて，レスポンスにエラーが発生する．そのため，同じチャレンジを複数回数実行した時のレスポンスの総和 (r_s)，つまりレスポンス 1 が出力されたビット数に対する CRP 数の分布は図 5.5 （図は繰り返し回数 63 の時）のようになる．*Q*-class 認証では r_s の値に従ってクラスを分類する．例えば図 5.5 のような 4-class の場合，class 1 ($r_s = 0$)，class 2 ($1 \leq r_s \leq 31$)，class 3 ($32 \leq r_s \leq 62$)，class 4 ($r_s = 63$)

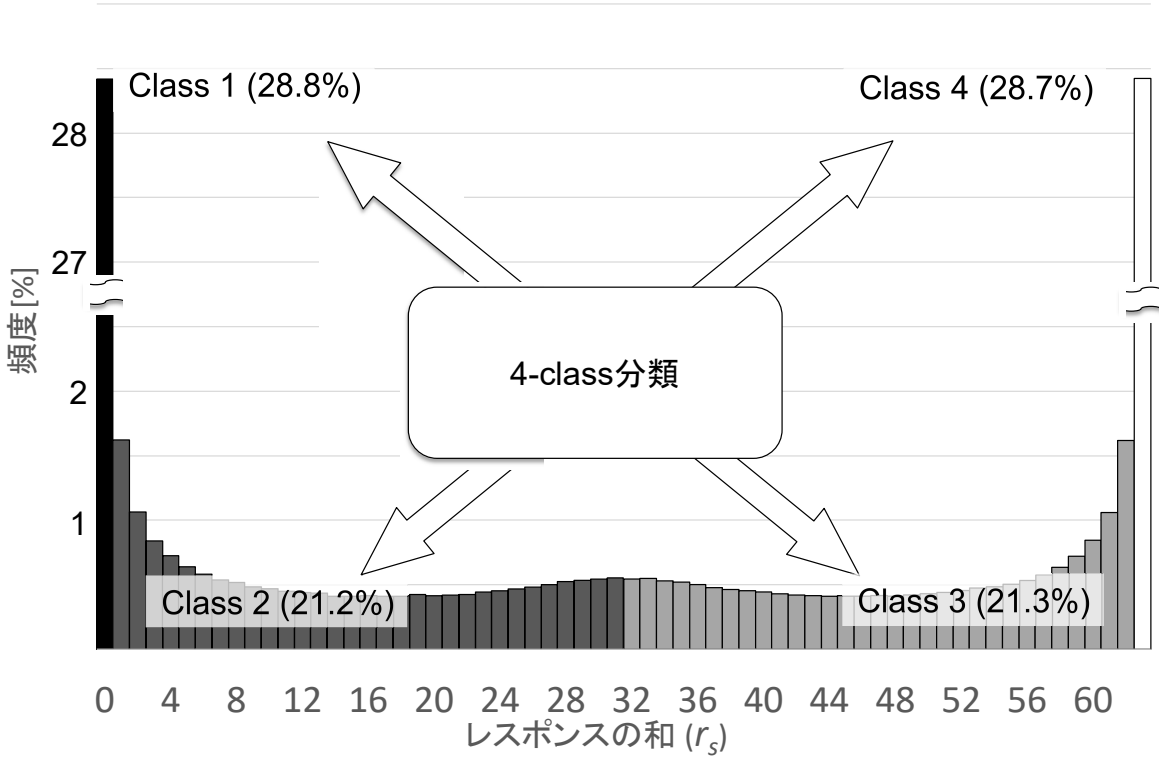

図. 5.5　Q-class 分類のイメージ

のように分類する.

Q-class 認証の具体的な認証フローを図 5.6 に示す．認証フローは次のようになる.

1. 検証のために必要なパラメータを決定する (Q：クラスの値，r_s：レスポンスの総和，m_R：CRP を作成する際の PUF を実行反復回数，k：認証に必要なチャレンジの数)
2. 登録フェーズでは，検証者は正当な PUF から CRP のデータベースを作成する．この時，PUF のレスポンスはクラス番号を指す.
3. 検証フェーズでは，サーバーは保存されている CRP からマスターチャレンジを選択し,，それを証明者 (PUF) に送信する.

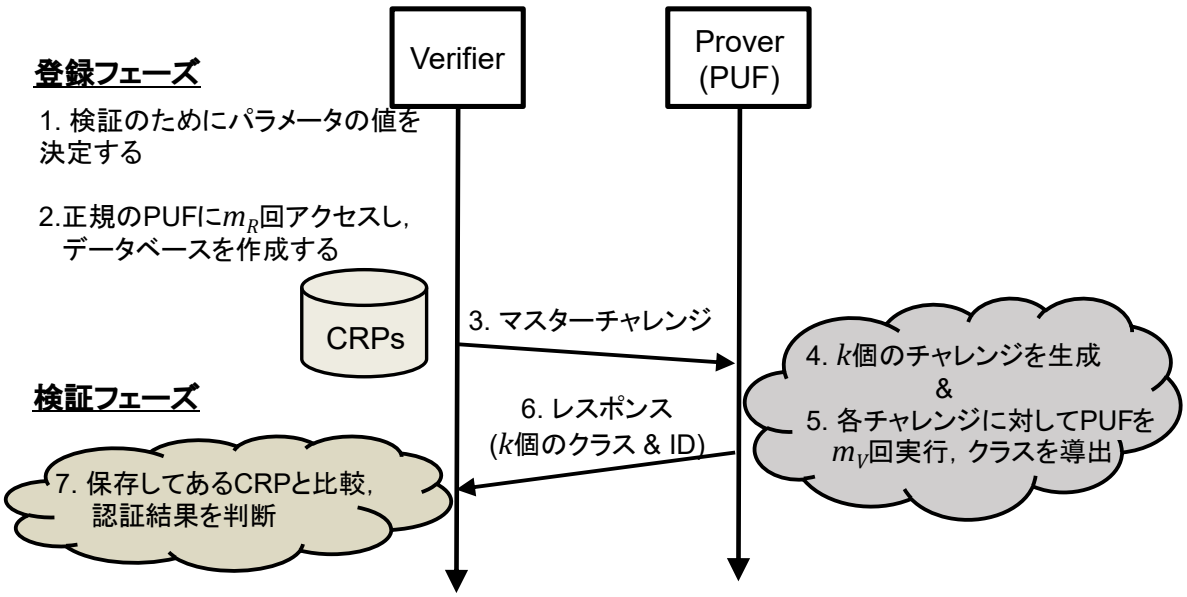

図. 5.6　Q-class 認証フロー

4. 証明者はマスターチャレンジを受け取り，そこから k 個のチャレンジを生成する．
5. 証明者は，チャレンジごとに PUF を m_P 回実行し，レスポンス 1 のカウントにより，対応するクラスを導き出す．
6. 証明者は，k 個のクラス番号と ID を含むレスポンスデータを検証者に送信する．
7. 検証者は，レスポンスを保存されている CRP のデータと比較し，同値である確率が閾値以上かによって認証結果 (成功または失敗) を判断する．

5.3.2　安全性評価

5.3.2.1　安全性評価に用いるシミュレーション PUF

シミュレーションでは，実際の PUF から計測した値を参考に，平均が 0 で，標準偏差が 10.75 のガウス分布から遅延時間差 $(\delta_l^0,\ \delta_l^1)$ をランダムに決定する．すなわち，遅延時間差は

チャレンジビットによって異なることを想定する．今回は 128 段 n-XOR PUF (n = 2–6) を 10 個シミュレーションする．

シミュレーション PUF では遅延時間差パラメータを決定し，遅延時間差の総和によりレスポンスを導出するため環境ノイズは存在しない．Q-class 認証では複数回同じチャレンジを実行した際のレスポンスのエラーを利用する．そこで，シミュレーション PUF 上で環境ノイズの再現を行った．今回シミュレーションのベースにした PUF では，Arbiter PUF のレスポンス 1 ビットにエラーが起こる確率 Bit Error Rate (BER) が 0.0296 だったため，それを参考に環境ノイズを再現した．今回は Q の値は 2 から 4 としてクラス分類を行う．

5.3.2.2 実験環境

表 5.4 に Keras で用いたパラメータを示す．トレーニングデータセット，テストデータセット共に利用するレスポンスは Q-class に対応した多値レスポンスとする．トレーニングデータセットは 1,000,000 CRP を用い，テストデータセットは 10,000 CRP とし，何 % のレスポンスを正しく予測できたかによって予測成功率を計算する．

表 5.4　安全性評価に使用した Keras のパラメータ

隠れ層	$h1$	$h2$	$h3$	$h4$	$h5$
ユニット数	5000	1000	500	100	50
活性化関数	tanh	sigmoid	tanh	sigmoid	tanh
Dropout	0.1	0.1	0.1	0.1	–

5.3.2.3　実験結果

安全性評価の結果を図 5.7 に示す．図に使用した予測成功率は 10 個の PUF の平均とする．2-XOR PUF と 3-XOR PUF では，Q の数を増やすことで予測成功率は 2% ほど低下したが，有意な差は得られなかった．4-XOR PUF では 3-class にした時に予測成功率が最も高くなり，4-class は 2-class より低下したが，ほとんど低下しなかった．5-XOR PUF と 6-XOR PUF では Q-class 認証にすることで通常の n-XOR PUF (Q=2) より予測成功率が高くなることがわかった．

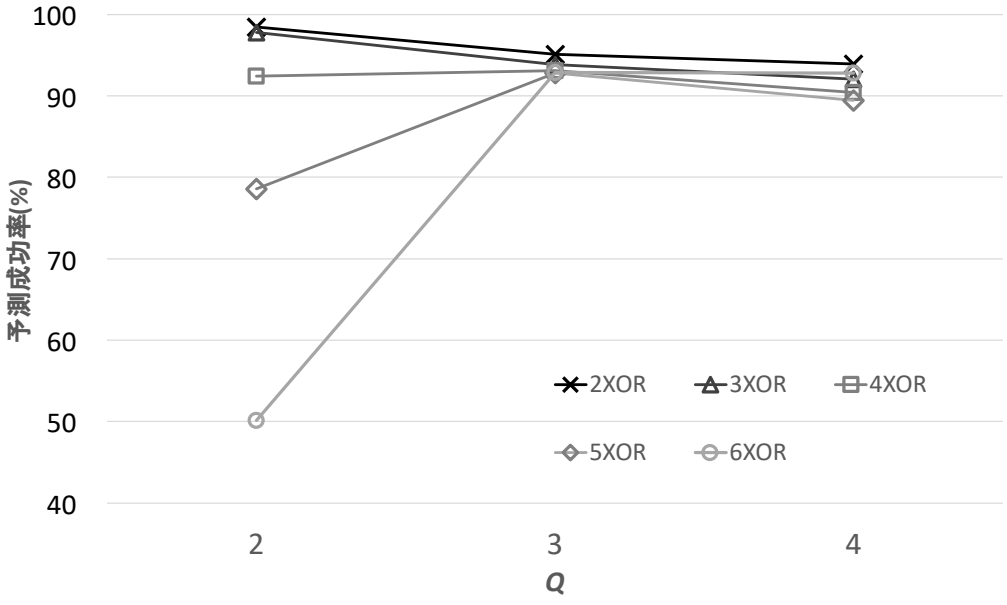

図. 5.7　n-XOR PUF を用いた Q-class 認証 (Q=2–4) に対する安全性評価

Q-class 認証は各クラスに属する CRP 数に偏りがあると学習しやすくなることがあるため，各クラスの分布率と BER を導出する．表 5.5 に n-XOR PUF の BER，表 5.6 に各クラスに分類される CRP の分布率を示す．BER はシミュレーション PUF 上で環境ノイズを考慮しない場合のレスポンスをベースとして，環境ノイズを考慮したシミュレーション PUF のレスポン

スでどれくらいビット反転が起きているか平均確率を導出する．10 個の PUF ごとに Q =2–4 の場合における各クラスの分布率を導出し，最終的に全ての PUF の分布率を平均する．分布率の表示は閾値となるレスポンスの総和 (r_s) が小さいクラスを左から示す．安全性評価の結果，表 5.5 から n-XOR PUF は n の値が増えるほど BER が上がるため，表 5.6 に示すように不安定なレスポンスが増えているのがわかる．一方で，同様に各クラスの分布数の Q=2 を見ると n が増えるほど 0 と 1 の偏りが小さくなっていることがわかる．Q=3 の時，$n \geq 4$ では不安定なレスポンスが増えて中央のクラスに他の 2 クラスよりも多く割り振られていることがわかる．Q=4 の時，$n \geq 5$ にすると中央のクラスが他のクラスより 10% 以上多く割り振られていることがわかる．

表 5.5　n-XOR PUF の BER

n	2	3	4	5	6
BER	0.0572	0.0836	0.1075	0.131	0.1527

表 5.6　Q と n-XOR の違いによる各クラスの分布率

Q	2		3			4			
2-XOR	0.509	0.491	0.352	0.314	0.334	0.352	0.157	0.157	0.334
3-XOR	0.501	0.499	0.284	0.433	0.283	0.284	0.217	0.216	0.283
4-XOR	0.500	0.500	0.236	0.528	0.236	0.236	0.264	0.264	0.236
5-XOR	0.500	0.500	0.195	0.610	0.195	0.195	0.305	0.305	0.195
6-XOR	0.500	0.500	0.162	0.676	0.162	0.162	0.338	0.338	0.162

5.4 考察

RG-DTM PUF では，段数を 32 段から 128 段に増やすことによって PUF ごとに遅延時間差の総和のばらつきが大きくなった．そして，今回設定した遅延時間差を等間隔に分割した閾値では，レスポンスの偏りが大きかった PUF3 ではクローンの予測成功率が高くなった．このことから，分割のための閾値が予測成功率に大きく関係していることが考えられる．RG-DTM PUF は Arbiter 回路の閾値によりユニーク性を増加させるため，PUF3 のように各ブロックに属する CRP が多い方が理想である．ただし，5.3.2.3 で示したように，理想の状況にすると予測成功率が高くなるため，ユニーク性が増加してもクローニング攻撃への耐性が低下する．

Q-class 認証では，Q の数が増えると予測成功率が低下すると考えられていた．しかし，実際には Q=2 と Q=4 を比較した場合，Q=4 の方が複雑な構成をしているが予測成功率が上昇した．

これまでの結果から，時系列処理は予測成功率が高くする要因となり，つまりクローニング攻撃への耐性を低下させるとわかる．RG-DTM PUF では，理想的な閾値を PUF の性能評価の結果から考える必要がある．具体的にはユニーク性を上げるために閾値の分割を増やすと，再現性やクローニング攻撃への耐性へ影響が出る可能性が高い．そのため，アプリケーションや実装方法を想定しつつ，理想的な閾値数を設定する必要がある．また，安全性評価の結果によっては今回のように遅延時間ベースではなく，CRP 数ベースの分割方法などを検討する必要がある．一方，Q-class 認証は時系列処理により予測成功率が低下したため，安全性向上という目的には適さない．

