# Arbiter PUFに対する攻撃手法に関する一考察

八代　理紗[1,a)]　菅原　健[1,b)]　崎山　一男[1,c)]

**概要**：ICチップの真正性の確認に有効とされる技術にPhysical Unclonable Function (PUF) がある．PUFは，製造時に意図せずに生まれる物理的特徴を利用し，個体ごとに異なるID情報をデバイスに付与する技術である．PUFの1種にArbiter PUFが存在する．Arbiter PUFは配線遅延やゲート遅延を基に固有のID情報を出力するPUFである．Arbiter PUFに対する攻撃手法に，機械学習を用いて挙動を模倣したクローンを作製する手法が一般的に知られている．最新の研究では，Deep Learningを用いることにより，攻撃性能が向上したという報告もある．本論文では，各攻撃手法の特徴について議論する．

**キーワード**：PUF, 攻撃, 機械学習, Deep Learning

# A Study of Attack Method for Arbiter PUF

Risa Yashiro[1,a)]　Takeshi Sugawara[1,b)]　Kazuo Sakiyama[1,c)]

***Abstract:*** Physical Unclonable Function (PUF) is a technology that enables to identify the IC chip. PUF uses physical characteristic that is generated in the manufacturing phase and embed different IDs to the devices for each individual. There is an Arbiter PUF that is a kind of PUF. Arbiter PUF generates the unique ID that is based on the gate delay and wiring delay. In general, there is known generates the clone that imitates legitimate one's behavior using machine learning as the attack method for Arbiter PUF. It has been reported the attack result is improved using deep learning in the latest study. In this paper, we discuss the feature of each attack method for Arbiter PUF.

## 1. はじめに

近年，様々なデバイスが通信機能を持つInternet of Things (IoT) の登場により，日常生活の質の向上が期待されている．それに伴い，機密情報保護のためにデバイスへのセキュリティ技術の搭載が強く求められている．一方，現実にはIntegrated Circuit (IC) チップの偽造品が，多く流通しており問題となっている．例えば，偽造されたICチップに，ハードウェア版トロイの木馬型ウイルス (ハードウェアトロジャン) が潜伏していた場合，ユーザが認識することなく情報流出を起こしている可能性もある．このような問題を未然に防ぐためには，信頼のできるメーカの出荷している正規品のICチップか否かをユーザが確認できる必要がある．

ICチップの真正性を確認する技術としてPhysical Unclonable Function (PUF) [1], [2] が存在する．PUFの利用により，ICチップの真正性を保証でき，結果として，IoTのセキュリティを向上させることが可能になる．PUFは入力 (チャレンジ) が与えられた時に，製造時に意図せずに生まれた物理的特徴を用いて，個体ごとのID (レスポンス) を作製可能とする技術である．

一般的に，PUFは大きく分けてWeak PUFとのStrong PUFの2種類に分類できる [5]．Weak PUFはチャレンジ空間が狭く，主に鍵生成に用いられる．ここで挙げたチャレンジ空間は，与えることが可能なチャレンジの総数を指す．代表例であるラッチPUFは，電源を入れた時のラッチの出力値を利用したPUFであり，チャレンジは電源のON/OFFで，レスポンスはラッチの出力値である．一方，

[1] 電気通信大学
The University of Electro-Communications, Chofu, Tokyo 182–8585, Japan
[a)] yashiro@uec.ac.jp
[b)] sugawara@uec.ac.jp
[c)] sakiyama@uec.ac.jp

