The Interpose PUF (iPUF): Secure PUF Design against State-of-the-art Machine Learning based Modeling Attacks

Phuong Ha Nguyen,
Durga P. Sahoo, Kaleel Mahmood, Chenglu Jin,
Ulrich Rührmair and Marten van Dijk

CHES 2019
1. Concept - Overview - Motivation
2. Strong PUFs: APUF, XOR APUF and Interpose PUF (iPUF)
3. Short-term Reliability
4. Reliability based modeling attacks on XOR PUF: understanding
5. Interpose PUF — a lightweight PUF which is secure against state-of-the art modeling attacks
6. Conclusion
1. Concept - Overview - Motivation
Concept - Overview — Motivation [1]

Challenge C → Hardware Primitive [Device] → Response R

Weak PUF - small #CRPs: RO PUF, SRAM PUF, etc.

Strong PUF - large #CRPs: Application: device Identification, authentication and crypto key generation

No Security Proof: Power Grid PUF, Clock PUF, Crossbard

Security Proof: LPN PUFs - heavy

Broken but lightweight: APUF, XOR APUF, Feed Forward PUF, Lightweight Secure PUF, Bistable Ring PUF, MPUF, etc.

Nature: process variation — physically unclonability - unique

PUF’s Category:
Concept - Overview — Motivation [2]

Challenge C  →  Hardware Primitive [Device]  →  Response R

Nature: process variation – physically unclonability - unique

Application: device Identification, authentication and crypto key generation
Concept - Overview — Motivation [3]

Challenge C  →  Hardware Primitive [Device]  →  Response R

PUF’s Category:

Weak PUF - small #CRPs: RO PUF, SRAM PUF, etc.

Strong PUF – large #CRPs:

- PUF’s Modeling Attacks on CRPs only
  - No Security Proof:
    - Power Grid PUF, Clock PUF, Crossbard
  - Security Proof:
    - LPN PUFs

- Broken but lightweight:
  - APUF, XOR APUF, Feed Forward PUF, Lightweight Secure PUF, Bistable Ring PUF, MPUF, etc.

- PUF’s Category:
  - Classical ML attacks—reliable CRPs:
    - Support Vector Machine (SVM), Logistic Regression (LR), Evolution Strategy (ES), Covariance Matrix Adaptation ES (CMA-ES), Perceptron, Boolean Attacks, Deep Neural Network Attacks (DNN)
  - Advanced ML attacks—noisy CRPs:
    - CMA-ES + noisy CRPs
Concept - Overview — Motivation [4]

Challenge C → Hardware Primitive [Device] → Response R

PUF's Category:
- Weak PUF - small #CRPs: RO PUF, SRAM PUF, etc.
- Strong PUF - large #CRPs:

PUF's Modeling Attacks with CRPs only:
- Classical ML attacks — reliable CRPs:
  Support Vector Machine (SVM), Logistic Regression (LR), Evolution Strategy (ES), Covariance Matrix Adaptation ES (CMA-ES), Perceptron, Boolean Attacks, Deep Neural Network Attacks (DNN)
- Advanced ML attacks — noisy CRPs: CMA-ES + noisy CRPs
**Concept - Overview — Motivation [5]**

- **Challenge C** → **Response R**

**PUF's Category:**

- **Weak PUF** - small #CRPs: RO PUF, SRAM PUF, etc.
- **Strong PUF** - large #CRPs:
  - Broken but lightweight: Arbiter PUF/APUF, XOR APUF, Feed Forward PUF, Lightweight Secure PUF, Bistable Ring PUF.
  - No Security Proof: Power Grid PUF, Clock PUF, Crossbar PUF
  - Security Proof: LPN PUFs - Large HW footprint

**PUF's Modeling Attacks with CRPs only:**

- **Classical ML attacks**
  - reliable CRPs:
    - Support Vector Machine (SVM), Logistic Regression (LR), Evolution Strategy (ES), Covariance Matrix Adaptation ES (CMA-ES), Perceptron, Boolean Attacks, Deep Neural Network Attacks (DNN)