5.5 まとめ

本章では時系列処理を行った PUF に対する安全性評価として RG-DTM PUF と Q-class 認証に対して安全性評価を行った．RG-DTM PUF に対する安全性評価では深層学習を用いることで LR よりも予測成功率が大きくなることがわかった．しかし，段数および分割数が多い RG-DTM PUF では，予測成功率が低くなった．RG-DTM の分割方法には遅延時間差によって分割する手法と CRP 数によって分割する手法の 2 つが考えられる．今回 RG-DTM PUF は遅延時間差を等間隔に分割している．そのため，段数を増やした時に遅延時間差のばらつきによっては分割に偏りが生じ，安全性評価に影響を与える可能性がある．Q-class 認証はクローニング攻撃への耐性を向上させるために提案された認証方式である．しかし結果として Q-class にすることによる優位性が見られず，6-XOR PUF については予測成功率が上がった．今回は n-XOR PUF に注目したため，先行研究で扱った DAPUF への安全性評価は実行していない．しかし，3-1 DAPUF と 6-XOR PUF は最終的に XOR するセレクターチェーンの数が同じという点において類似するため，結果は大きく変わらないことが考えられる．RG-DTM PUF や Q-class に対する安全性評価の結果から時系列処理を行った PUF では，レスポンスに 1 つの PUF のどのような遅延時間差を持つかの情報が含まれており，深層学習に対して有利に働いたことが考えられる．今回，時系列処理による安全性向上という結果は得られなかった．それどころか時系列処理を行うことで安全性が低下する場合があるという知見を得た．

第 6 章

安全性評価 2: 並列実装された PUF に対する安全性評価

6.1　はじめに

本章では並列実装された PUF に対して安全性評価を行う．並列実装された PUF としては，Arbiter PUF を n 個並列に実装する n-XOR PUF を取り扱う．本章で取り扱う n-XOR PUF は，実際の IC チップ上に実装された Arbiter PUF である．

本章では，n の値が大きい n-XOR PUF に対して深層学習を用いた安全性評価を行い，並列実装が深層学習を用いたクローニング攻撃への耐性を向上することが可能なことを示す．

6.2　安全性評価

6.2.1　実験環境

n-XOR PUF の構成を図 6.1 に示す．

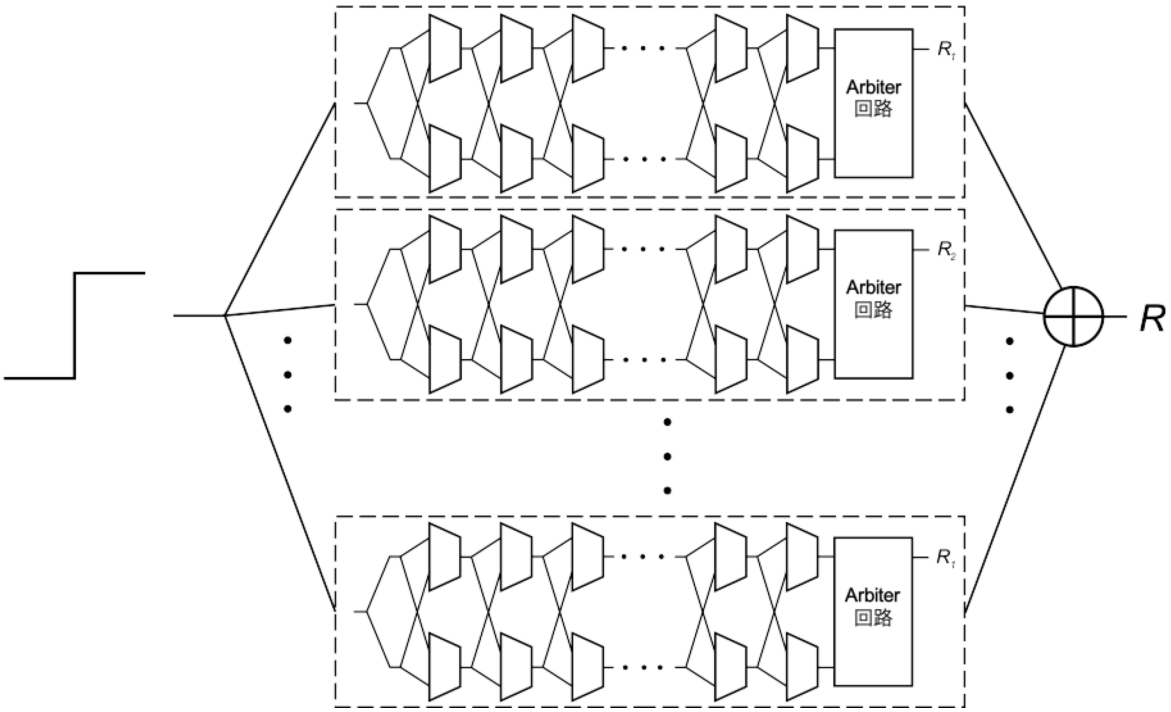

図. 6.1　*n*-XOR PUF の構成

今回の実験では 180nm シリコンチップに実装された Arbiter PUF を利用し，*n*-XOR PUF のレスポンスをソフトウェア処理により実現する．IC チップは 9 個あり，各チップには 4 つの Arbiter PUF が実装されている．今回の実験では，*n*=20 を最大値とし，全 36 個の Arbiter PUF のうち，20 個の Arbiter PUF から CRP を取得した (図 6.1 内点線部). 安全性評価に使用する *n*-XOR PUF のデータセットに変換するため，2 から 20 個の Arbiter PUF に対して同じチャレンジを与えたときのレスポンスをソフトウェア上で XOR し，*n*-XOR PUF の CRP とした．実際に *n*-XOR PUF を認証プリミティブとして運用する場合，*n*-XOR PUF のチャレンジは，各 Arbiter PUF に異なるチャレンジを与える場合と同じチャレンジを与える場合が考えられる．今回は，全ての Arbiter PUF に同じチャレンジを入力する *n*-XOR PUF を想定する．この時，攻撃者に必要な推定パラメータは *n* 個の Arbiter PUF に対する攻撃と同じになる．

使用した Keras のパラメータを表 6.1 に示す．今回利用した Keras のパラメータは第 4 章で説明した活性化関数および複数のパラメータを用いて最もよい結果を導出した．

表 6.1　安全性評価に使用した Keras のパラメータ

隠れ層	$h1$	$h2$	$h3$	$h4$	$h5$
ユニット数	5000	1000	500	100	50
活性化関数	tanh	sigmoid	tanh	sigmoid	tanh
Dropout	0.1	0.1	0.1	0.1	–

トレーニングデータセットは 1,000,000 CRP とし，テストデータセットは 10,000 CRP とした．データセットは 2 種類用意する．Arbiter PUF のレスポンスは，同じチャレンジに対するレスポンスを取得した場合でも，環境ノイズなどの影響によりビット反転を起こす可能性がある．つまり，ハードウェア実装した PUF から得られるレスポンスには，環境ノイズの影響を受けたエラービットが含まれる．このことから，データセットとして次の 2 種類を作成した．1 つは，チップから取得したエラービットを含むレスポンス値をそのまま使用した未加工データセットである．もう 1 つは，同じチャレンジに対して 32 回 PUF を動作させ，32 ビットのレスポンスを取得し，多数決で 1 ビットのレスポンスを決定した多数決データセットである．チャレンジによって，レスポンスが受ける環境ノイズの影響に差はあるが，多数決をとることにより環境ノイズの影響を緩和することが可能である．

本章で行った安全性評価では，トレーニングデータセットとテストデータセットの両方で未加工データと多数決データの 2 種類を用意する．ただし，理想のテストデータセットは，環境ノイズの影響がない CRP である．PUF を利用した認証システムでは，認証結果を導出する際に，レスポンスの誤りに対して閾値を設け，その閾値より被認証者から送られたレスポンスの

誤りが低ければ認証を通過させる方式がある．この時，環境ノイズの影響を受けるレスポンスを予測するのは困難である．そのため，クローニング攻撃で作製されたクローンが被認証者として検証される時，設けられた閾値よりレスポンス予測の誤りが低いか判断するには，テストデータに環境ノイズの影響がない CRP が最適となる．仮に，クローンが環境ノイズの影響を学習できていたとしても検証者は誤りレスポンス数で認証の可否が判断できるためである．そこで，未加工データ，多数決データそれぞれをトレーニングデータおよびテストデータとした場で評価を行い，さらに，未加工データのトレーニングデータに対して多数決データをテストデータとして評価を行った．

6.2.2　実験結果

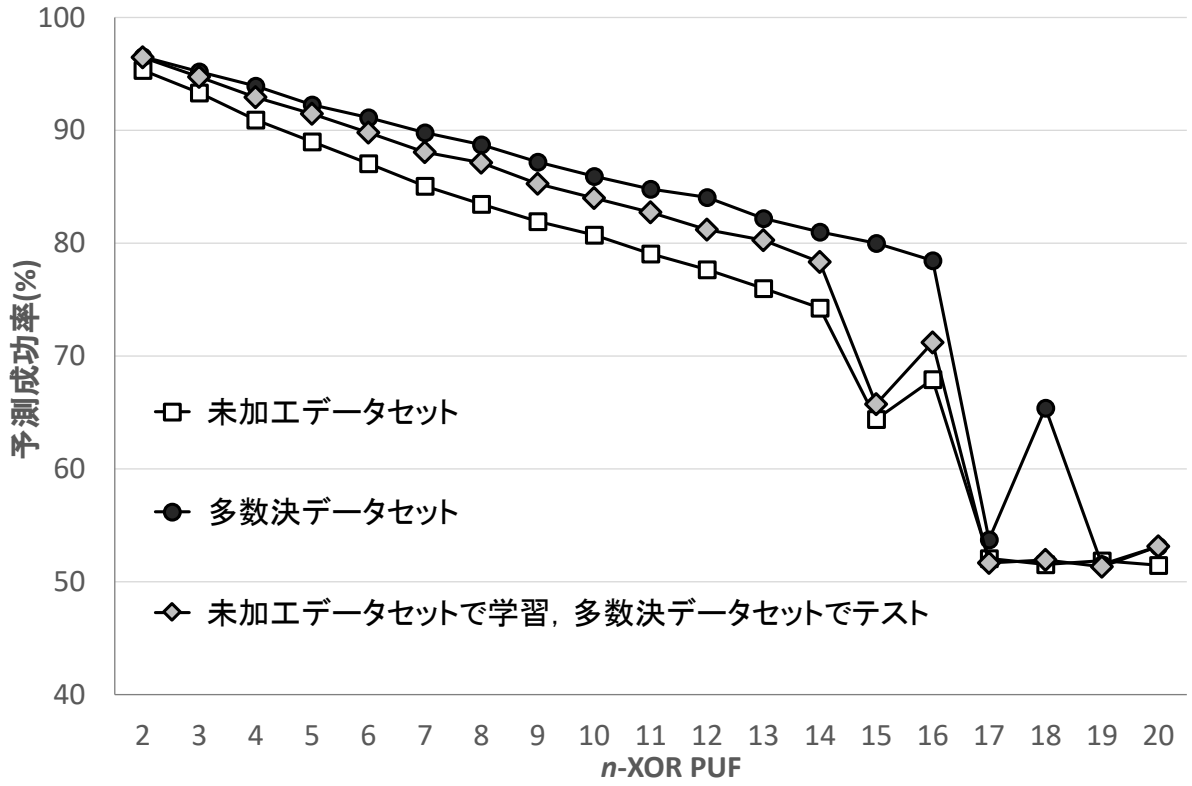

図. 6.2　*n*-XOR PUF に対する安全性評価の結果

表 6.2　*n*-XOR PUF に対する安全性評価の結果

n	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
□	95.35	93.35	90.92	89.01	87.07	85.04	83.48	81.95	80.71
○	96.50	95.23	93.93	92.29	91.12	89.77	88.70	87.18	85.91
◇	96.49	94.71	92.95	91.48	89.82	88.09	87.16	85.24	84.01

11	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
79.05	77.65	75.98	74.24	64.37	67.94	52.08	51.52	51.88	51.46
84.80	84.05	82.21	81.01	79.97	78.47	53.75	65.41	51.50	53.16
82.73	81.18	80.27	78.32	65.71	71.23	51.65	51.91	51.36	53.12

図 6.2，表 6.2 に安全性評価の結果を示す．縦軸は予測成功率を表し，横軸は n の値を表す．正方形でプロットした線は未加工データの予測成功率を示しており，黒丸でプロットした線は多数決データで学習した予測成功率を示す．ひし形でプロットした線は未加工データをトレーニングデータとし，多数決データをテストデータとした結果である．図からわかるように，並列実装の n の数が増えるほど予測成功率が低下する．未加工データの場合，$2 \leq n \leq 14$ で予測成功率が直線的に低下しており，$n = 15$ になった時に予測成功率が 50% に近づくことがわかる．多数決データの場合，$2 \leq n \leq 16$ で予測成功率が低下しており，$n = 17$ になった時に予測成功率が 50% に近づく．また，多数決データの方が未加工データデータを使用した場合よりも全体的に予測成功率が高くなっている．未加工データをトレーニングデータおよびテストデータに利用した場合，予測成功率が最も低くなる．これはテストデータに環境ノイズによるエラービットが含まれており，エラービットを予測できなかったためである．そのため，未加工データセットを利用したトレーニングデータに対して，2 種類のテストデータの結果を比較

表 6.3　各 n-XOR PUF の Bit Error Rate (BER) (%)

n	2	3	4	5	6	7	8	9	10
BER	3.28	4.72	6.00	7.42	8.93	10.16	11.35	12.55	13.79
11	12	13	14	15	16	17	18	19	20
14.96	16.02	17.14	18.40	19.46	20.49	21.39	22.23	22.95	23.62

すると，多数決データをテストデータに用いた時の方が，全体的に予測成功率が高い値になっていることがわかる．13-XOR PUF の時，その差が最も開き，未加工データを利用したトレーニングデータに対して，未加工データによるテストデータでは予測成功率が 75.98%，多数決データによるテストデータでは 80.27% となり，4.29% の差が開いた．未加工データを利用したトレーニングデータでは，n が 15 になった時に n が 14 の時に比べて予測成功率が大きく下がった．そして，n が 17 以上では予測成功率が 50% 前後となった．多数決データを利用したトレーニングデータでは，n が 17 になった時に n が 16 の時に比べて予測成功率が大きく下がり，n が 19 以上では予測成功率が 50% 前後となった．

表 6.3 に，各 n-XOR PUF の Bit Error Rate (BER) を示す．BER は次の式で導出できる．

$$BER = \frac{|\sum_{i=1}^{b}\sum_{k=1}^{t} r_{i,k} - r'_i|}{b \times t} \tag{6.1}$$