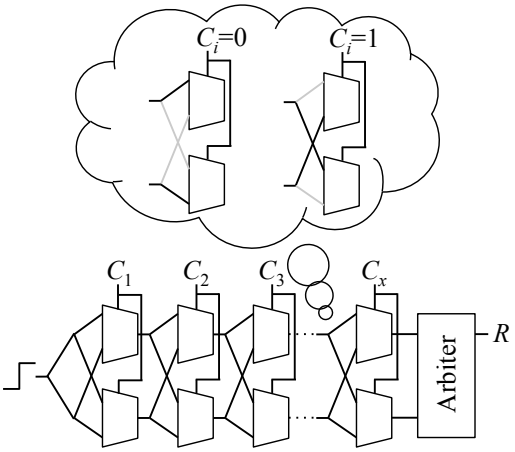


**図 1** Arbiter PUF の構成

Strong PUF はチャレンジ空間が広く，認証に用いられる．Arbiter PUF [3] は，その先駆けとして有名である．

Arbiter PUF は，2 経路を通る信号の伝搬遅延の違いによって固有の ID を出力する．Arbiter PUF のチャレンジは $x$ ビットのバイナリ列であり，レスポンスは 2 経路の伝搬遅延の競争結果であり，1 ビットで表す．Arbiter PUF は，機械学習を用いることにより，攻撃可能と報告されている [4], [5]．Arbiter PUF に対する攻撃は，未知のチャレンジに対するレスポンスの予測成功率を用いて，性能評価を行う．Arbiter PUF とそのバリアントに関しては，2.1 章にて，攻撃手法に関しては，2.2 章にてそれぞれ詳細に説明する．

## 2. 関連研究

### 2.1 Arbiter PUF とバリアント

Arbiter PUF の構成を図 1 に示す．Arbiter PUF は，2 個のセレクタ (段) と 1 つの Arbiter (例 Set-Reset ラッチ) をビルディングブロックとして構成される．各段の 2 個のセレクタには，前の段から得られる出力をそれぞれのセレクタに入力するように設計されている．$x$ 段の Arbiter PUF に対して，チャレンジと呼ぶ $x$ ビットのバイナリ列で経路を決定する．$i$ ビット目のチャレンジ $C_i$ が 0 の場合，$i$ 番目の経路は直進し，1 の場合は交差する．チャレンジによって決定された 2 経路を伝搬した信号は，最終的に Arbiter によってどちらが速かったかを競争する．その結果，0 または 1 の出力を得ることができる．これをレスポンスと呼ぶ．

#### 2.1.1 *n*-XOR Arbiter PUF

$n$-XOR Arbiter PUF は Arbiter PUF の機械学習攻撃耐性を向上させる手法として，2007 年に Suh らによって提案された [7]．$n$-XOR Arbiter PUF ($n$=3) の構成を図 2 に示す．$n$-XOR Arbiter PUF は Arbiter PUF を $n$ 個並列に並べ，各 Arbiter PUF から得られる $n$ ビットの出力を XOR させ，1 ビットのレスポンスを得る PUF である．$n$-XOR Arbiter PUF は，攻撃の難易度が $n$ の値に伴い，指数関数的に向上するとしていた．