- **Advanced ML attacks**
  - noisy CRPs: CMA-ES + noisy CRPs
Concept - Overview - Motivation [6]

- **Challenge C** → **Response R**
- **Hardware Primitive [Device]**
- **PUF’s Category:**
  - Weak PUF - small #CRPs: RO PUF, SRAM PUF, etc.
  - Strong PUF - large #CRPs:
- **PUF’s Modeling Attacks with CRPs only:**
  - Classical ML attacks - reliable CRPs: Arbiter PUF/APUF, XOR APUF, Feed Forward PUF, Lightweight Secure PUF, Bistable Ring PUF.
  - Advanced ML attacks - noisy CRPs: CMA-ES + noisy CRPs
- **Security Proof** → **Vulnerability**
  - Lightweight, Precise Math. Model
  - XOR APUF

Broken but lightweight:
Arbiter PUF/APUF, XOR APUF, Feed Forward PUF, Lightweight Secure PUF, Bistable Ring PUF.
Concept - Overview — Motivation [7]

- **Challenge C** → **Hardware Primitive [Device]** → **Response R**

**PUF’s Category:**
- **Weak PUF** - small #CRPs: RO PUF, SRAM PUF, etc.
- **Strong PUF** - large #CRPs:
  - **Classical ML attacks** - reliable CRPs: XOR APUF, Feed Forward PUF, Lightweight Secure PUF, Bistable Ring PUF.
  - **Advanced ML attacks** - noisy CRPs: CMA-ES + noisy CRPs

**Security Proof:**
- Lightweight, Precise Math. Model
- Interpose PUF (iPUF)
Concept - Overview — Motivation [8]

**Hardware Primitive [Device]**

- **Challenge C** → **Response R**

**PUF's Category:**
- **Weak PUF** - small #CRPs: RO PUF, SRAM PUF, etc.
- **Strong PUF** - large #CRPs:

**PUF's Modeling Attacks with CRPs only:**
- Classical ML attacks — reliable CRPs:
- Advanced ML attacks — noisy CRPs: CMA-ES + noisy CRPs

**Security Proof**

**Security Philosophy**

**Design Philosophy**

**Broken but lightweight:** Arbiter PUF/APUF, XOR APUF, Feed Forward PUF, Lightweight Secure PUF, Bistable Ring PUF.

**Lightweight, Precise Math. Model**

**XOR APUF** → **interpose PUF (iPUF)**
2. APUF- XOR APUF - iPUF
APUF, XOR APUF and iPUF [1]

- Extremely lightweight and large number of CRPs i.e, $2^n$ CRPs

- Environmental noises make the PUF’s outputs unreliable sometimes

- Not secure against modeling attacks
APUF, XOR APUF and iPUF [2]
The Interpose PUF / iPUF

\[ c = (c[0], \ldots, c[i], \ldots, c[n - 1]) \rightarrow x\text{-XOR APUF} \]

\[ (c[0], \ldots, c[i], a, c[i + 1], \ldots, c[n - 1]) \rightarrow y\text{-XOR APUF} \rightarrow r \]
Arbiter PUF (APUF)

x-XOR Arbiter PUF

Interpose PUF (iPUF)

\[ \Delta > 0 \rightarrow r = 1 \]

\[ \Delta = d_{\text{upper}} - d_{\text{lower}} = \omega \cdot \Psi \]

\[ \omega : \text{unique weight vector, delay representation for any APUF instance} \]

\[ \Psi_i = j = i, \ldots, n - 1, -c_j, i = 0, \ldots, n - 1, \Psi_n = 1 \]

Precise linear model + CRPs + ML = practically and softwarelly clonable

Precise non-linear model + CRPs + classical ML = impractically softwarelly clonable

\[ x \cdot \text{IPUF} + y \cdot \text{XOR PUF} \approx y \cdot \text{XOR PUF} \text{ if } a \text{ is inserted at the middle} \]

XOR APUF is not Secure against noisy CRPs + CMA-ES [Advanced ML]! (CHES2015) why?