ここで，$r_{i,k}$ は t 回のうち k 回目の試行における b ビット中 i ビット目のデータのレスポンス，r'_i は多数決をとった b ビット中 i ビット目のレスポンスを指す．レスポンスが安定しておらずランダムの場合には，この式で表すと BER は 50% となる．表 6.3 からわかるように，BER は n の値が大きくなればなるほど線形に増加していることがわかる．2-XOR PUF の時，BER は 3% 程度だったが，16-XOR PUF の時に 20% を超える．これは，255 ビット認証を考えた

表 6.4　各 n-XOR PUF におけるレスポンス 1 の頻度 (%)

n	2	3	4	5	6	7	8	9	10
freq.	42.10	49.87	43.77	51.04	45.27	51.81	45.87	51.79	47.90
11	12	13	14	15	16	17	18	19	20
50.21	48.15	50.24	48.33	50.22	48.53	50.14	48.65	50.14	48.70

時に全体の約 5 分の 1 に相当する 50 ビットが環境ノイズによって反転していることを示している．BER の結果と，トレーニングデータが未加工データセットおよび多数決データセットの場合を比較すると，その差が線形に広がっていることがわかる．ただし，n に対するトレーニングデータによる予測成功率の差の広がりは BER の差と比べると小さいことがわかる．

n=15 の時は，未加工データを利用したトレーニングデータでは予測成功率が大きく下がったが，n=16 の時は n=15 よりも予測成功率が高くなっている．そして，n=17 では，両方のデータで予測成功率が低下しているが，n=18 の時に，多数決データを用いたトレーニングデータでは予測成功率が上昇している．このような結果となる理由は，明確にはわかっていない．しかし，文献 [64] にて，n の値が偶数の n-XOR PUF のレスポンスには偏りが生じやすいことが述べられている．表 6.4 に，各 n-XOR PUF におけるレスポンス 1 が出る頻度を示す．表から n の数が増えるほど偏りが少しずつ小さくなっている．これは n が増えるにしたがって，レスポンスに複数の Arbiter PUF の偏りが含まれるようになり，多数決が偏りの影響を改善させたためと考えられる．また，文献 [64] で述べられているように，n の値が偶数の n-XOR PUF では体系的にレスポンスに偏りが生じている．さらに，n が奇数の場合と，偶数の場合それぞれで比較をすると n の増加に伴って偏りが改善しているのがわかる．

6.3 考察

今回の結果から，並列実装されたハードウェアインスタンスにより，クローンの予測成功率を低下させることができた．また，環境ノイズによるビット反転の影響は，深層学習による安全性評価への影響が小さいことがわかった．*n*-XOR PUF では *n* の値が大きくなればなるほど BER も大きくなる．しかし，BER が予測成功率に与える影響は小さく，深層学習ではノイズの影響を低減した学習が可能だと考えられる．例えば，14-XOR PUF の BER は 18.4% のため，未加工データのレスポンスには約 18% のビット反転したレスポンスが含まれている．14-XOR PUF に対する予測成功率の結果は，多数決データセットで 81% 程度，データセットで 78% 程度となっている．従って，その差は約 3% となり，環境ノイズの影響はそこまで大きくないことがわかる．

また，IC チップに実装された *n*-XOR PUF には偏りが生じ，*n* が偶数の時にレスポンスの偏りが大きくなることが見られた．原因としては Arbiter 回路に特有の遅延が生じていたと考えられるが，具体的な要因分析はできていない．現状の理解としては，*n* が偶数である *n*=16 および *n*=18 の時に予測成功率が上がったのはレスポンスに偏りが生じ，深層学習影響を及ぼしたためと考えられる．

第 5 章で取り扱ったシミュレーション実装の *n*-XOR PUF では，$n = 6$ になった時点で予測成功率が 50% まで低下した．しかし，実チップでは $n = 6$ で約 90% であり，シミュレーションの結果とは大きく異なった．これは，シミュレーション実装の場合，各段で生じる遅延時間差に第 3.4 章で説明したような支配的な遅延時間差が存在せず，推定パラメータ数が減少しなかったためと考えられる．実際にチップ上に実装された PUF の回路レイアウトを確認したところ，128 段 Arbiter PUF が 16 段ごとで折り返されていたため，配線長が長くなり第 3.4 章

で説明したような支配的な遅延時間差が生じ，予測成功率に影響を与えたと考えられる．しかし，PUF は耐タンパー性を有しており，実装された PUF の内部を分析することは困難なため，支配的な遅延時間差が発生している箇所を特定することはできない．また，今回実験に用いた Arbiter PUF は性能評価では優れた値を示しており，PUF として利用する上で問題がないと評価されていた．このことから，シミュレーション実装の PUF では安全性評価の結果が高い理想的な状態を再現しやすく，実際の PUF を用いた評価の重要性が明らかとなった．

6.3.1 モンテカルロシミュレーションを用いた認証システムに対する考察

今回の安全性評価の結果を認証システムに適用した時の False Acceptance Rate (FAR) および False Rejection Rate (FRR) を示す．FAR は他人受入率と呼ばれる別人を本人と誤って認証する確率のことを指す．FRR は本人拒否率と呼ばれる本人を別人と誤って拒否する確率のことを指す．今回の場合，本人は正規 PUF，別人はクローンのことを表す．FAR と FRR は認証の閾値を設定する時に役立つ．具体的には FAR と FRR が交差している場合には正規 PUF とクローンの判別が困難であることを示す．また，交差していない場合には FAR 及び FRR が 0 になった範囲で閾値を設定すると正規 PUF とクローンの判別が可能なことを示す．

今回の安全性評価の結果から，n =2，14，17 の FAR と FRR を導出し，その結果を図 6.3 に示す．認証では 255 ビットのレスポンスを用いることを想定し，FAR はクローンの予測成功率から導出し，FRR は BER を用いて導出する．クローンの予測成功率および BER は 1 ビットに対する確率のため，255 回試行することで 1 回の認証に対する PUF のレスポンスとした．認証は 100 回行い，ある閾値に対する FAR と FRR を導出した．図の横軸は，255 ビット中保存してあるレスポンスと同じ値になったビット数，つまり正答数である認証閾値を示す．ある認証閾値を設定した時に誤認証が起きる確率を縦軸に示す．

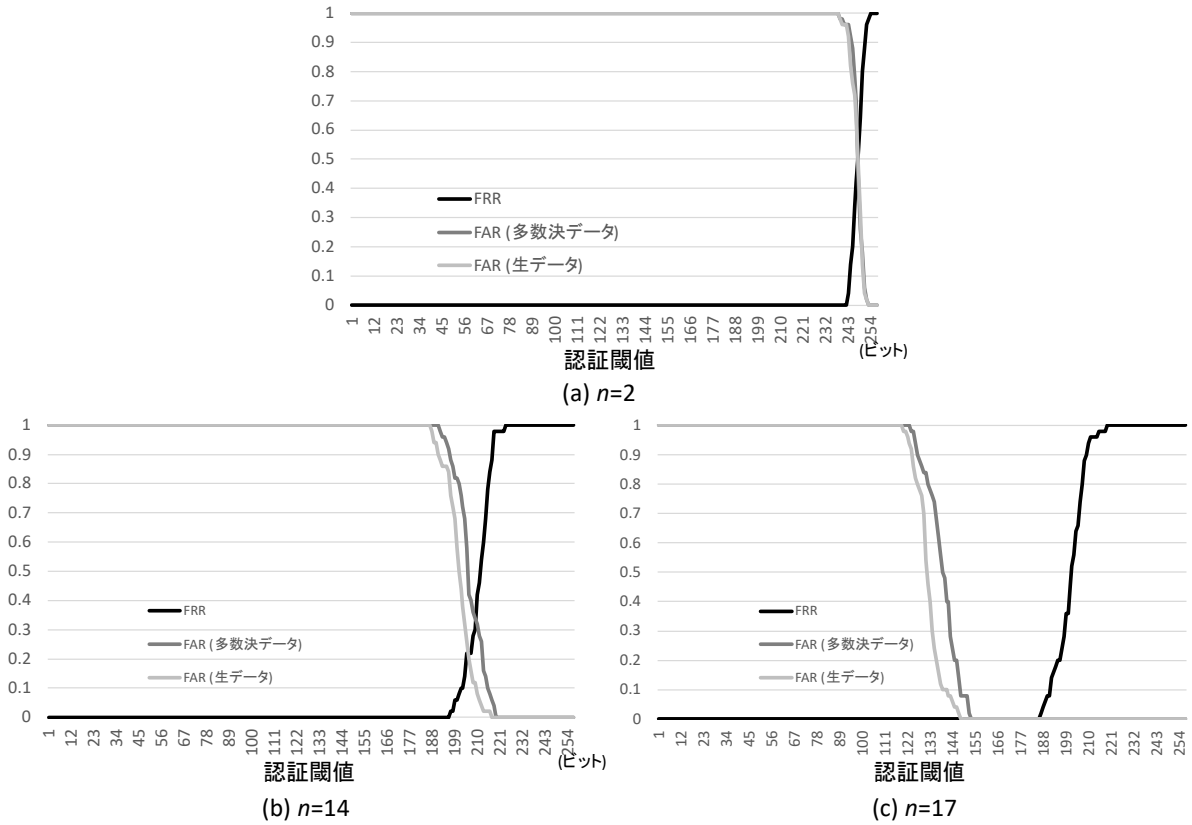

図. 6.3　(a) n=2 の時，(b) n=14 の時，(c) n=17 の時の FAR および FRR

n=2 の時，多数決データおよび未加工データの 2 つのクローン共に FAR と FRR は 0.5 付近で交差している．これは正規 PUF とクローンを識別するのが困難であることを示す．具体的には，247 ビットに閾値を設定した時，正規品の PUF を 36% の確率で拒否，クローンを 51% または 46% の確率で誤認証する．誤認証を防ぐことを目的とすると，今回の実験では 253 ビット以上に閾値を設定するとクローンを全て拒否することが可能だが，その際に 97% の正規 PUF も拒否することになる．そのため，n=2 では閾値によってクローンの誤まって受け入れされることを防ぐことが可能だが，正規 PUF とクローンの識別は困難である．

n=14 の時，FAR と FRR は未加工データを用いたクローンでは 206 ビットに閾値を設定した時に 0.25 付近，交多数決データを用いたクローンとは 208 ビットに閾値を設定した時に 0.4

で交差している．この場合も正規 PUF とクローンを識別することは困難となる．未加工データを用いたクローンは，216 ビット以上に閾値を設定することで拒否することは可能だが，少なくとも 90% の確率で正規 PUF を拒否する．多数決データを用いたクローンは，219 ビット以上に閾値を設定することで拒否することが可能だが，少なくとも 96% の確率で正規 PUF を拒否する．そのため，n=14 では正規 PUF とクローンの識別は困難である．

n=17 の時，FAR と FRR は交差していない．この場合，正規 PUF とクローンを識別することが可能となる．未加工データを用いたクローンは，147 ビット以上に閾値を設定すると拒否することが可能である．多数決データを用いたクローンは，152 ビット以上に閾値を設定すると拒否することが可能である．また，185 ビット以下に閾値を設定すると，正規 PUF の拒否率を 0 にすることが可能である．つまり，152 ビット以上 185 ビット以下に閾値を設定することで正規 PUF とクローンを識別することが可能となる．

6.4 まとめ

本章では並列実装された PUF に対して安全性評価を行った．並列数 (n) が増えれば増えるほど，攻撃者の予測成功率が下がることがわかった．つまり，並列実装により PUF の安全性が向上することが実験的に示された．ただし，n-XOR PUF では n の値が偶数か奇数かによってレスポンス値に偏りが見られ，n を大きくしても予測成功率が高くなることがあった．

また，第 5 章にて時系列処理は予測成功率を上げることが明らかにされている．本章で取り扱った，多数決データでは環境ノイズの影響を緩和させるためには繰り返しという時系列処理を行っている．未加工データと多数決データを比較した結果，本章でも同様に，時系列処理を行った場合に，予測成功率が上がることを明示できた．

今回の実験結果から，シミュレーションと実際の実装では予測成功率に違いが生じる場合が

あることがわかった．シミュレーションの結果では推定パラメータ数の減少等も起きず予測成功率が低くならず攻撃者にとって不利な状況となっていた．そのため，理想的なシミュレーション PUF を安全性評価に用いることは認証システムの安全性を過大評価しかねない．今後，安全性評価を行うには実際に実装された PUF の評価結果をシミュレーションにフィードバックするなどし，実際の PUF の動作に近づける工夫が必要と考える．

第 7 章

安全性評価 3: 意図的なエラーを用いた認証

7.1　はじめに

深層学習を用いることにより，複雑な構成をした Arbiter PUF に対してクローニング攻撃が可能であることが報告されている [6, 23, 67, 74]．そこで，クローニング攻撃に対する対策として意図的なエラーを用いた認証方式の提案をする．Arbiter PUF のレスポンスは，環境ノイズの影響を受け，ビット反転することがある．環境ノイズを利用した PUF の認証は先行研究で提案されており [19, 22]，環境ノイズが攻撃者の予測成功率を下げることも報告されている [63]．今回用いる「意図的なエラー」はこの環境ノイズとは異なり，出力されるレスポンスに対して意図的にビット反転をすることを意味する．環境ノイズによってビット反転するレスポンスは，2 信号の遅延時間差が 0 に近い場合である．すなわち，遅延時間差の絶対値が小さくなるため，内部の遅延時間差の推定に与える影響は小さくなると考えられる．一方，意図的

なエラーの場合は遅延時間差の値によらず，任意のレスポンスに対して注入されるため，内部の遅延時間差の推定に影響すると考えられる．そこで本章では意図的なエラーをわざと注入することにより，クローンの精度がどのくらい変化するのか，またその結果が認証結果にどのように影響を及ぼすのかを検証する．

具体的には，深層学習を用いた安全性評価を意図的にエラーを付与された PUF のレスポンスを含む CRP に対して行い，深層学習を用いたクローニング攻撃への耐性を向上することが可能なことを示す．

7.2 提案する意図的なエラーを用いた認証システム

意図的なエラーを用いたシステムとして，チャレンジレスポンス認証システムと鍵共有システムの 2 方式が考えられる．この 2 つでは，想定するべき攻撃者の条件が大きく異なる．ここでは，それぞれのシステムと想定される攻撃シナリオについて説明する．なお，本章では 255 ビットレスポンスで認証を行うことを想定する．

7.2.1 チャレンジレスポンス認証

本論文で扱う意図的なエラーを用いたチャレンジレスポンス認証は，図 7.1 に示すような PUF を用いたシンプルな認証方式である．チャレンジレスポンス認証の認証フローは次の通りである．

1. 検証者は，事前に PUF からレスポンス (r) を複数取得しておき，データベースに保存しておく （登録フェーズ）
2. 検証フェーズでは，検証者はデータベースから無作為に CRP を複数選択し，チャレン