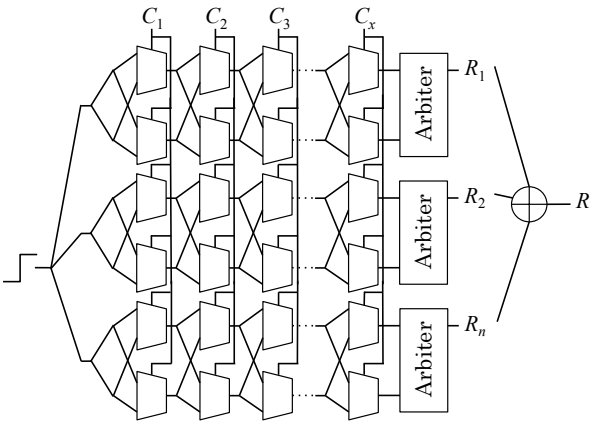


**図 2** 3-XOR Arbiter PUF の構成

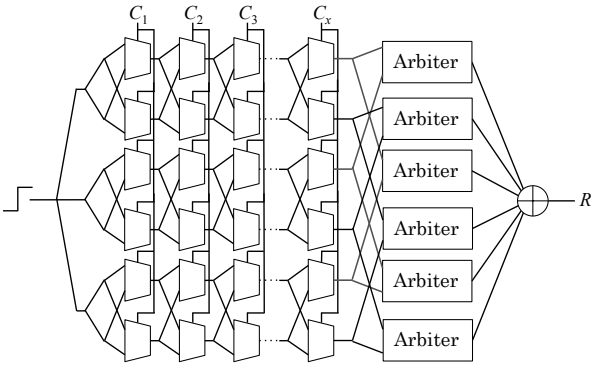


**図 3** 3-1 Double Arbiter PUF の構成

#### 2.1.2 Double Arbiter PUF

Field-Programmable Gate Array (FPGA) 上に実装された Arbiter PUF は，固有性 [13], [14] が低くなることが指摘されている．Double Arbiter PUF は町田らによって固有性を向上するために 2014 年に提案された [10], [12]．$n$-1 Double Arbiter PUF ($n$=3 の時) の構成を図 3 に表す．$n$-1 Double Arbiter PUF は，Arbiter PUF を $n$ 個並列に並べ，同じ経路を通った信号の伝搬遅延を競争させた PUF である．Double Arbiter PUF は，固有性の改善だけではなく，機械学習攻撃への耐性も向上したとしている．

### 2.2 Arbiter PUF に対する攻撃

Arbiter PUF に対する攻撃の目的は，正規品の PUF の挙動を模倣したクローンを作製することである．クローンは未知のチャレンジに対するレスポンスの推定を可能とする．攻撃シナリオの条件は，攻撃対象の PUF からチャレンジとそれに対応するレスポンス (チャレンジレスポンスペア) を取得可能であることである．攻撃者は，攻撃対象のチャレンジレスポンスペアを複数用いることによってクローンを作製する．クローンの性能評価は，未知のチャレンジに対するレスポンスを予測し，実際の PUF から得ら

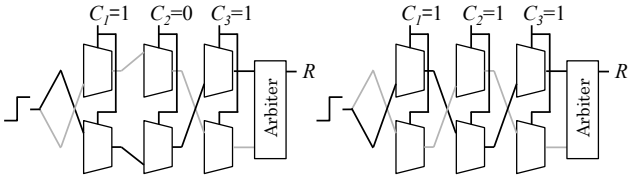


図 4 $\vec{C} = (1,0,1)$ ならびに $\vec{C} = (1,1,1)$ に対する経路

れるレスポンスと比較した際の正答率により評価される．

### 2.2.1 機械学習を用いたモデリング攻撃 [6]

Arbiter PUF に対する攻撃手法として，機械学習を用いたモデリング攻撃 [4], [5] [6] が知られている．モデリング攻撃は，モデリング式と実際の PUF から取得したチャレンジスポンスペアを用い，内部の遅延情報を推定することによりクローンを作製する．

複製に用いるモデリング式について説明を記述する．Arbiter に入力時に発生している遅延時間差 $\Delta$ は，チャレンジ $\vec{C} = (C_1, C_2, \ldots, C_x)$ を変形させた経路情報である $\vec{\Phi}$, 内部の遅延情報である $\vec{w}$ を用いて $\Delta = \vec{w}^T \vec{\Phi}$, としている．このとき，内部の遅延情報 $\vec{w}$ は，$\vec{w} = (w^1, w^2, \ldots, w^x, w^{x+1})^T$ で表し，$w^i$ は次式 (1) で表現する．

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

なお，$\delta^1$ の経路は交差，$\delta^0$ の経路は直進とし，$\delta_i^{0/1}$ は $i$ 段目で発生する遅延時間を表す．次にチャレンジを変形させた経路情報を以下の (2) 式から導出する．

$$\vec{\Phi} = (\Phi^1(\vec{C}), \Phi^2(\vec{C}), \ldots, \Phi^x(\vec{C}), 1)^T. \quad (2)$$

このとき，$\Phi^l(\vec{C})$ それぞれの値は $l = 1, \ldots, x$ の範囲となり，次式 (3) で表現する．

$$\Phi^l(\vec{C}) = \prod_{i=l}^{x} (1 - 2b_i). \quad (3)$$

例えば，$\vec{C} = (1,0,1)$ のとき，$\vec{\Phi}$ を上記の (3) 式によって求めると，$\vec{\Phi} = (1,-1,-1,1)^T$ となり，$\vec{C} = (1,1,1)$ のときは，$\vec{\Phi} = (-1,1,-1,1)^T$ となる．図 4 と $\vec{\Phi}$ の値を見比べてわかるように，$\vec{\Phi}$ の値によって，上下どちらの経路を通るかが表現可能となる．