Why not for iPUF?
APUF, XOR APUF and iPUF [5]

**Arbiter PUF (APUF)**

- \( \Delta > 0 \rightarrow r = 1 \). Otherwise \( r = 0 \).
- \( \Delta = d_{\text{upper}} - d_{\text{lower}} = w \cdot \Phi \).
- \( w \): unique for any APUF instance.
- \( \Phi \) is the parity vector.
  \[ \Phi[i] = \prod_{j=i,n-1} (1 - c[j]), i = 0, ..., n - 1, \Phi[n] = 1 \]

**x-XOR Arbiter PUF**

\[ A_0 \rightarrow r_0 \]
\[ \vdots \]
\[ A_{x-1} \rightarrow r_{x-1} \]
\[ \oplus \rightarrow r \]

**Interpose PUF (iPUF)**

- \( c = (c[0], ..., c[i], ..., c[n-1]) \)
- \( y \)-XOR APUF
- \( x \)-XOR APUF

- Precise linear model
- Large CRP space
- Vulnerable to ML attacks

A \( n \)-stage classic Arbiter PUF with challenge \( c \in \{0,1\}^n \).
Arbiter PUF (APUF)

- Δ > 0 → r = 1. Otherwise r = 0
- Δ = d_{upper} - d_{lower} = w \cdot \Phi
- w : unique for any APUF instance
- \Phi is the parity vector
- \Phi[i] = \prod_{j=i,...,n-1}(1 - c[j]), i = 0, ..., n - 1, \Phi[n] = 1

Precise linear model
- Large CRP space
- Vulnerable to ML attacks

x-XOR Arbiter PUF

Interpose PUF (iPUF)

- Precise non-linear model
- Large CRP space
- Secure against classical ML
- Vulnerable to advanced ML

\[ r_{XOR \text{ APUF}} = \text{sign} \left( \prod_{i=1}^{x} w_i^T \Phi \right) \] [2]
**APUF, XOR APUF and iPUF [7]**

**Arbiter PUF (APUF)**

- Precise linear model
- Large CRP space
- Vulnerable to ML attacks

**x-XOR Arbiter PUF**

- Precise non-linear model
- Large CRP space
- Secure against classical ML
- Vulnerable to advanced ML

**Interpose PUF (iPUF)**

- (x, y) - IPUF
  \[ \approx \left( y + \frac{x}{2} \right) - \text{XOR PUF} \]
  if a is inserted at the middle

- **Precise linear model**
- Large CRP space
- Vulnerable to ML attacks
### APUF, XOR APUF and iPUF [8]

**Arbiter PUF (APUF)**

- \( \Delta > 0 \rightarrow r = 1 \). Otherwise \( r = 0 \)
- \( \Delta = d_{\text{upper}} - d_{\text{lower}} = w \cdot \Phi \)
- \( w \): unique for any APUF instance
- \( \Phi \) is the parity vector

\[
\Phi[i] = \prod_{j=i,n-1} (1 - c[j]), i = 0, ..., n - 1, \Phi[n] = 1
\]

**x-XOR Arbiter PUF**

- Precise non-linear model
- Large CRP space
- Secure against classical ML
- Vulnerable to advanced ML

**Interpose PUF (iPUF)**

- Precise non-linear model
- Large CRP space
- Secure both classical ML and advanced ML

\[
(x, y) - \text{IPUF} \approx \left( y + \frac{x}{2} \right) - \text{XOR PUF}
\]

if \( a \) is inserted at the middle

---

A n-stage classic Arbiter PUF with challenge \( c \in \{0, 1\}^n \).
3. Short-term Reliability
Arbiter: Repeatability — short-term Reliability [1]