ジ (c) を被認証者に送信する．

3. 被認証者は，受け取ったそれぞれのチャレンジ (c) を用いて PUF を動作し，レスポンス （r'）を取得する．
4. **取得したレスポンス （r'）に対して意図的なエラー (e) を注入する．**
5. 被認証者は，検証者に意図的なエラーを注入したレスポンス ($r' + e$) を送信する．
6. 検証者は，保存してあるレスポンス (r) と受け取ったレスポンス ($r' + e$) を比較し，異なるビット数が閾値以下であれば認証が成功する．

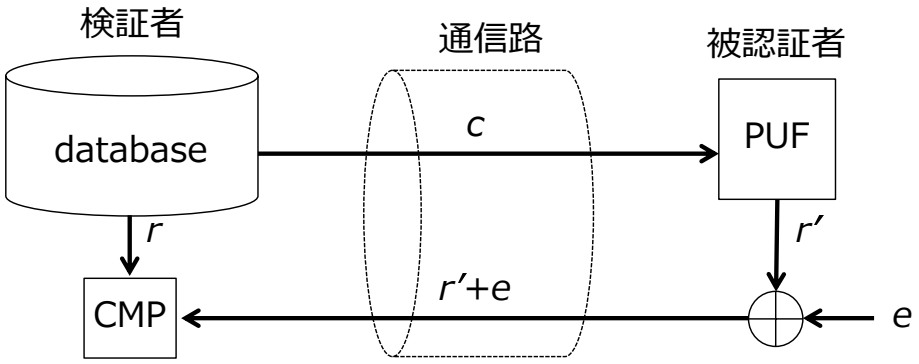

図. 7.1　意図的なエラーを利用したチャレンジレスポンス認証方式

模倣品を流通させるために，自分が発行する模倣 PUF を正規品と誤認識させるのが攻撃者の目的である．チャレンジレスポンス認証において，その目的達成のための最もシンプルな攻撃手法は中間者攻撃である．中間者攻撃は攻撃者が検証者と被検証者間の通信に干渉可能な状況で起こりうる．攻撃者は検証者に対して認証リクエストを送信しチャレンジ (c) を受け取る．次に，受け取ったチャレンジを被検証者に送信し，レスポンス ($r' + e$) を取得する．取得したレスポンス ($r' + e$) を検証者に送信すれば攻撃者は被検証者として不正に認証される．このように，中間者攻撃では，攻撃者は常に自分が発行した模倣品 PUF の通信を傍受し，模倣

品 PUF で認証プロセスが始まったら正規 PUF への通信を行い，受け取ったレスポンスを模倣品 PUF へ渡すことになる．模倣品の流通という目的では，攻撃者は常に流通させた膨大な模倣品の通信を監視することになるため，攻撃の実現は困難である．

他の脅威として考えられる攻撃手法としてクローニング攻撃がある．クローニング攻撃を行う攻撃者は検証者と被検証者間の通信路を盗聴し，チャレンジとレスポンスのペア（c と $r'+e$）を取得する．取得したチャレンジとレスポンスのペアからクローンを作製する．クローンの精度が高ければ，未知のチャレンジに対するレスポンスを予測可能となる．精度の高いクローンを正規 PUF の代わりに実装すれば，誤認証される可能性がある．そこで本論文ではより大きな脅威と考えられる深層学習を用いたクローニング攻撃に注目する．

7.2.2 鍵共有システム

PUF には，チャレンジレスポンス認証の他に秘密鍵として用いる利用方法がある．図 7.2 に意図的なエラーを用いた鍵共有システムを示す．第 1 章で述べたように，鍵共有システムは Fuzzy Extractor [**?**] や Reverse Fuzzy Extractor [20] を用いるのが一般的である．鍵生成において，1 ビットでも異なる値が鍵生成に利用されると同じ鍵が共有ができない．そのため，PUF のように環境ノイズの影響を受けやすいプリミティブを基に鍵生成をする場合には，誤り訂正技術が必要になる．ここで，ヘルパーデータを用いてデータの誤りを訂正する技術を Fuzzy Extractor と呼ぶ．またここでは，被認証者の回路コストが小さくなる Reverse Fuzzy Extractor をベースとし，誤り訂正符号には BCH 符号を用いる．鍵共有システムの認証フローは次の通りである．

1. 検証者は，事前に PUF からレスポンス (r) を複数取得しておき，データベースに保存し

ておく（登録フェーズ）

2. 検証フェーズでは，検証者はデータベースから無作為に CRP を複数選択し，チャレンジ (c) を被認証者に送信する．
3. 被認証者は，受け取ったそれぞれのチャレンジ (c) を用いて PUF を動作し，レスポンス（r'）を取得する．
4. **取得したレスポンス（r'）に対して意図的なエラー (e) を注入する．**
5. 被認証者は，意図的なエラーを注入したレスポンス ($r' + e$) で秘密鍵 (K) を生成すると同時に BCH 符号などを用いてヘルパーデータ (h) を生成する．
6. 被検証者は検証者に対してヘルパーデータ (h) を送信する．
7. 検証者は保存してあるレスポンス (r) に対してヘルパーデータ (h) を用いた誤り訂正を行い，意図的なエラーを注入したレスポンス ($r' + e$) を導出する．
8. 最後に，意図的なエラーを注入したレスポンス ($r' + e$) から秘密鍵 (K') を導出する．この時，保存してあるレスポンス (r) と意図的なエラーを注入したレスポンス ($r' + e$) が誤り訂正能力の範囲であれば $K = K'$ となり，鍵共有が可能となる．

鍵共有システムに対する攻撃手法として，チャレンジレスポンス認証と同様に中間者攻撃が考えられる．しかし前述の通り，中間者攻撃では攻撃者が模倣品 PUF を常に監視する必要があるため，攻撃者の負担が大きく，攻撃困難と考える．そこで，深層学習を用いたクローニング攻撃を想定する．意図的なエラーを注入したレスポンス ($r' + e$) を取得するために，攻撃者は PUF を実装しているハードウェア基板に対して物理攻撃を行う必要がある．

今回の攻撃シナリオでは，PUF および意図的なエラー付与部分が耐タンパーチップ内で保護されているものとする．さらに，各 CRP の使用は 1 回とする．あるいは，鍵生成部分が耐

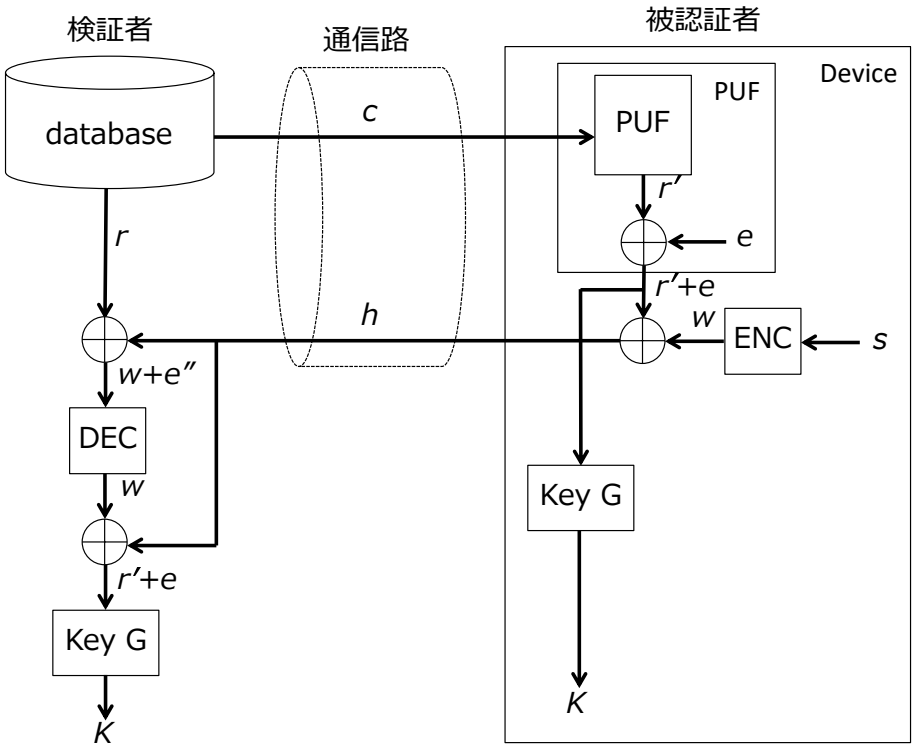

図. 7.2　意図的なエラーを利用した鍵共有システム

タンパー性チップ内に実装されているとを想定する．ただし，攻撃者は基板上のデータバスの盗聴などにより意図的なエラーを注入したレスポンス ($r' + e$) は取得することが可能である．

PUF が，上述のような耐タンパー性チップのような物理攻撃対策なしで実装されていた場合，攻撃者は物理攻撃によって PUF のレスポンス (r') に直接アクセスするのが自然である．PUF のレスポンス (r') にアクセスが可能であれば意図的なエラーの影響を受けずにより予測成功率の高いクローンの作製が可能となるためである．

鍵生成部分が耐タンパー性チップのような物理攻撃対策なしに実装されていた場合，攻撃者は秘密鍵 (K) にアクセス可能である．もし，同じ CRP が複数回使いまわされた場合には，攻撃者はクローンを作製することなく，秘密鍵 (K) を取得可能となる．鍵生成部分が耐タンパー性チップ上に実装されていたとしても攻撃者は取得した意図的なエラーを注入したレスポ

ンス ($r' + e$) を用いて秘密鍵 (K) の作成が可能なためである．そのため，今回のシナリオでは CRP の使用は一回きりという条件が必要になる．

PUF チップと鍵生成回路は別のラインで製造されるが，同じ基板上に実装されることがある．この場合，攻撃者は PUF チップと鍵生成回路間のバスを盗聴することが可能であり，今回のシナリオでは意図的なエラーを注入したレスポンス ($r' + e$) を取得可能である．

7.3 安全性評価

7.3.1 シミュレーション PUF

今回，MATLAB を用いて遅延時間差を再現した 128 段 n-XOR PUF(n=2–6) を用いる．

Santikellur らは文献 [55] で Sahoo らが提案したシミュレーション PUF に対してパラメータの値が小さな深層学習にて攻撃可能と報告している．Sahoo らによるシミュレーション PUF [54] は，図 7.3 に示すようにチャレンジビットによる遅延時間差が考慮されておらず，特殊な条件下の PUF である ($\delta_l^0 = \delta_l^1$)．しかし，実際の PUF ではチャレンジビットによって伝搬経路が異なるため，遅延時間差がわずかに異なる （$\delta_l^0 \neq \delta_l^1$)．チャレンジビットに関係なく遅延時間差が同じになる可能性もあり得るが，物理特性はコントロールできず，さらに伝搬経路が異なるため可能性は低い．また第 3.4.1 章で述べたように攻撃者が推定するべきパラメータが $\frac{1}{2}$ まで減少するため攻撃者に有利な状況になる．

本論文では，MATLAB を用いてミュレーション PUF (図 7.3) を実装し，遅延時間差はチャレンジビットにより異なる場合と等しい場合の 2 つの場合を想定する．このシミュレーションでは，実際の PUF から計測した値を参考に，平均が 0 で，標準偏差が 10.75 のガウス分布から遅延時間差 (δ_l^0，δ_l^1) をランダムに決定した．また，予測する PUF のパラメータ数と安全性

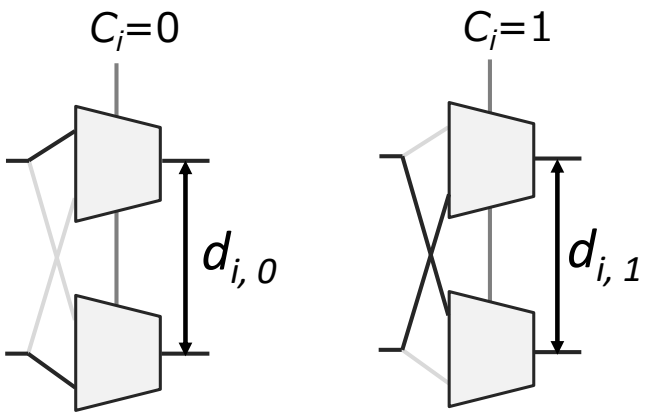

図. 7.3　各段におけるチャレンジビットによる遅延時間差

評価の影響を検証するために，Sahoo らのシミュレーション PUF のような各段における推定遅延時間差パラメータが 1 つの場合 (第 3.4.1.1 章) を想定する．ただし，Sahoo らが公開しているデータセットを利用すると，MATLAB によるシミュレーションと条件が大きく異なる可能性がある．そのため，チャレンジビットが 0 の場合の遅延時間差 (δ_l^0) の値を利用し，各段の遅延時間差を $\delta_l^0 = \delta_l^1$ とした．今回のシミュレーション PUF では環境ノイズはなく，Arbiter 回路で発生する遅延時間差が発生しないことを想定する．

シミュレーション PUF は，128 段の PUF として，各段に遅延時間差が 1 つ (δ_l) の PUF と，各段に遅延時間差が 2 つ (δ_l^0，δ_l^1) の PUF それぞれ 5 つ用意する．意図的なエラーは，BCH 符号の誤り訂正能力を基に 255 ビットのレスポンスに対して，0，7，15，23，31，42，47，55，63 ビットを確率的に注入する．

7.3.2　評価環境

安全性評価には Keras を用いた深層学習環境を用いる．Keras に利用したパラメータを表 7.1 に示す．今回利用したパラメータは第 4 章で説明した活性化関数および複数のパラメー

タを試行し，最もよい試行のパラメータを利用した．

表 7.1　安全性評価に使用した Keras のパラメータ

隠れ層	$h1$	$h2$	$h3$	$h4$	$h5$
ユニット数	5000	1000	500	200	100
活性化関数	tanh	sigmoid	tanh	sigmoid	sigmoid
Dropout	0.5	0.2	0.2	0.2	–

トレーニングデータセットには，意図的なエラーが注入されたレスポンスを含む 500,000 CRP を用い，テストデータセットには意図的なエラーを含まない 100,000 CRP を用いる．検証者が持っているデータセットは，意図的なエラーを含まないレスポンス (r) である．つまり，攻撃者はトレーニングデータ ($r + e$) ではなく，比較対象である意図的なエラーを含まない場合 (r) を予測する方が認証を通る確率が高くなる．そのため，テストデータセットは意図的なエラーを含まないレスポンス (r) とした．

7.3.3　実験結果

7.3.3.1　各段の遅延時間差が 1 つ (δ_l) のシミュレーション PUF に対するクローニング攻撃

各段の遅延時間差が 1 つ (δ_l) のシミュレーション PUF に対するクローニング攻撃の結果を図 7.4 と表 7.2 に示す．図 7.4 は，5 つの PUF に対するクローンの予測成功率の平均値を表しており，表 7.2 は 5 つのクローンのうち予測成功率の最大値と最小値を示す．クローンの予測成功率は 50% でクローンの精度が低いことを意味する．逆に 50% から値が離れた場合は，クローンの精度が高いことを意味する．仮にクローンの予測成功率が 10% だった場合，予測したレスポンスに対してビット反転を行えば 90% 当たるクローンが作製されたということにな