最後に，遅延時間差 $\Delta$ をシグナム関数に与えることにより，レスポンス値の取得が可能である．

$$R = \mathrm{sgn}(\Delta) = \mathrm{sgn}(\vec{w}^T \vec{\Phi}). \quad (4)$$

シグナム関数は値が負の時 $-1$ を，値が正のとき 1 を返すため，Arbiter と同じ挙動をする．

実際の攻撃手法としては，チャレンジレスポンスペアから $\Delta$ と $\vec{\Phi}$ を計算し，機械学習をすることで，内部の遅延情報 $\vec{w}$ を導出する．得られた $\vec{w}$ と未知のチャレンジを用いてレスポンスの予測をし，攻撃を行う．

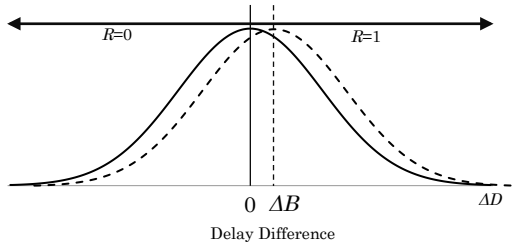


図 5 Arbiter PUF の遅延時間差

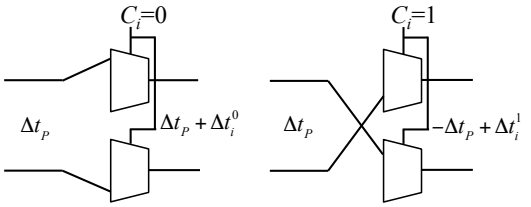


図 6 各段での遅延時間差

### 2.2.2 サイドチャネルモデリング攻撃 [9]

レスポンスは環境ノイズや測定ノイズの影響によって，異なる値になってしまうことがある．信頼性 [*1] という指標によって異なる値を出力する確率を表すことが可能である．レスポンスの信頼性をサイドチャネル情報とし，機械学習を用いずにクローンを作製する手法が提案されている [9].

Arbiter PUF 内部で発生する遅延時間差は，図 5 のように正規分布に従うと仮定できる．この時，遅延時間差が正ならばレスポンスが 1，負であればレスポンスが 0 になる．つまり，遅延時間差が 0 に近いチャレンジでは，レスポンスの値が測定ノイズの影響を受けやすく，信頼性が低くなることが推測できる．このことを利用すれば，信頼性の値によって遅延時間差の予測が可能になることを示唆している．

各段で生じる遅延時間差を図 6 のように定義する．経路が直進するときに追加される遅延時間差は $\Delta t_i^0$，交差するときは $\Delta t_i^1$ とする．各段の入力前に生じている遅延時間差 $\Delta t_P$ は，入力時点で下の経路で発生する遅延時間から上の経路で発生する遅延時間を引いた差分である．そのため，経路が交差する場合，$\Delta t_P$ の値の符号が反転し，該当の段で生じた遅延時間差を加算する．

Arbiter PUF 内で最終的に生じる遅延時間差 $\Delta t$ は次式により，導出可能であるとされている．

$$\Delta t = \vec{\Phi} \vec{\tau} = \vec{\Phi} (\tau_1, \tau_2, \ldots, \tau_{x+1})^T. \quad (5)$$

このとき，$\vec{\Phi}$ は 2.2.1 章で述べたものと同じであり，$\vec{\tau}$ は，

[*1] [13] では Steadiness, [14] では Reliability と定義されている．

$$\vec{\tau} = \frac{1}{2}\begin{pmatrix} & & \Delta t_1^0 - \Delta t_1^1 \\ \Delta t_1^0 + \Delta t_1^1 & + & \Delta t_2^0 - \Delta t_2^1 \\ & \vdots & \\ \Delta t_{x-1}^0 + \Delta t_{x-1}^1 & + & \Delta t_x^0 - \Delta t_x^1 \\ \Delta t_x^0 + \Delta t_x^1 & & \end{pmatrix}. \quad (6)$$