Challenge C (C,1), APUF A

Δ = w · Φ + noise

Measurements

Δ₁, Δ₁ < 0,

Responses

r₁, (r₁ = 0),

Reliability Measurements

Δ₁, Δ₂, …, Δₘ

Δ₁ < 0,

Δ₂ > 0,

…,

Δₘ < 0

Δ = w · Φ + noise

Δ = w · Φ > 0, r = 0

Δ = w · Φ > 0, r = 1

w

Challenge C $(C_1, C_2)_e$

APUF A

Measurements
$\Delta_1, \Delta_2$
$\Delta_1 < 0$, $\Delta_2 > 0$

Responses
$r_1, r_2$
$r_1 = 0$, $r_2 = 1$

$\Delta = w \cdot \Phi + \text{noise}$

$\Delta = w \cdot \Phi > 0$, $r = 0$

$\Delta = w \cdot \Phi > 0$, $r = 1$

$w$
Challenge C
(C,1), (C,2), ...
(C,m)

APUF A

Measurements
Δ₁, Δ₂, ..., Δₘ
Δ₁ < 0,
Δ₂ > 0,
..., Δₘ < 0

Responses
r₁, r₂, ..., rₘ
(r₁ = 0),
(r₂ = 1),
..., (rₘ = 0)

Δ = w · Φ + noise

Δ₁ < 0,
Δ₂ > 0,
..., Δₘ < 0

Δ = w · Φ > 0, r = 0
Δ = w · Φ > 0, r = 1

Arbiter: Repeatability — short-term Reliability [3]

Challenge C
(C₁,1), (C₂,2), ...., (Cₘ,m)

APUF A

Δ = w · Φ + noise

Δ₁, Δ₂, ..., Δₘ
Δ₁ < 0,
Δ₂ > 0,
 ..., 
Δₘ < 0

Responses
r₁, r₂, ..., rₘ
(r₁ = 0),
(r₂ = 1),
 ..., 
(rₘ = 0)

Reliability
R = (r₁+r₂+...+rₘ)/m

Challenge C
(C,1),
(C,2),
....
(C,m)

Δ = w · Φ + noise

APUF A

Measurements
Δ₁, Δ₂, ..., Δₘ
Δ₁ < 0,
Δ₂ > 0,
....
Δₘ < 0

Responses
r₁, r₂, ..., rₘ
(r₁ = 0),
(r₂ = 1),
....
(rₘ = 0)

Reliability
R = (r₁ + r₂ + ... + rₘ)/m

Δ = w · Φ > 0, r = 0

Δ = w · Φ > 0, r = 1

Δ = w ⋅ Φ + noise

Measurements
Δ₁, Δ₂, ..., Δₘ
Δ₁ < 0,
Δ₂ > 0,
..., Δₘ < 0

Responses
r₁, r₂, ..., rₘ
(r₁ = 0),
(r₂ = 1),
..., (rₘ = 0)

Reliability
R = (r₁ + r₂ + ... + rₘ)/m

Δ = w ⋅ Φ > 0, r = 0

Δ = w ⋅ Φ > 0, r = 1

\[ \Delta = w \cdot \Phi + \text{noise} \]

**Challenge C**

\( (C,1), (C,2), \ldots, (C,m) \)

**APUF**

**Measurements**

\( \Delta_1, \Delta_2, \ldots, \Delta_m \)

\( \Delta_1 < 0, \Delta_2 > 0, \ldots, \Delta_m < 0 \)

**Responses**

\( r_1, r_2, \ldots, r_m \)

\( \begin{align*}
  (r_1 &= 0), \\
  (r_2 &= 1), \\
  \ldots \\
  (r_m &= 0)
\end{align*} \)

**Reliability**

\( R = \frac{r_1 + r_2 + \ldots + r_m}{m} \)

\[ \Delta = w \cdot \Phi > 0, r = 0 \]

Reliable

Noisy

\[ \Delta = w \cdot \Phi > 0, r = 1 \]

Reliability of C and \( w \) are related

The Gap Between Promise and Reality: On the Insecurity of XOR Arbiter PUFs CHES, 2015, Georg T. Becker
4. Reliability based Modeling Attacks
A n-stage classic Arbiter PUF with challenge $c \in \{0, 1\}^n$.