る．つまり，クローンの精度が最も低いのはランダムに予測した場合，すなわち予測成功率が 50% になる．表 7.2 の太字の部分は，クローンの予測成功率が 50±10% の値である．

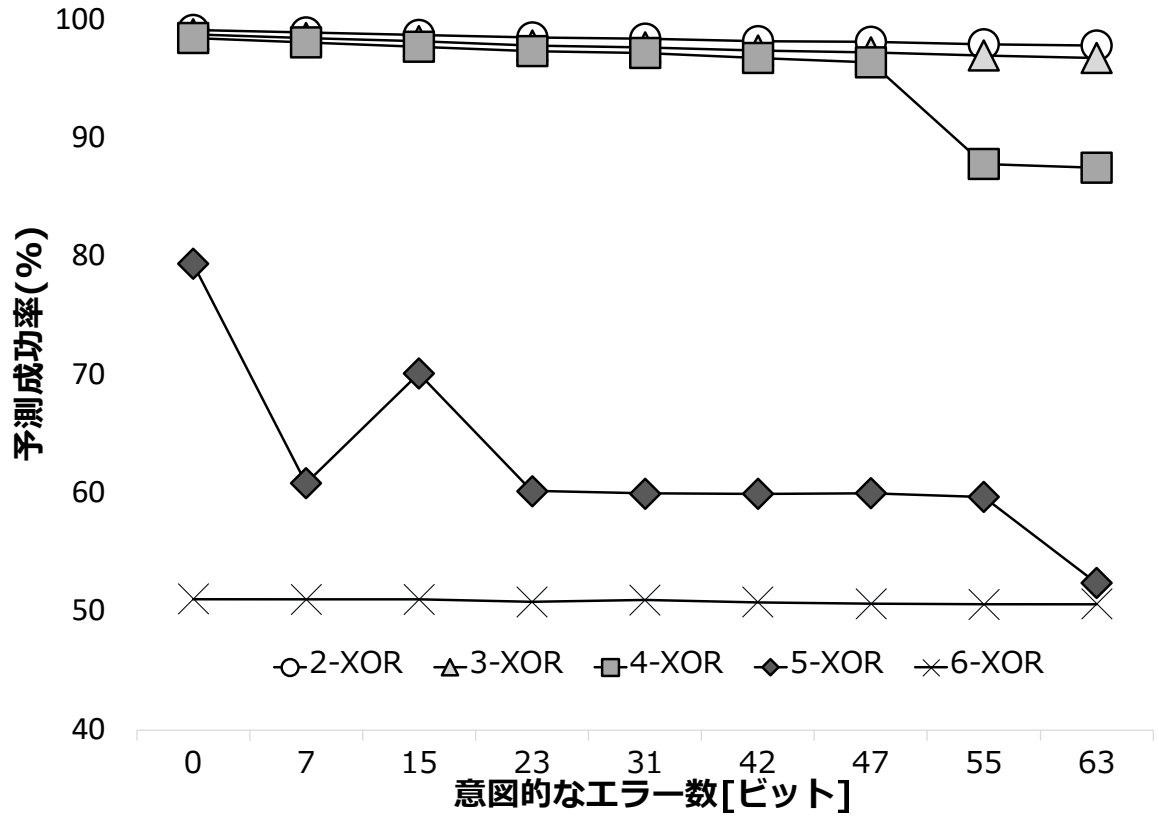

図. 7.4　各段の遅延時間差が 1 つ (δ_l) のシミュレーション PUF に対するクローニング攻撃の結果

2-XOR PUF と 3-XOR PUF では，意図的なエラーを注入してもクローンの予測成功率はほとんど低下しなかった．4-XOR PUF では，意図的なエラーを 55 ビット以上加えることで，1 つの PUF に対して予測成功率を 50% まで低下させることができた．5-XOR PUF では，意図的なエラーを注入していない場合 (0 ビット) でも予測成功率が 50% のクローンが 2 つ存在した．23 ビットの意図的なエラーを加えた時に，2 つの PUF の予測成功率は 50% まで下がったが，1 つの PUF は 63 ビットの意図的なエラーを注入するまで予測成功率が 50% まで低下しなかった．6-XOR PUF では，意図的なエラーを注入していない場合でも予測成功率が 50%

表 7.2　各段の遅延時間差が 1 つ (δ_l) のシミュレーション PUF に対するクローンの予測成功率の最大値と最小値

		0	7	15	23	31	42	47	55	63
2-XOR	max	99.20	98.98	98.77	98.61	98.57	98.41	98.23	98.15	98.10
	min	99.13	98.87	98.66	98.50	98.27	98.12	98.05	97.82	97.63
3-XOR	max	98.91	98.61	98.38	98.00	97.89	97.55	97.48	97.26	96.93
	min	98.71	98.37	98.12	97.73	97.47	97.21	97.14	96.84	96.63
4-XOR	max	98.54	98.32	97.78	97.46	97.40	96.95	96.79	96.44	96.18
	min	98.40	97.98	97.65	97.19	96.99	96.63	95.82	**54.09**	**53.64**
5-XOR	max	98.14	97.82	97.81	97.27	96.55	96.41	96.38	96.02	**59.14**
	min	**50.99**	**50.91**	**51.00**	**50.62**	**50.22**	**50.34**	**50.37**	**50.32**	**50.30**
6-XOR	max	**51.50**	**51.34**	**51.35**	**51.13**	**51.52**	**51.33**	**51.08**	**51.21**	**50.96**
	min	**50.81**	**50.80**	**50.90**	**50.14**	**50.71**	**50.46**	**50.47**	**50.15**	**50.18**

であった.

7.3.3.2　各段の遅延時間差が 2 つ (δ_l^0，δ_l^1) のシミュレーション PUF に対するクローニング攻撃

各段の遅延時間差が 2 つ (δ_l^0，δ_l^1) のシミュレーション PUF に対するクローニング攻撃の結果を図 7.5 と表 7.3 に示す.

2-XOR PUF と 3-XOR PUF では，各段の遅延時間差が 1 つ (δ_l) のシミュレーション PUF と同様に意図的なエラーを注入してもクローンの予測成功率はほとんど低下しなかった．4-XOR

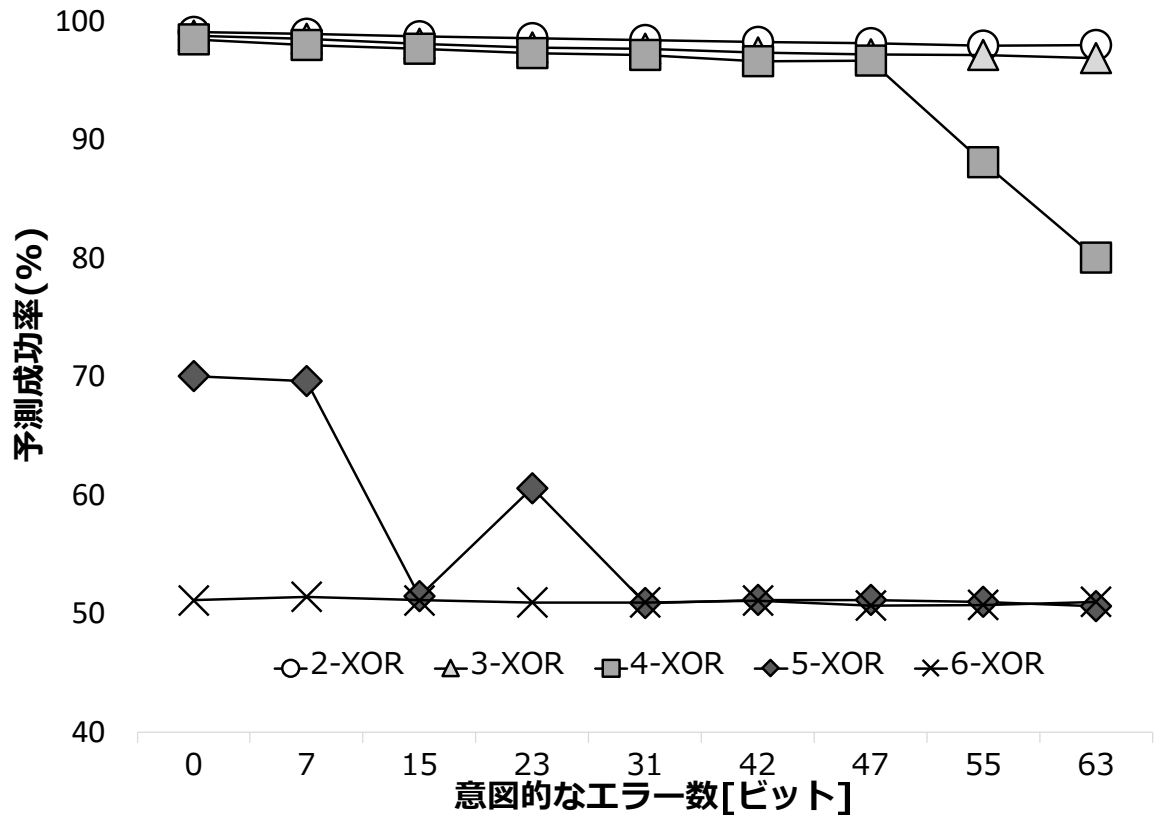

図. 7.5　各段の遅延時間差が 2 つ (δ_l^0，δ_l^1) のシミュレーション PUF に対するクローニング攻撃の結果

PUF では，各段の遅延時間差が 1 つ (δ_l) のシミュレーション PUF と同様に意図的なエラーを 55 ビット注入することで 1 つの PUF に対して予測成功率が 50% まで低下させることができた．また，意図的なエラーを 63 ビット注入した場合には，予測成功率が 50% 近くまで低下するクローンが 2 つあった．5-XOR PUF では，意図的なエラーを注入していない場合でも予測成功率が 50% のクローンが 3 つ存在した．また，意図的なエラーを 31 ビット以上注入した場合，5 つ全てのクローンにおいて予測成功率が 50% となった．6-XOR PUF では，各段の遅延時間差が 1 つ (δ_l) のシミュレーション PUF と同様に意図的なエラーを注入していない場合でも予測成功率が 50% であった．

表 7.3　各段の遅延時間差が 2 つ (δ_l^0，δ_l^1) のシミュレーション PUF に対するクローンの予測成功率の最大値と最小値

		0	7	15	23	31	42	47	55	63
2-XOR	max	99.20	99.09	98.79	98.77	98.52	98.40	98.43	98.13	98.04
	min	99.01	98.83	98.67	98.45	98.39	98.07	97.93	97.76	97.96
3-XOR	max	98.85	98.65	98.32	97.99	97.90	97.58	97.55	97.30	97.08
	min	98.72	98.30	97.65	97.44	97.26	97.15	97.04	97.01	96.80
4-XOR	max	98.54	98.30	98.03	97.53	97.30	96.92	96.82	96.62	96.20
	min	98.42	97.37	97.11	96.96	96.98	96.41	96.51	**55.68**	**56.11**
5-XOR	max	98.25	97.89	**51.83**	97.14	**51.64**	**51.25**	**51.63**	**51.64**	**51.26**
	min	**51.03**	**50.65**	**51.14**	**50.70**	**50.26**	**50.96**	**50.70**	**50.21**	**50.00**
6-XOR	max	**51.26**	**51.89**	**51.55**	**51.26**	**51.66**	**51.90**	**51.37**	**51.51**	**51.69**
	min	**50.85**	**50.85**	**50.82**	**50.52**	**50.28**	**50.64**	**50.15**	**50.01**	**50.33**

7.4　考察

実験の結果から，クローンの予測成功率は大きく 3 種類あることがわかる．1 つ目が意図的なエラーの注入に関わらず予測成功率が 50% の場合，2 つ目が意図的なエラーの注入にかかわらず予測成功率が 95% 以上を維持する場合，そして意図的なエラーのビット数によって予測成功率が 50% に低下する場合である．

まず，意図的なエラーに関わらず予測成功率が 50% である 6-XOR PUF の場合，正規品とクローンを判別可能な確率が高い．6-XOR PUF は，今回の想定の中で最も n が大きいため，

他の PUF よりも環境ノイズの影響を受けるレスポンス数が多くなる．認証の閾値は，正規品を認証成功させるために環境ノイズを考慮して余裕を持って設定する必要がある．許容誤りを増やすということは，クローンのレスポンス予測が誤る場合も許容するということになる．クローンの予測成功率はテストデータから得られる平均値のため，実際には予測成功率である 50% を平均とした二項分布に従う．つまり，255 ビット中，許容誤りビット数 (認証閾値) を 127 ビット以上にするとクローンを誤認証する確率が高くなる．意図的なエラーを注入により，認証閾値は環境ノイズと意図的なエラーの 2 つを考慮する必要がある．つまり，許容誤りを増やす必要があるため，クローンを誤認証する可能性を上げないために意図的なエラーは使用しない方が良いと考えられる．

次に，意図的なエラーを注入しても予測成功率が 95% 以上のまま変化が得られない 2-XOR や 3-XOR PUF の場合，意図的なエラーを加えることでクローンを誤認証しやすくなる．意図的なエラーを使用しなくてもクローンの予測成功率が約 99% であることから，255 ビット中 3 ビットほどしかレスポンスを誤らない．そのため，環境ノイズの影響が多い PUF では正規品とクローンを判別できない．意図的なエラーを加える場合，認証の閾値を意図的なエラーのビット分低く設定する必要がある．クローンの予測成功率は，255 ビット中 63 ビットの意図的なエラーを注入しても 2%，5 ビットほどしか下がらない. そのため，意図的なエラーの 63 ビットを許容誤りの閾値とすると，5 ビットしかレスポンス予測を誤らないクローンでは容易に誤認証される．ただし，意図的なエラーを今回実験を行なった 63 ビットよりも増やすことでクローンの予測成功率が下がる可能性がある．もしも意図的なエラーを 63 から 127 ビットに増やすことで，クローンの予測成功率が 50% まで下がる場合，2-XOR や 3-XOR PUF に対しても意図的なエラーを加えることが有効となる．