によって求められる．

実際の Arbiter PUF 内で最終的に生じる遅延時間差 $\Delta t$ は，測定ノイズ $\Delta N$ と Arbiter 部分で発生する偏り $\Delta B$ の影響を受ける．つまり，$\Delta B$ の影響により，図 5 のような遅延時間差が従う正規分布の平均値は $\Delta B$ 偏ることになる．これにより，レスポンスの値は 0 または 1 に偏りやすくなる．

これらのことから，実際の Arbiter PUF から出力されるレスポンスの信頼性 $S$ は次式で求められる．

$$S = \frac{1}{2}\mathrm{erfc}(\frac{\Delta B - \Delta t}{\sqrt{2}\sigma_N}). \quad (7)$$

信頼性を erfc を用いて表現したとき，10%〜90%の区間に該当する箇所は線形近似が可能であると主張されている．そこで，10%〜90%のチャレンジと信頼性の値を取得し，前述の (5) 式に代入することによって，内部情報である $\vec{\tau}$ を導出する．得られた内部情報 $\vec{\tau}$ はチャレンジを与えることによって，最終の遅延時間差が推定可能になるため，クローンとして利用可能であると提案されている．

#### 2.2.3 信頼性攻撃 [8]

$n$-XOR Arbiter PUF に対する攻撃手法としてレスポンスの信頼性を用いた信頼性攻撃が提案されている [8]．信頼性攻撃は前述のサイドチャネルモデリング攻撃を改良し，$n$-XOR Arbiter PUF に適応させた攻撃手法である．信頼性攻撃を用いることにより，$n$-XOR Arbiter PUF の攻撃困難性が指数的から線形的まで低下すると報告されている．この攻撃手法では共分散行列適応進化戦略 (CMA-ES) アルゴリズムを用いて攻撃を行う．CMA-ES は非決定的アルゴリズムなため，学習のたびに違うモデルを出力する．信頼性攻撃では $n$-XOR Arbiter PUF に対して，サイドチャネルモデリング攻撃を CMA-ES を用いて $n$ 回行う．つまり，攻撃結果として $n$ 個の異なる Arbiter PUF のモデルを得ることができる．この時，それぞれのモデルのパラメータは異なっていることが理想的であるとされている．得られた $n$ 個のモデルを用いて得られたレスポンスを XOR させることにより，$n$-XOR Arbiter PUF への攻撃耐性を指数的から線形的まで低下させることが可能であると報告されている．

#### 2.2.4 Deep Learning を用いた攻撃

機械学習を用いたモデリング攻撃の最新研究として，Deep Learning (DL) を用いた攻撃研究が報告されている [11], [15]. DL を用いることにより，攻撃性能の著しい向上がみられ，攻撃不可能としていた Double Arbiter PUF に対しても攻撃可能であることが報告された [15]．これまで Arbiter PUF の機械学習攻撃というと Support Vector Machine (SVM) や Logistic Regression (LR) を用いた手法が一般的であった．しかし，これらの機械学習方式は線形識別に向いているとされており，Arbiter PUF の変形種に対しては構造が複雑になる程攻撃が困難になっていた．DL では，非線形の識別も可能とするため，Double Arbiter PUF の攻撃も可能となったと飯塚らは述べている．

## 3. 各攻撃手法の特徴

これまで述べてきた攻撃手法と本論文の特徴を，表 1 にまとめた．また，各攻撃手法には次のような欠点が存在すると考える．まず，Rührmair et al. [6] の提案手法は，Arbiter PUF や構造が簡単なバリアントに対しては攻撃可能であるが，構造が複雑になると，攻撃が困難となる傾向がある．次に，Delvaux et al. [9] の提案手法は，より Arbiter PUF の内部情報に近いモデルを取得できる一方で，機械学習を用いる攻撃に比べ，破棄するチャレンジレスポンスペアが存在するため非効率である．そして，Becker [8] の提案手法は，$n$-XOR Arbiter PUF に対する既存の攻撃手法と比べて攻撃困難性を下げることが可能だが，$n$-XOR Arbiter PUF にしか対応していない点と，$n$-XOR Arbiter PUF の性質によっては攻撃精度が低下する可能性がある．最後に， 飯塚ら [15] の提案手法は，攻撃精度が著しく向上し，Double Arbiter PUF の攻撃を可能としたが，Deep Learning は作製したモデルの分析が現状ではできないため，Arbiter PUF のモデリング攻撃に対する脆弱性の要因特定への応用は難しいと考える．