- $\Delta > 0 \rightarrow r = 1$. Otherwise $r = 0$
- $\Delta = d_{upper} - d_{lower} = w \cdot \Phi$
Covariance Matrix Adaptation Evolution Strategy (CMA-ES) Algorithm

\( w: \text{target} \)

\( \hat{w}: \text{estimator or model} \)

\[ \hat{w} = w \]
Reliability-based modeling attack on APUFs using CMAES [1]

\[ \Delta = w \cdot \Phi > 0, r = 0 \]

\[ \Delta = w \cdot \Phi < 0, r = 1 \]

\( Q: \) set of CRPs, 
\( w: \) APUF

\( Q_n: \) noisy challenges

\( Q_r: \) reliable challenges
Reliability-based modeling attack on APUFs using CMAES [2]

**Target**

\[ Q, w \]

- \( Q_n \): noisy
- \( Q_r \): reliable

**Model**

\[ Q, \epsilon_1, \hat{w}_1 \]

\[
Q \rightarrow \text{challenge } c \rightarrow \Phi(c) \rightarrow \Delta = \hat{w}_1 \cdot \Phi(c)
\]

- \(|\Delta| \leq \epsilon_1 \rightarrow \text{challenge } c \text{ is noisy} \)
- \(|\Delta| > \epsilon_1 \rightarrow \text{challenge } c \text{ is reliable} \)

Iteration 1
Reliability-based modeling attack on APUFs using CMAES [3]

Iteration 1

\[ Q, \epsilon_1, \hat{W}_1 \]

\[ Q_n, Q_r, Q_n, Q_r, \epsilon_1 \]

\[ Q_n : \text{noisy} \]

\[ Q_r : \text{reliable} \]
Reliability-based modeling attack on APUFs using CMAES [4]

Iteration 1

$Q_n$: noisy

$Q_r$: reliable

$Q, w$

$Q, \epsilon_1, \hat{W}_1$

Compute the matching rate $\rho_1$ between $[Q, w]$ and $[Q, \epsilon_1, \hat{W}_1]$
Reliability-based modeling attack on APUFs using CMAES [5]
Reliability-based modeling attack on APUFs using CMAES [6]

Iteration 1

$Q, w$

$Q_n$: noisy

$Q_r$: reliable

$Q, \epsilon_1, \hat{W}_1$

$Q, \epsilon_2, \hat{W}_2$

$Q, \epsilon_i, \hat{W}_i$

$Q, \epsilon_k, \hat{W}_k$

$\rho_1$

$\rho_2$

$\rho_i$

$\rho_k$

High matching rate: Kept

Low matching rate: Discarded
Reliability-based modeling attack on APUFs using CMAES [7]

Iteration 2

High matching rate: Kept
Reliability-based modeling attack on APUFs using CMAES [8]

Iteration M

$Q, w$

$Q_n : \text{noisy}$

$Q_r : \text{reliable}$

$Q, \epsilon_1, \hat{w}_1$

$Q, \epsilon_2, \hat{w}_2$

$Q, \epsilon_1, \hat{w}_1$

$Q, \epsilon_k, \hat{w}_k$

$\rho_1$

$\rho_2$

$\rho_1$

$\rho_k$

High matching rate: Kept
$x$-XOR APUF

$x$-XOR PUF.
Reliability-based modeling attack on XOR APUFs using CMAES [1]

\[ \bar{w}_1, \ldots, \bar{w}_k \text{ are STILL models of APUF} \]

The model of one APUF \( i \) among \( x \) APUFs in \( x \)-XOR APUF

High matching rate: Kept

CMA-ES attack on XOR APUF
Reliability-based modeling attack on XOR APUFs using CMA-ES [2]

XOR APUF $\rightarrow Q_{\text{new}}$

$Q, w$

$Q_n: \text{noisy}$

$Q_r: \text{reliable}$

CMA-ES attack on XOR APUF

High matching rate: Kept

$\hat{w}_1, \ldots, \hat{w}_k$ are STILL models of APUF

The model of another APUF $j$ among $x$ APUFs in $x$-XOR APUF
Question 1: How does the attack on XOR PUF work?