最後に, 意図的なエラーのビット数によって予測成功率が 50% に低下する 4-XOR や 5-XOR

PUF の場合，意図的なエラーを含ませることで正規品とクローンを判別可能となる．4-XOR PUF では，表 7.2 や表 7.3 に示すように，98% 以上の予測成功率だったクローンが，意図的なエラーを加えることで 50% 近くまで予測成功率が下がる場合がある．クローンの予測成功率はランダムな場合でも 50% ほどになる．そのため，許容誤りの認証閾値は 255 ビット中 127 ビットが最大値となる．そこで，環境ノイズの影響を受けるレスポンスが 127 ビットから意図的なエラーの 63 ビットを引いた 64 ビット以下であれば，当該クローンは正規品の PUF と識別可能である．また，63 ビットの意図的なエラーで予測成功率が 50% になるクローンが複数存在するため，さらに意図的なエラーを増やせば 50% の予測成功率であるクローンが増えると想定できる．5-XOR PUF では，31 ビットの意図的なエラーを注入することで，クローンの予測成功率が 50% まで低下する．そのため，環境ノイズの影響を受けるレスポンスが 89 ビット以下であれば，正規品とクローンを識別できることが示唆された．

7.4.1 モンテカルロシミュレーションを用いた認証システムに対する考察

PUF を用いた認証システムにおいて，クローンが誤まって受け入れられる確率を、モンテカルロシミュレーションを用いて考察した。今回の結果では，意図的なエラーやクローン作製に対して BER の影響は考慮されていない．そこで，ここではクローンの予測成功率のみ注目する．

今回の結果の内，各段の遅延時間差が 2 つ $(\delta_l^0,\ \delta_l^1)$ のシミュレーション PUF のクローニング攻撃の結果である表 7.3 を用いる．想定する認証システムは 255 ビットとし，n =3，5，6 における最も予測成功率が高い予測成功率を用いた．図 7.6 は，255 ビット中で許容される誤りビット数を設定した時に，クローンが誤認証される確率を示す．図の横軸は 255 ビット中許容できる誤りビット数の認証閾値を示す．縦軸は，ある閾値を設定した時にクローンが誤認証

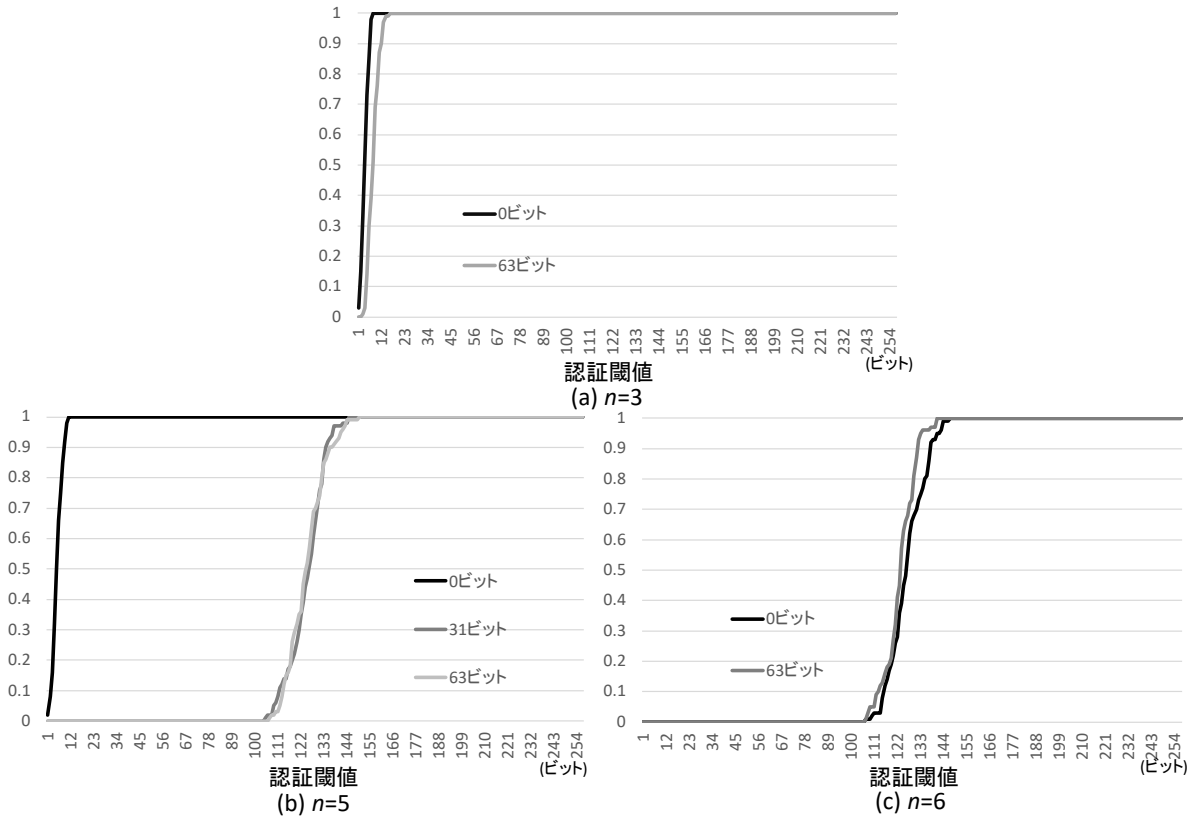

図. 7.6　認証閾値 (許容誤りビット数) における (a) n=3 の時，(b) n=5 の時，(c) n=6 の時にクローンが誤認証される確率

される確率を示す．認証システムは，正規品を認証するために，意図的なエラ―のビット数によって，許容できる誤りビットの閾値を設定する必要がある．

n=3 の時，意図的なエラーのビット数を増やしてもクローンの予測成功率は高く変化が起きなかった．意図的なエラーが 0 ビットの時，255 ビット中 254 ビット予測成功する試行があった．意図的なエラーが 63 ビットの時には，255 ビット中 252 ビット予測成功する試行があった．意図的なエラーを 63 ビット注入した場合，正規品の誤りビット数を 63 ビット許容する必要がある．クローンは 4 ビットしか誤らないため正規品として誤認証される．また，今回は環境ノイズがない環境で実験を行っているため，正規品のレスポンスは 255 ビット全て正答する

が，実際には環境ノイズにより誤りビットが存在する．そのため，意図的なエラーが 0 ビットであったとしてもクローンと正規品の判別は難しい．

n=5 の時，意図的なエラーの数を増やすことによって予測成功率は低下した．意図的なエラーが 0 ビットの時，255 ビット中 254 ビット予測成功する試行があった．意図的なエラ―が 31 ビットの時，最も良い試行では 255 ビット中 147 ビット予測成功した．意図的なエラ―が 63 ビットの時には，255 ビット中 145 ビット予測成功する試行が最も良い試行となった．意図的なエラーが 0 ビットの時は n=3 の時と同様に正規品とクローンを判別することは難しい．意図的なエラ―が 31 ビットの時，正規品を認証するために閾値を 31 に設定すると，クローンは 108 ビット誤ったため閾値と 77 ビットの差が生じる．意図的なエラ―が 63 ビットの時，クローンは 111 ビット誤ったため閾値を 63 に設定すると，48 ビットの差が生じる．意図的なエラーが 31 ビットおよび 63 ビットの場合，正規品とクローン間の誤りビット数の差が大きいため，判別が可能である．ただし，意図的なエラーを 31 ビット注入した際にクローンの予測成功率が十分に下がっている．そのため，意図的なエラーを 63 ビット注入した場合よりも 31 ビット注入した方が，正規品とクローンの差が大きくなるため，意図的なエラーを 31 ビット注入する方が適している．

n=6 の時，意図的なエラーの数を増やしてもクローンの予測成功率は低いままであった．そのため，今回使用した 6-XOR PUF では意図的なエラーを注入しない方が正規品とクローンの判別が容易である．

7.5 まとめ

本章では，意図的なエラーを注入したレスポンスが予測成功率に与える影響を調査した．各段の遅延時間差が 1 つ (δ_l) の場合と各段の遅延時間差が 2 つ (δ_l^0，δ_l^1) の場合の比較では，結

果が大きく変わることはなかったが，各段の遅延時間差が 2 つ (δ_l^0，δ_l^1) の方が予測成功率が下がることがわかった．これにより，予測に必要なパラメータ数は予測成功率に影響を及ぼすことが実験的に示された．

2-XOR PUF や 3-XOR PUF では，意図的なエラーを 255 ビット中 63 ビット，すなわち約 25% のエラーを注入しても予測成功率が 3% ほどしか下がらなかった．4-XOR PUF では意図的なエラーを 255 ビット中 63 ビット，すなわち約 22% 注入することで，予測成功率が 50% 程度まで下がる PUF があったが，63 ビットのエラーを注入してもすべての PUF で 50% まで下がることはなかった．5-XOR PUF では，意図的なエラーを 63 ビット注入することで，すべての PUF の予測成功率を 50% 程度まで下げることができた．これらの結果から，意図的なエラーを注入することによって予測成功率を大きく下げることができるような n-XOR PUF の n の範囲が存在することがわかった．範囲内である 4-XOR，5-XOR PUF では，並列実装によりハードウェアコストを上げなくても意図的なエラーを注入することで，認証システムにおいて利用可能であることを示した．

第 8 章

まとめと今後の展望

8.1　まとめ

本論文の第 1 章では，本研究のモチベーションと研究背景をまとめた．IoT 機器の普及とともに，セキュリティインシデントが発生しており，IoT 機器に対するセキュリティ上の懸念は高まっている．IoT 機器は，日々の生活で使われる機会が多く，プライバシー情報を取り扱うことが多い．そのため，IoT 機器の認証はユーザのセキュリティを高める上で必要不可欠である．模倣品流通を防ぐ技術の中で，IoT 機器に対しても有効な技術に PUF がある．本論文では，PUF を用いたシンプルな認証方式であるチャレンジレスポンス認証方式について注目した．

第 2 章では，先行研究を調査した．Extensive PUF として分類される Arbiter PUF，*n*-XOR PUF，DAPUF および RG-DTM PUF に注目した．代表例である Arbiter PUF は，回路上で発生する遅延時間をベースにした PUF である．同じ構成をした回路であっても，製品が異なれ

ば配線長や閾値電圧の違いによって，信号伝搬時間に違いが生じる．Arbiter PUF は，その遅延時間差の違いを個体固有の値として出力する．*n*-XOR PUF は Arbiter PUF を *n* 個並列に並べた構成をしている．DAPUF は，*n*-XOR PUF のレスポンスの偏りを改善するために提案された PUF である．DAPUF は *n*-XOR PUF と似た構成をしているが，Arbiter 回路が比較する遅延時間差の伝搬経路が *n*-XOR PUF とは異なる．RG-DTM PUF は，Arbiter PUF と同じ構成をした PUF であるが，Arbiter 回路の閾値を変更することができる．Arbiter PUF に対しては，機械学習を用いた攻撃手法が知られている．Arbiter PUF で生じる遅延時間差は，数学的にチャレンジから線形に表すことができるため，出力値を機械学習で推定することで正規 PUF と同じ出力をするクローンを作製可能である．また近年では，機械学習の一種である深層学習によって，より複雑な非線形で表現される PUF をクローニング攻撃できることが知られている．

第 3 章では，PUF がもつ性質と攻撃シナリオを想定し，攻撃コストについて考察した．PUF には Confined PUF と Extensive PUF があり，想定するべき攻撃シナリオが異なる．Confined PUF はチャレンジ空間が小さいため，チャレンジとレスポンスを取得可能な環境では安全性を保つことができない．Extensive PUF はチャレンジ空間が大きいため，一部のチャレンジとレスポンスを取得可能でもそのまま利用するだけでは脅威とはならない．そこで Extensive PUF に対する攻撃シナリオを考察する．まず攻撃者がもつ能力による攻撃シナリオの違いについて検討を行った．攻撃者の能力として，内部に対して物理攻撃が可能な場合やレスポンスの取得回数に制限がある場合などがあり，攻撃の困難性は大きく異なる．また，PUF の出力を決定する遅延時間差に関するパラメータ数やチャレンジビットの数によっても攻撃シナリオは異なる．明白だが，パラメータ数が少ない方が攻撃は容易になり，パラメータ数が増えるほど攻撃

コストが上昇する．

第 4 章では，本論文で用いる安全性評価実験の環境について述べた．近年の深層学習分野の発展とともに，公開されたライブラリが充実しており，深層学習の実装は容易である．本論文では Pylearn2，Keras，Pytorch の 3 つのライブラリについて検討を行った結果，Keras を利用することにした．深層学習のパラメータ数には，隠れ層のノード数以外にも活性化関数がある．そこで，シグモイド関数，tanh 関数および ReLU 関数についても特徴をまとめた. 遅延時間差の分析には，シミュレーションを用いた．本章で取り扱うシミュレーションは，チップに実装された Arbiter PUF から計測した遅延時間差を基に作成した．シミュレーションによって作製した Arbiter PUF は任意の遅延時間差を観測できるなため，遅延時間差の詳細な分析を行った．今回，Arbiter 回路における物理的特性は考慮しておらず，大きな遅延は発生しないと仮定したが，レスポンスの 0/1 の頻度に影響をおよぼすことがわかった．遅延時間差を基にトレーニングデータを作成し学習を行ったところ，一部の PUF では遅延時間差が 0 に近い CRP をトレーニングデータとして用いることで予測成功率が上がることが確認できた．

第 5 章では，時系列処理を行った PUF に対して安全性評価を行った．時系列処理を用いた PUF として RG-DTM PUF と Q-class 認証を対象とした．RG-DTM PUF は，遅延時間差の閾値を細かく変更することによって，レスポンスの複雑さを増した PUF である．RG-DTM PUF は，機械学習を用いたクローニング攻撃に対しても耐性を有することが報告されていた．しかし，推定するパラメータが少ない場合，深層学習を用いたクローニング攻撃により容易にクローンが作製可能なことがわかった．Q-class 認証は，PUF の構成自体は変更せずレスポンスを多値化する手法である．Q-class 認証は，深層学習攻撃に耐性があるとされていたが，

n-XOR PUF に対して，安全性評価の再試験をした結果，クローンの作製が可能であることが明らかになった．むしろ，*Q*-class 認証により深層学習の予測成功率が上がることがわかった．

第 6 章では，並列実装を行った *n*-XOR PUF に対して安全性評価を行った．実験では，1 つのチップに 4 個実装された Arbiter PUF を用い，XOR 回路はハードウェアに実装していない．つまり，*n*-XOR PUF の実装として，Arbiter PUF の出力をソフトウェア上で XOR をとったモノを用いた．*n*-XOR PUF の安全性評価では 14-XOR PUF までは予測成功率が *n* の数に合わせて線形に低下し，*n* がそれ以上であれば 50% まで低下した．すなわち，並列実装を行った場合，並列実装された PUF の数が増えるほど予測成功率が低くなることがわかった．また，レスポンスに対して多数決処理，つまり時系列処理を用いた *n*-XOR PUF のデータでは，未加工のデータと比べて予測成功率が高い結果が得られた．この結果は，第 5 章の結果と同様に，時系列処理を行うと予測成功率が高くなることを示している．