これらの攻撃研究から，攻撃性能の向上と攻撃に用いるリソースの必要量はトレードオフな関係であると考えられる．攻撃に用いるリソースは，PUF から得られるチャレンジレスポンスペアやモデルを作製する計算機の性能，計算時間などが挙げられる．つまり，攻撃性能の向上に対して攻撃者がどこまでリソースを用いることが可能かが攻撃研究の主軸となる可能性が高い．

これまで PUF の攻撃研究では，どの程度レスポンスが予測できるかという攻撃結果が重視されており，攻撃リソースなどに関する検証が不十分である．その一例として，攻撃の要であるモデリングに関して，十分な解析が行われていないことが挙げられる．モデリングの解析を行うことによって，脆弱性要因の特定などが期待でき，攻撃リソースを充足する必要性を高めることができると考える．そこで本稿では，脆弱性要因の特定に向けてソフトウェア実装の PUF を用いたモデリング解析を行う．

ソフトウェア上に実装した PUF を用いる理由は，ハードウェア上に実装した PUF に比べて詳細なパラメータの抽出が可能なため，作製したモデルの精度を伝搬遅延でより詳細に解析可能になるためである．また，ソフトウェア

表 1　各論文の特徴

| 論文 | 攻撃可能 PUF | | | 目的 | モデルの | 機械学習 | モデルの |
|---|---|---|---|---|---|---|---|
| | APUF[1] | XOR[2] | DAPUF[3] | | パラメータ利用 | | 推定手法 |
| Rührmair et al. [6] | ○ | — | — | 攻撃モデルの提案 | × | SVM，LR | — |
| Delvaux et al. [9] | ○ | × | — | 攻撃手法の提案 | × | × | 最小二乗法 |
| Becker [8] | — | ○ | — | 攻撃手法の提案 | × | CMA-ES | — |
| 八代ら [11] | ○ | ○ | × | DL の攻撃利用 | × | DL | — |
| 飯塚ら [15] | — | — | ○ | DL の攻撃利用 | × | DL | — |
| 本稿 | ○ | — | — | モデリング解析 | ○ | × | 最小二乗法 |