Question 2: How can we make the attack on XOR PUF fail?
Question 1: How does the attack on XOR PUF work?
The noisy and reliable challenges in XOR PUF

XOR APUF

APUF 1

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q

Q
Key idea of the attack on XOR PUF [1]

- (1) All the models $\hat{w}_i$ in CMA-ES are models of APUF
Key idea of the attack on XOR PUF [2]

- (1) All the models $\hat{\omega}_i$ in CMA-ES are models of APUF
- (2) $\hat{\omega}_i$ can only converge to an APUF instance
Key idea of the attack on XOR PUF [3]

- (1) All the models $\hat{w}_i$ in CMA-ES are models of APUF
- (2) $\hat{w}_i$ can only converge to an APUF instance
- (3) CMA ES maximizes the matching $Q$ of $\hat{w}_i$ and $Q$ of XOR APUF
Key idea of the attack on XOR PUF [4]

- (1) All the models $\hat{w}_i$ in CMA-ES are models of APUF
- (2) $\hat{w}_i$ can only converge to an APUF instance
- (3) CMA ES maximizes the matching $Q$ of $\hat{w}_i$ and $Q$ of XOR APUF
- (1)+(2)+(3) CMA ES forces $\hat{w}_i$ converges to APUF 10 because $Q$ of APUF 10 is the representative of $Q$ of XOR APUF.
Key idea of the attack on XOR PUF [5]

- Changing $Q$ makes $Q_3$ largest

10-XOR APUF | APUF 3 | $Q$, $\epsilon_i$, $\hat{W}_i$ | $Q$ | $Q_r$ | $Q_n$ | $Q_n$ : noisy | $Q_r$ : reliable

Noisy | Reliable | All PUF models | Challenge-Reliability Pairs Training Data
Key idea of the attack on XOR PUF [6]

- Changing Q makes $Q_3$ largest

Keep changing Q and applying CMA-ES attack on Q to get models of all APUF instances
Question 2: How to make the attack on XOR PUF fail?
Attack fails

2-XOR APUF with majority voting circuit at $A_0$

CMA ES never converges to APUF $A_0$ and always converges to APUF $A_1$ when majority voting mechanism in use.
5. Interpose PUF (iPUF) — Reliability based modeling attack resistance
Security of iPUF wrt Reliability-based modeling attack [1]

- **Reason 1**: the information of APUF instances in x-XOR PUF presented at the iPUF output is less compared to APUF instances in y-XOR PUF. Thus, the reliability based modeling attack never converges to any APUF instance in x-XOR PUF.
- **Reason 2:** to attack APUFs at $y$-XOR APUF, the adversary needs to compute $\Delta$. But compute $\Delta$ is infeasible because the output of $x$-XOR PUF ($a$) is not known.

\[
\Delta = \Delta(n - 1) = w \cdot \Phi^T
\]

\[
\Phi[i] = \prod_{j=i, i\neq n}^{n-1} (1 - c[j]), \quad i = 0, \ldots, n-1
\]

\[
\Phi[n] = 1
\]
Other Contributions

- **Theoretical**
  - Enhanced Reliability based Modeling Attacks on APUF and XOR APUFs
  - Proved Logistic Regression on XOR APUF is the best attack
  - Proved Logistic Regression on iPUF is not applicable

- **Engineering**
  - Implemented APUF, XOR, and iPUF on FPGA
  - Studied good and bad FPGA-implemented APUF based PUF
  - All source codes available online: https://github.com/scluconn/DA_PUF_Library/

- **Detailed tutorial online:**
  https://www.youtube.com/playlist?list=PLK5NNs4GceLQw7bOEHSdZ0wHlmSF1ZvSW
6. Conclusion

- We explain how the reliability-based modeling attack on XOR PUF works.
- We propose a new lightweight PUF design (iPUF) which is secure against the state-of-the-art of modelling attacks.
Literature

1. [https://slideplayer.com/slide/3927633/](https://slideplayer.com/slide/3927633/)

2. Cryptanalysis of electrical PUFs via machine learning algorithms – Master Thesis of Jan Solter


Thank you for your attention! and any questions?