第 7 章では，レスポンスに意図的なエラーを注入した認証方式の提案を行い，安全性評価を行った．一般的にエラーは機械学習や深層学習の妨げになると考えられる．実験の結果として，意図的なエラーを約 25% 注入しても *n* の数が小さな 2-XOR，3–XOR PUF では予測成功率は低下しなかったが，4-XOR PUF，5-XOR PUF では意図的なエラーの割合が高くなると予測成功率が低下した．6-XOR PUF ではエラー注入前の予測成功率がもともと 50% 前後だったため，変化は見られなかった．したがって，クローニング攻撃の予測成功率を低下させる上で，エラーを注入する手法は有効性であることがわかった．

以上の結果から，ハードウェアインスタンスでの処理が，深層学習による安全性評価に大き

く影響を与えることがわかった．時系列処理では 1 つの PUF に対して時系列処理を行うため，処理が増えるほどレスポンスに 1 つの PUF の挙動が表現されるため予測が容易となる．そのため，Q-class では予測成功率が上がってしまう結果となった．並列実装では並列実装の数 (n) が増えるほど予測成功率が低下した．これはレスポンスに複数の PUF ハードウェアインスタンスの挙動が表現されるためと考える．ただし，並列実装の場合，予測成功率の低下と実装コストのトレードオフが存在する．レスポンスに意図的なエラーを注入する方式では，意図的なエラーの注入により予測成功率が低下した．これは，意図的なエラーによってレスポンスが複雑になったためである．本論文で安全性評価を行った 3 方式では意図的なエラーの注入が最も有効な手段である．理由としては，意図的なエラーの注入は，実装コストを並列実装程上げずに実装が可能であり，クローニング攻撃の耐性向上が容易にできるためである．

8.2 今後の展望

今後の展望としては以下の点に注目したい．

■遅延時間差がクローニング攻撃に与える影響 本論文では，チップ上の PUF とシミュレーション PUF の二つでは安全性評価に 2 点の違いがみられた．シミュレーション PUF では 6-XOR PUF の段階でクローンの精度が大きく下がったがチップ上の PUF では 15-XOR PUF でもクローンの精度は高かった．また，チップ上の PUF では n の数が偶数か奇数かによって安全性評価に影響がある傾向がみられた．シミュレーション PUF は実測値を用いて理想的な PUF をシミュレートしている．しかし実際には，チップ上の PUF は一部に大きな遅延を生じる実装がされていたり，他の回路からの干渉を受けたりする可能性がある．事実，n の数が偶数か奇数かによって偏りが異なることを考えると，Arbiter 回路に特有の遅延が発生している

ことが考えられる．より正確な安全性評価を行うためには，チップ上に実装された PUF に近いシミュレーションを実装する必要がある．本論文のシミュレーション PUF では Arbiter 回路の遅延については検討を行っていないため，Arbiter 回路が安全性評価に与える影響については今後の課題である．

■環境ノイズの与える影響　本論文では，シミュレーション PUF に環境ノイズを付与していない．しかし，第 6 章の結果から環境ノイズはわずかではあるが，クローンの精度を下げることがわかっている．また，シミュレーション PUF に対して Q-class 認証では，2-class より 3-class，4-class の方が予測成功率が高くなることがわかっている．先行研究で RG-DTM PUF は 32 段 RG-DTM PUF に対して性能評価を行っているが，段数を増やした時については明らかにされていない．Arbiter PUF は段数が増えれば Arbiter 回路で測定する遅延時間差のばらつきが大きくなることがわかっている．RG-DTM PUF においてばらつきが大きくなれば，性能評価に大きく影響を与えると想定される．クローニング攻撃による安全性評価の結果だけではなく，性能評価の結果も考慮した安全性評価は PUF を運用する上で必要不可欠である．環境ノイズが安全性評価に与える影響については今後の課題である．

■より効果的なエラー注入の方法　本論文では，意図的なエラーはランダムに注入されている．しかし，ランダムなエラーの最適性は検証できていない．時系列処理を行った PUF や環境ノイズの結果の考察から，意図的なエラーの注入箇所に関しても検討の余地がある．そのため，遅延時間差やチャレンジのハミングウェイトなどによって意図的なエラーの注入方法を変化させるなどの検証が必要と考える．

参考文献

[1] GitHub - jgamblin/Mirai-Source-Code. https://github.com/jgamblin/Mirai-Source-Code.

[2] Hacking a Capsule Hotel - Ghost in the Bedrooms. https://www.blackhat.com/us-21/briefings/schedule/#hacking-a-capsule-hotel---ghost-in-the-bedrooms-23093.

[3] ISO/IEC 20897-1:2020 Information Security, Cybersecurity and Privacy Protection — Physically Unclonable Functions. https://www.iso.org/standard/76353.html.

[4] Top 200 most common passwords of the year 2020. https://nordpass.com/most-common-passwords-list/.

[5] Manos Antonakakis, Tim April, Michael Bailey, Matt Bernhard, Elie Bursztein, Jaime Cochran, Zakir Durumeric, J Alex Halderman, Luca Invernizzi, Michalis Kallitsis, et al. Understanding the Mirai Botnet. In *Proceedings of 26th USENIX security symposium (USENIX Security 17)*, pp. 1093–1110, 2017.

[6] Hiromitsu Awano, Tomoki Iizuka, and Makoto Ikeda. PUFNet: A Deep Neural Network Based Modeling Attack for Physically Unclonable Function. In *Proceedings of the IEEE*

International Symposium on Circuits and Systems (ISCAS), pp. 1–4, 2019.

[7] Georg T Becker. The Gap Between Promise and Reality: On the Insecurity of XOR Arbiter PUFs. In *Proceedings of International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, pp. 535–555, 2015.

[8] Nicolas Bruneau, Jean-Luc Danger, Adrien Facon, Sylvain Guilley, Soshi Hamaguchi, Yohei Hori, Yousung Kang, and Alexander Schaub. Development of the Unified Security Requirements of PUFs During the Standardization Process. In *Proceedings of International Conference on Security for Information Technology and Communications (SecITC)*, pp. 314–330. Springer, 2018.

[9] Mario Cardullo and William Parks. Transponder Apparatus and System, 1973. US Patent 3,713,148.

[10] Francois Chollet, et al. Keras. GitHub, 2015. `https://github.com/fchollet/keras`.

[11] Tyler Cultice and Himanshu Thapliyal. PUF-Based Post-Quantum CAN-FD Framework for Vehicular Security. *Information*, Vol. 13, No. 8, p. 382, 2022.

[12] Jean-Luc Danger, Risa Yashiro, Tarik Graba, Yves Mathieu, Abdelmalek Si-Merabet, Kazuo Sakiyama, Noriyuki Miura, and Makoto Nagata. Analysis of Mixed PUF-TRNG Circuit Based on SR-Latches in FD-SOI Technology. In *Proceedings of 2018 21st Euromicro Conference on Digital System Design (DSD)*, pp. 508–515. IEEE, 2018.

[13] Jeroen Delvaux and Ingrid Verbauwhede. Side Channel Modeling Attacks on 65nm Arbiter PUFs Exploiting CMOS Device Noise. In *Proceedings of International Symposium on Hardware-Oriented Security and Trust (HOST)*, pp. 137–142, 2013.

[14] Kota Fruhashi, Mitsuru Shiozaki, Akitaka Fukushima, Takahiko Murayama, and Takeshi

Fujino. The Arbiter-PUF with High Uniqueness Utilizing Novel Arbiter Circuit with Delay-Time Measurement. In *Proceedings of 2011 IEEE International Symposium of Circuits and Systems (ISCAS)*, pp. 2325–2328. IEEE, 2011.

[15] Dennis Gabor. A New Microscopic Principle, 1948.

[16] Blaise Gassend, Dwaine Clarke, Marten Van Dijk, and Srinivas Devadas. Silicon Physical Random Functions. In *Proceedings of 9th ACM Conference on Computer and Communications Security (ACM 2002)*, pp. 148–160, 2002.

[17] Ian J Goodfellow, David Warde-Farley, Pascal Lamblin, Vincent Dumoulin, Mehdi Mirza, Razvan Pascanu, James Bergstra, Frédéric Bastien, and Yoshua Bengio. Pylearn2: a Machine Learning Research Library. *arXiv preprint arXiv:1308.4214*, 2013.

[18] Jorge Guajardo, Sandeep S Kumar, Geert-Jan Schrijen, and Pim Tuyls. FPGA Intrinsic PUFs and Their Use for IP Protection. In *Proceedings of International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, pp. 63–80. Springer, 2007.

[19] Charles Herder, Ling Ren, Marten Van Dijk, Meng-Day Yu, and Srinivas Devadas. Trapdoor Computational Fuzzy Extractors and Stateless Cryptographically-Secure Physical Unclonable Functions. *IEEE Transactions on Dependable and Secure Computing*, pp. 65–82, 2016.

[20] Anthony Van Herrewege, Stefan Katzenbeisser, Roel Maes, Roel Peeters, Ahmad-Reza Sadeghi, Ingrid Verbauwhede, and Christian Wachsmann. Reverse Fuzzy Extractors: Enabling Lightweight Mutual Authentication for PUF-Enabled RFIDs. In *Proceedings of International Conference on Financial Cryptography and Data Security (FC'12)*, pp. 374–389. Springer, 2012.

[21] Yohei Hori, Takahiro Yoshida, Toshihiro Katashita, and Akashi Satoh. Quantitative and

Statistical Performance Evaluation of Arbiter Physical Unclonable Functions on FPGAs. In *Proceedings of 2010 International Conference on Reconfigurable Computing and FPGAs (ReConFig 2010)*, pp. 298–303, 2010.

[22] Chenglu Jin, Charles Herder, Ling Ren, Phuong Ha Nguyen, Benjamin Fuller, Srinivas Devadas, and Marten Van Dijk. FPGA Implementation of a Cryptographically-Secure PUF Based on Learning Parity with Noise. *Cryptography*, p. 23, 2017.

[23] Mahmoud Khalafalla and Catherine Gebotys. PUFs Deep Attacks: Enhanced Modeling Attacks Using Deep Learning Techniques to Break the Security of Double Arbiter PUFs. In *Proceedings of Design, Automation And Test in Europe (DATE 2019)*, pp. 204–209, 2019.

[24] Constantinos Kolias, Georgios Kambourakis, Angelos Stavrou, and Jeffrey Voas. DDoS in the IoT: Mirai and Other Botnets. *Computer*, Vol. 50, No. 7, pp. 80–84, 2017.

[25] BV Santhosh Krishna and T Gnanasekaran. A Systematic Study of Security Issues in Internet-of-Things (IoT). In *Proceedings of 2017 International Conference on IoT in Social, Mobile, Analytics and Cloud (I-SMAC)*, pp. 107–111. IEEE, 2017.

[26] Sandeep S Kumar, Jorge Guajardo, Roel Maes, Geert-Jan Schrijen, and Pim Tuyls. The Butterfly PUF Protecting IP on Every FPGA. In *Proceedings of 2008 IEEE International Workshop on Hardware-Oriented Security and Trust (HOST)*, pp. 67–70. IEEE, 2008.

[27] Jae W Lee, Daihyun Lim, Blaise Gassend, G Edward Suh, Marten Van Dijk, and Srinivas Devadas. A Technique to Build a Secret Key in Integrated Circuits for Identification and Authentication Applications. In *Proceedings of 2004 Symposium on VLSI Circuits. Digest of Technical Papers (IEEE Cat. No. 04CH37525)*, pp. 176–179. IEEE, 2004.

[28] Daihyun Lim. Extracting Secret Keys from Integrated Circuits. Master's thesis, Mas-

sachusetts Institute of Technology, 2004.

[29] Daihyun Lim, Jae W Lee, Blaise Gassend, G Edward Suh, Marten Van Dijk, and Srinivas Devadas. Extracting Secret Keys from Integrated Circuits. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 13, No. 10, pp. 1200–1205, 2005.

[30] Huichen Lin and Neil W Bergmann. IoT Privacy and Security Challenges for Smart Home Environments. *Information*, Vol. 7, No. 3, p. 44, 2016.

[31] Zhen Ling, Yiling Xu, Yier Jin, Cliff Zou, and Xinwen Fu. New Variants of Mirai and Analysis. *Encyclopedia of Wireless Networks, Springer, First Online*, Vol. 10, pp. 1–8, 2020.

[32] Takanori Machida, Dai Yamamoto, Mitsugu Iwamoto, and Kazuo Sakiyama. A New Mode of Operation for Arbiter PUF to Improve Uniqueness on FPGA. In *Proceedings of 2014 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pp. 871–878. IEEE, 2014.

[33] Takanori Machida, Dai Yamamoto, Mitsugu Iwamoto, and Kazuo Sakiyama. A New Arbiter PUF for Enhancing Unpredictability on FPGA. *The Scientific World Journal*, Vol. 2015, , 2015.

[34] Takanori Machida, Dai Yamamoto, Mitsugu Iwamoto, and Kazuo Sakiyama. Implementation of Double Arbiter PUF and Its Performance Evaluation on FPGA. In *Proceedings of The 20th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 6–7. IEEE, 2015.

[35] Roel Maes, Anthony Van Herrewege, and Ingrid Verbauwhede. PUFKY: A Fully Functional PUF-Based Cryptographic Key Generator. In *Proceedings of International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, pp. 302–319. Springer, 2012.

[36] Roel Maes, Pim Tuyls, and Ingrid Verbauwhede. Low-Overhead Implementation of a Soft Decision Helper Data Algorithm for SRAM PUFs. In *Proceedings of International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, pp. 332–347, 2009.

[37] Abhranil Maiti, Vikash Gunreddy, and Patrick Schaumont. A Systematic Method to Evaluate and Compare the Performance of Physical Unclonable Functions. In *Proceedings of Embedded systems design with FPGAs*, pp. 245–267. Springer, 2013.

[38] Hiroyuki Matsumoto, Itsuo Takeuchi, Hidekazu Hoshino, Tsugutaka Sugahara, and Tsutomu Matsumoto. An Artifact-Metric System Which Utilizes Inherent Texture. *IPSJ Journal*, Vol. 42, No. 7, pp. 1–14, 2001.

[39] Warren S McCulloch and Walter Pitts. A Logical Calculus of the Ideas Immanent in Nervous Activity. *The bulletin of mathematical biophysics*, Vol. 5, No. 4, pp. 115–133, 1943.

[40] Marvin Minsky and Seymour Papert. *Perceptrons: An Introduction to Computational Geometry*. MIT Press, Cambridge, MA, USA, 1969.

[41] Vinod Nair and Geoffrey E Hinton. Rectified Linear Units Improve Restricted Boltzmann Machines. In *Icml*, pp. 807–814, 2010.

[42] Vincent Omollo Nyangaresi and Nenad Petrovic. Efficient PUF Based Authentication Protocol for Internet of Drones. In *Proceedings of 2021 International Telecommunications Conference (ITC-Egypt)*, pp. 1–4. IEEE, 2021.