[1]Arbiter PUF，[2]$n$-XOR Arbiter PUF，[3]Double Arbiter PUF

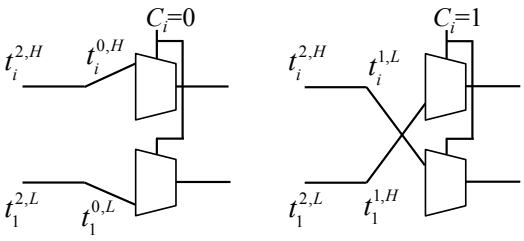


図 7　ソフトウェア実装 Arbiter PUF のパラメータ設定

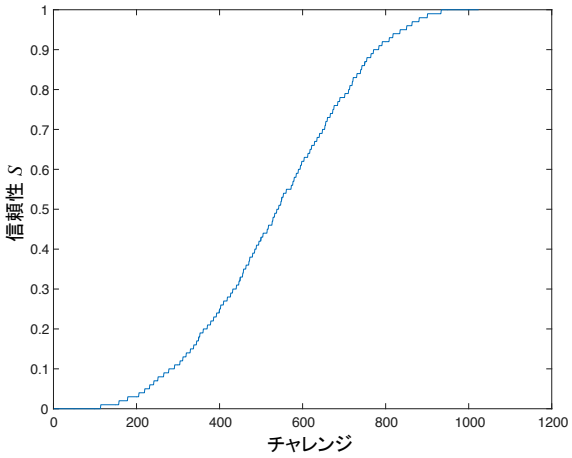


図 8　ソフトウェア実装 PUF の信頼性の累積度数

実装ではパラメータの変更がハードウェア実装に比べて容易のため，十分なスケーラビリティを獲得することが可能である．ソフトウェア実装の PUF に設定するパラメータを図 7 に示す．ソフトウェア PUF に設定するパラメータは 1 段に対し，6 つ用意し，$t_i^{0,H/L}$，$t_i^{1,H/L}$，$t_i^{2,H/L}$ とする．それぞれ対応するパラメータの $H-L$ を各段で発生する決定的な遅延時間差 $\Delta t$ として導出する．また，ソフトウェア実装の PUF では測定ノイズが載らないため，$\Delta t$ を平均とする正規分布に従う乱数を生成し，測定ノイズ入り遅延時間差 $\Delta t'$ とする．レスポンス 0/1 は，$\Delta t'$ を経路に合わせて負の符号をつけ総和をとり，総和の符号によって決定する．また，今回実装した PUF では $\Delta B = 0$ になるようにパラメータを決定した．

そして本稿の目的であるモデリング解析を行うには，内部情報が取得できるサイドチャネルモデリング攻撃が適切な手法だと考え，攻撃に向けた分析を行った．まず，図 8 に異なる 1024 チャレンジに対するレスポンスの信頼性の累積度数を示す．信頼性はレスポンスから導出ができるため，ハードウェア実装，ソフトウェア実装にかかわらず取得することが可能である．信頼性が 0 または 1 のときは，レスポンスが安定していることを表し，全体的に割合が高いことがわかる．

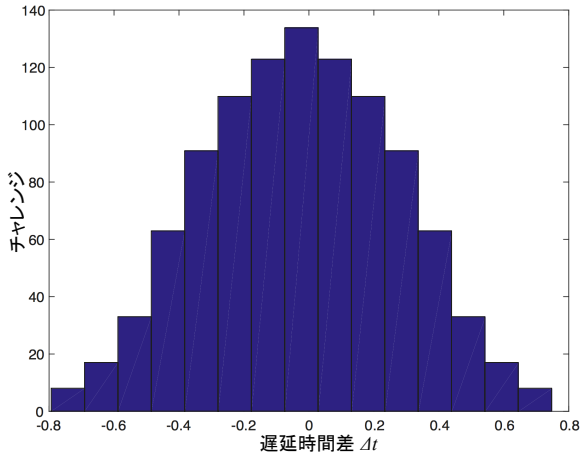


図 9　ソフトウェア実装 PUF の遅延時間差のヒストグラム

次にソフトウェア実装 PUF から得られた遅延時間差のヒストグラムを図 9 に示す．ハードウェア実装の PUF では遅延時間差を取得することができないが，ソフトウェア実装 PUF では設定した伝搬遅延のパラメータを抽出可能である．サイドチャネルモデリング攻撃は遅延時間差の分布が正規分布に従っているという仮定の元で攻撃を行う．この図から遅延時間差のばらつきが正規分布に従っており，今回ソフトウェア実装した PUF の挙動が想定通りであることがわかる．

最後に，図 10 にソフトウェア実装 PUF の遅延時間差の累積分布関数を示す．遅延時間差の累積分布関数と信頼性の累積度数の図を比較すると，形状が類似していることがわかる．そのため線型性を満たす信頼性が 0.1〜0.9 の間のチャレンジと信頼性を用いることで攻撃可能であることが示唆できる．