[43] OECD and European Union Intellectual Property Office. *Trends in Trade in Counterfeit and Pirated Goods*. 2019. `https://www.oecd-ilibrary.org/content/publication/g2g9f533-en`.

[44] Pierluigi Paganini. Hacking Drones ⋯ Overview of the Main

Threats. https://resources.infosecinstitute.com/topic/hacking-drones-overview-of-the-main-threats/.

[45] Vishal Pal, B Srikrishna Acharya, Somesh Shrivastav, Sourav Saha, Ashish Joglekar, and Bharadwaj Amrutur. PUF Based Secure Framework for Hardware and Software Security of Drones. In *Proceedings of 2020 Asian Hardware Oriented Security and Trust Symposium (AsianHOST)*, pp. 1–6. IEEE, 2020.

[46] Ravikanth Pappu. *Physical One-way Functions*. PhD thesis, Massachusetts Institute of Technology, 2001.

[47] Ravikanth Pappu, Ben Recht, Jason Taylor, and Neil Gershenfeld. Physical One-Way Functions. *Science*, Vol. 297, No. 5589, pp. 2026–2030, 2002.

[48] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019.

[49] Frank Rosenblatt. The Perceptron: a Probabilistic Model for Information Storage and Organization in the Brain. *Psychological review*, Vol. 65, No. 6, p. 386, 1958.

[50] Ulrich Rührmair, Frank Sehnke, Jan Sölter, Gideon Dror, Srinivas Devadas, and Jürgen Schmidhuber. Modeling Attacks on Physical Unclonable Functions. In *Proceedings of 17th ACM Conference on Computer and Communications Security (CCS)*, pp. 237–249, 2010.

[51] Ulrich Ruhrmair and Jan Solter. PUF Modeling Attacks: An Introduction and Overview. In *Proceedings of 2014 Design, Automation Test in Europe Conference Exhibition (DATE)*, pp. 1–6, 2014.

[52] Ulrich Rührmair, Jan Sölter, Frank Sehnke, Xiaolin Xu, Ahmed Mahmoud, Vera Stoyanova, Gideon Dror, Jürgen Schmidhuber, Wayne Burleson, and Srinivas Devadas. PUF Modeling Attacks on Simulated and Silicon Data. *IEEE transactions on information forensics and security*, Vol. 8, No. 11, pp. 1876–1891, 2013.

[53] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning Representations by Back-Propagating Errors. *nature*, Vol. 323, No. 6088, pp. 533–536, 1986.

[54] Durga Prasad Sahoo, Phuong Ha Nguyen, Chenglu Jin, and Kaleel Mahmood. DA_PUF_Library. `https:\\github.com\scluconn\DA_PUF_Library`.

[55] Pranesh Santikellur, Aritra Bhattacharyay, and Rajat Subhra Chakraborty. Deep Learning Based Model Building Attacks on Arbiter PUF Compositions. Cryptology ePrint Archive, Report 2019/566, 2019. `https:\\eprint.iacr.org\2019\566`.

[56] Mitsuru Shiozaki, Kousuke Ogawa, Kota Furuhashi, Takahiko Murayama, Masaya Yoshikawa, and Takeshi Fujino. Security Evaluation of RG-DTM PUF Using Machine Learning Attacks. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, Vol. 97, No. 1, pp. 275–283, 2014.

[57] Vijay Sivaraman, Hassan Habibi Gharakheili, Arun Vishwanath, Roksana Boreli, and Olivier Mehani. Network-level Security and Privacy Control for Smart-home IoT Devices. In *Proceedings of 2015 IEEE 11th International conference on wireless and mobile computing, networking and communications (WiMob)*, pp. 163–167. IEEE, 2015.

[58] Ying Su, Jeremy Holleman, and Brian P Otis. A Digital 1.6 pJ/bit Chip Identification Circuit Using Process Variations. *IEEE Journal of Solid-State Circuits*, Vol. 43, No. 1, pp. 69–77, 2008.

[59] G Edward Suh and Srinivas Devadas. Physical Unclonable Functions for Device Authentication and Secret Key Generation. In *Proceedings of 44th annual Design Automation Conference (DAC)*, pp. 9–14, 2007.

[60] Mark Mohammad Tehranipoor, Ujjwal Guin, and Domenic Forte. Counterfeit Integrated Circuits. In *Counterfeit Integrated Circuits*, pp. 15–36. Springer, 2015.

[61] Arijit Ukil, Soma Bandyopadhyay, and Arpan Pal. IoT-privacy: To Be Private or Not To Be Private. In *Proceedings of 2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 123–124. IEEE, 2014.

[62] Armed Service U.S.C. Inquiry into Counterfeit Electronic Parts in the Department of Defense Supply Chain., 2012.

[63] Yuejiang Wen. Improving Security and Reliability of Physical Unclonable Functions Using Machine Learning. Master's thesis, Clemson University, 2018.

[64] Nils Wisiol and Niklas Pirnay. Short Paper: XOR Arbiter PUFs Have Systematic Response Bias. In *Proceedings of International Conference on Financial Cryptography and Data Security (FC'20)*, pp. 50–57. Springer, 2020.

[65] Dai Yamamoto, Kazuo Sakiyama, Mitsugu Iwamoto, Kazuo Ohta, Masahiko Takenaka, and Kouichi Itoh. Variety Enhancement of PUF Responses Using the Locations of Random Outputting RS Latches. *Journal of Cryptographic Engineering*, Vol. 3, No. 4, pp. 197–211, 2013.

[66] Risa Yashiro, Yohei Hori, Toshihiro Katashita, and Kazuo Sakiyama. Deep Learning Attack against Large n-XOR PUFs on 180nm Silicon Chips. In *Proceedings of 2020 International Workshop on Nonlinear Circuits, Communications and Signal Processing (NCSP)*. 2020 International Workshop on Nonlinear Circuits, Communications and Signal Processing, 2020.

[67] Risa Yashiro, Takanori Machida, Mitsugu Iwamoto, and Kazuo Sakiyama. Deep-Learning-Based Security Evaluation on Authentication Systems Using Arbiter PUF and Its Variants. In *Proceedings of International Workshop on Security (IWSEC 2016)*, pp. 267–285, 2016.

[68] Risa Yashiro, Takeshi Sugawara, Mitsugu Iwamoto, and Kazuo Sakiyama. Q-Class Authentication System for Double Arbiter PUF. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, Vol. 101, No. 1, pp. 129–137, 2018.

[69] Xu Zhuang, Yan Zhu, Chin-Chen Chang, and Qiang Peng. Security Issues in Ultralightweight RFID Authentication Protocols. *Wireless Personal Communications*, No. 1, pp. 779–814, 2018.

[70] 宮本雅子. 住宅照明の現状と将来展望. 日本家政学会誌, Vol. 71, No. 5, pp. 324–330, 2020.

[71] 経済産業省. 模倣品対策技術及びその普及に向けた調査. 2018. https://www.meti.go.jp/policy/ipr/reports/pdf/kanbetugijutu30fy.pdf.

[72] 総務省. 情報通信白書: 進化するデジタル経済とその先にある Society 5.0. 情報通信白書 : ICT 白書. 日経印刷, 2019.

[73] 嶋田努, 長田幸恵, 原祐輔, 登真良, 旭野欣也, 牧野智成, 崔吉道. 個別認証技術を用いた医療用麻薬フェンタニル注射液の院内個別管理. 医療薬学, Vol. 46, No. 5, pp. 249–256, 2020.

[74] 飯塚知希, 粟野皓光, 池田誠. 深層ニューラルネットワークを用いた Double-Arbiter PUF に対するモデリング攻撃. 電子情報通信学会技術研究報告, Vol. 117, No. 455, pp. 231–236,

2018.

[75] 牧野智成, 旭野欣也, 登真良, 神崎智至. 人工物メトリクスによる偽造防止技術の紹介: 印刷の微細な違いを利用する個別認証技術 (SAMP)(特集 医療現場の安全・安心). 自動認識, Vol. 32, No. 8, pp. 20–26, 2019.

投稿論文の再利用に関して

学術雑誌

1. 八代理紗, 堀洋平, 片下敏宏, 崎山一男. 意図的なエラーを付与することによる深層学習を用いた Arbiter PUF へのクローニング攻撃の対策

   印刷公表の方法および時期： 情報処理学会論文誌, Vol.61, No.12, pp.1871-1880, 2020.

   (6 章に関連)

国際会議 (査読あり)

1. Risa Yashiro, Yohei Hori, Toshihiro Katashita, and Kazuo Sakiyama. A Deep Learning Attack Countermeasure with Intentional Noise for a PUF-based Authentication Scheme

   印刷公表の方法および時期：International Conference on Security for Information Technology and Communications (SecITC'19) , LNCS 12001, Springer-Verlag, pp.78-94, 2019.

   (6 章に関連)

2. Risa Yashiro, Yohei Hori, Toshihiro Katashita, and Kazuo Sakiyama. Deep Learning At-

tack against Large n-XOR PUFs on 180nm Silicon Chips

印刷公表の方法および時期：RISP International Workshop on Nonlinear Circuits, Communications and Signal Processing (NCSP'20), pp.598-601, 2020.

(5 章に関連)

口頭発表

1. 八代理紗, 堀洋平, 片下敏宏, 汐崎充, 崎山一男. RG-DTM PUF に対する Deep Learning を用いたクローニング攻撃

   印刷公表の方法および時期：2020 年暗号と情報セキュリティシンポジウム (SCIS'20), 3E1-1, 6 pages, 2020.

   (7 章に関連)

謝辞

本研究を遂行するにあたり，主任指導教官として終始ご助言をくださり，また丁寧にご指導いただいた崎山一男教授，副指導教官としてご指導ご教示をいただいた岩本貢教授，菅原健准教授に心より感謝の意を表します．また，リサーチアシスタントとして研究の場を与えてくださり，ご指導もいただいた産業技術総合研究所の堀洋平主任研究員，片下敏宏主任研究員に深く感謝申し上げます．そして，審査委員として，御指導いただいた大坐畠智准教授，李陽准教授に深く感謝いたします．

最後に，社会人として働きながらの学位取得にご理解いただいたセコム株式会社 IS 研究所の皆様，博士進学から学位取得まで暖かく見守ってくれた家族に感謝いたします．

八代　理紗

2023 年　3 月

発表論文目録

学術雑誌

1. Risa Yashiro, Takeshi Sugawara, Mitsugu Iwamoto, and Kazuo Sakiyama. Q-class Authentication System for Double Arbiter PUF. IEICE Trans. Fundam. Electron. Commun. Comput. Sci., Vol.E101-A, No.1, pp.129-137, 2018.
2. 八代理紗, 堀洋平, 片下敏宏, 崎山一男. 意図的なエラーを付与することによる深層学習を用いた Arbiter PUF へのクローニング攻撃の対策. 情報処理学会論文誌, Vol.61, No.12, pp.1871-1880, 2020.

国際会議 (査読あり)

1. Risa Yashiro, Takanori Machida, Mitsugu Iwamoto, and Kazuo Sakiyama. Deep-Learning-Based Security Evaluation on Authentication Systems Using Arbiter PUF and Its Variants. In Proceedings of International Workshop on Security 2016 (IWSEC'16), LNCS 9836, Springer-Verlag, pp.267-285, 2016.
2. Jean-Luc Danger, Risa Yashiro, Tarik Graba, Sylvain Guilley, Yves Mathieu, Noriyuki

Miura, Abdelmalek Si-Merabet, Kazuo Sakiyama, and Makoto Nagata. Analysis of Mixed PUF-TRNG Circuit Based on SR-Latches in FD-SOI Technology. In Proceedings of Euromicro Conference on Digital System Design (DSD'18), IEEE, pp.508-515, 2018.

3. Risa Yashiro, Yohei Hori, Toshihiro Katashita, and Kazuo Sakiyama. A Deep Learning Attack Countermeasure with Intentional Noise for a PUF-based Authentication Scheme. In Proceedings of International Conference on Security for Information Technology and Communications (SecITC'19) , LNCS 12001, Springer-Verlag, pp.78-94, 2019.
4. Risa Yashiro, Yohei Hori, Toshihiro Katashita, and Kazuo Sakiyama. Deep Learning Attack against Large n-XOR PUFs on 180nm Silicon Chips. In Proceedings of RISP International Workshop on Nonlinear Circuits, Communications and Signal Processing (NCSP'20), pp.598-601, 2020.

口頭発表

1. 八代理紗, 町田卓謙, 岩本 貢, 崎山一男. Deep Learning を用いた Double Arbiter PUF の安全性評価. IEICE2016 年総合大会, 2016.
2. 八代理紗, 藤井達哉, 岩本貢, 崎山一男. Deep Learning を用いた RSA に対する単純電磁波解析," IEICE2016 年ソサイエティ大会, 2016.
3. Risa Yashiro, Mitsugu Iwamoto, and Kazuo Sakiyama. Q-Class Authentication System Using DAPUF. Poster Session, AsianHOST'16, 2016.
4. 八代理紗, 菅原健, 崎山一男. Arbiter PUF に対する攻撃手法に関する一考察. 情報処理学会 DA シンポジウム 2018 (特別セッション), 6 pages, 2018.

5. 八代理紗, 藤聡子, 菅原健, 崎山一男. Arbiter PUF へのサイドチャネルモデリング攻撃の実装と応用. IEICE2018 年ソサイエティ大会, 2018.
6. Risa Yashiro, Takeshi Sugawara, Mitsuru Shiozaki, Takeshi Fujino, and Kazuo Sakiyama. A TEG Chip of Arbiter PUF for Efficient Simulation Model. In Conference Record of International Conference on Computer and Communication Systems (ICCCS'19), 2019
7. 八代理紗, 堀洋平, 片下敏宏, 汐崎充, 崎山一男. RG-DTM PUF に対する Deep Learning を用いたクローニング攻撃. 2020 年暗号と情報セキュリティシンポジウム (SCIS'20), 3E1-1, 6 pages, 2020.

その他

- 発表
  1. 八代理紗, 町田卓謙, 岩本 貢, 崎山一男. Double Arbiter PUF に対する Deep Learning を使った安全性評価. Hot Channel Workshop 2016, 2016.
  2. 八代理紗, 菅原健, 岩本貢, 崎山一男. PUF への機械学習攻撃と耐性強化に向けて. PUF 技術シンポジウム 2018, 2018.
  3. Jean-Luc Danger, Risa Yashiro, Tarik Graba, Sylvain Guilley, Yves Mathieu, Noriyuki Miura, Abdelmalek Si-Merabet, Kazuo Sakiyama, and Makoto Nagata. Analysis of Mixed PUF-TRNG Circuit Based on SR-Latches in FD-SOI Technology(from DSD 2018). 情報セキュリティ研究会 (ISEC), 2019.
- 表彰
  1. 2016 年度 電気通信大学　学生表彰

2. SEC 道後 2017 学生研究賞

3. ISEC 研究会 活動貢献感謝状 2019 年 5 月研究会