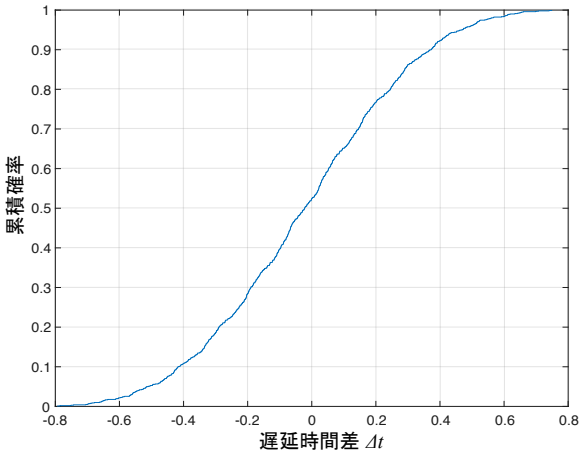


**図 10** ソフトウェア実装 PUF の遅延時間差の累積分布関数

## 4. まとめ

本論文では，これまで提案された Arbiter PUF に対する攻撃手法について調査し，それぞれの攻撃手法の特徴についてまとめた．また，攻撃手法のなかでもっとも分析が行いやすいサイドチャネルモデリング攻撃を行うため，ソフトウェア実装の PUF について分析を行った．今後は，今回分析したソフトウェア実装 PUF を用いて，モデルの精度や脆弱性要因の特定に向けた検証を行いたい．

**謝辞** 本発表の成果の一部は，国立研究開発法人新エネルギー・産業技術総合開発機構 (NEDO) の委託業務の結果得られたものです．

### 参考文献

[1] R. Pappu, “Physical one-way functions,” PhD Thesis, Massachusetts Institute of Technology, 2001

[2] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld, “Physical One-Way Functions,” *Science*, vol. 297, no. 5589, 2002, pp. 2026–2030.

[3] B. Gassend, D. Clarke, M. Van Dijk, and S. Devadas, “Silicon Physical Random Functions,” in *Proceedings of the 9th ACM conference on computer and communications security*. ACM, 2002, pp. 148–160.

[4] D. Lim, “Extracting Secret Keys from Integrated Circuits,” Master’s thesis, Massachusetts Institute of Technology, 2004.

[5] U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas, and J. Schmidhuber, “Modeling Attacks on Physical Unclonable Functions,” in *Proceedings of the 17th ACM conference on computer and communications security*. ACM, 2010, pp. 237–249.

[6] U. Rührmair, and J. Sölter, “PUF modeling attacks: An introduction and overview,” in *Proceedings of the conference on Design, Automation & Test in Europe*, 2014, p348.

[7] G. E. Suh and S. Devadas, “Physical Unclonable Functions for Device Authentication and Secret Key Generation,” in *Proceedings of the 44th annual Design Automation Conference*. ACM, 2007, pp. 9–14.

[8] G. T. Becker, “The Gap between Promise and Reality: on The Insecurity of XOR Arbiter PUFs,” in *International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*. Springer, 2015, pp. 535–555.

[9] J. Delvaux and I. Verbauwhede, “Side Channel Modeling Attacks on 65nm Arbiter PUFs Exploiting CMOS Device Noise,” in *Hardware-Oriented Security and Trust (HOST), 2013 IEEE International Symposium on*. IEEE, 2013, pp. 137–142.

[10] T. Machida, D. Yamamoto, M. Iwamoto, and K. Sakiyama, “A New Mode of Operation for Arbiter PUF to Improve Uniqueness on FPGA,” in *Computer Science and Information Systems (FedCSIS), 2014 Federated Conference on*, 2014, pp. 871–878.

[11] R. Yashiro, T. Machida, M. Iwamoto, and K. Sakiyama, “Deep-Learning-Based Security Evaluation on Authentication Systems Using Arbiter PUF and Its Variants,” in *International Workshop on Security*. Springer, 2016, pp. 267–285.

[12] T. Machida, D. Yamamoto, M. Iwamoto, and K. Sakiyama, “A New Arbiter PUF for Enhancing Unpredictability on FPGA,” *The Scientific World Journal*, 2015.

[13] Y. Hori, T. Yoshida, T. Katashita, and A. Satoh, “Quantitative and statistical performance evaluation of arbiter physical unclonable functions on FPGAs,” In *P*roceedings of 2010 International Conference on Reconfigurable Computing and FPGAs, ReConFig 2010, 2010, pp. 298–303.

[14] A. Maiti, G. Vikash, and S. Patrick, “A systematic method to evaluate and compare the performance of physical unclonable functions,” In *E*mbedded systems design with FPGAs, 2013, pp. 245–267.

[15] 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 VLD 設計技術研究会, 2018, VLD2017-127.