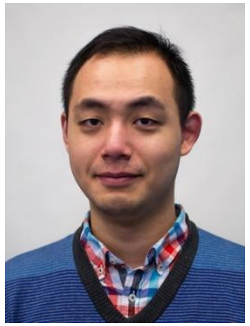
The Interpose PUF (iPUF): Secure PUF Design against State-of-the-art Machine Learning based Modeling Attacks

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

Ha

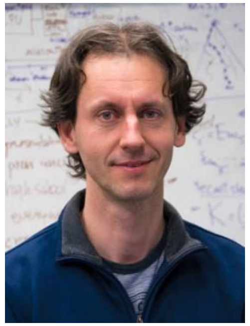
Durga



Chenglu

Kaleel

Uli



Marten

CHES 2019

UCONN

Content

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

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:

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

UCONN

Concept - Overview – Motivation [5]

Challenge C → Hardware Primitive [Device] → Response R

PUF's Category:

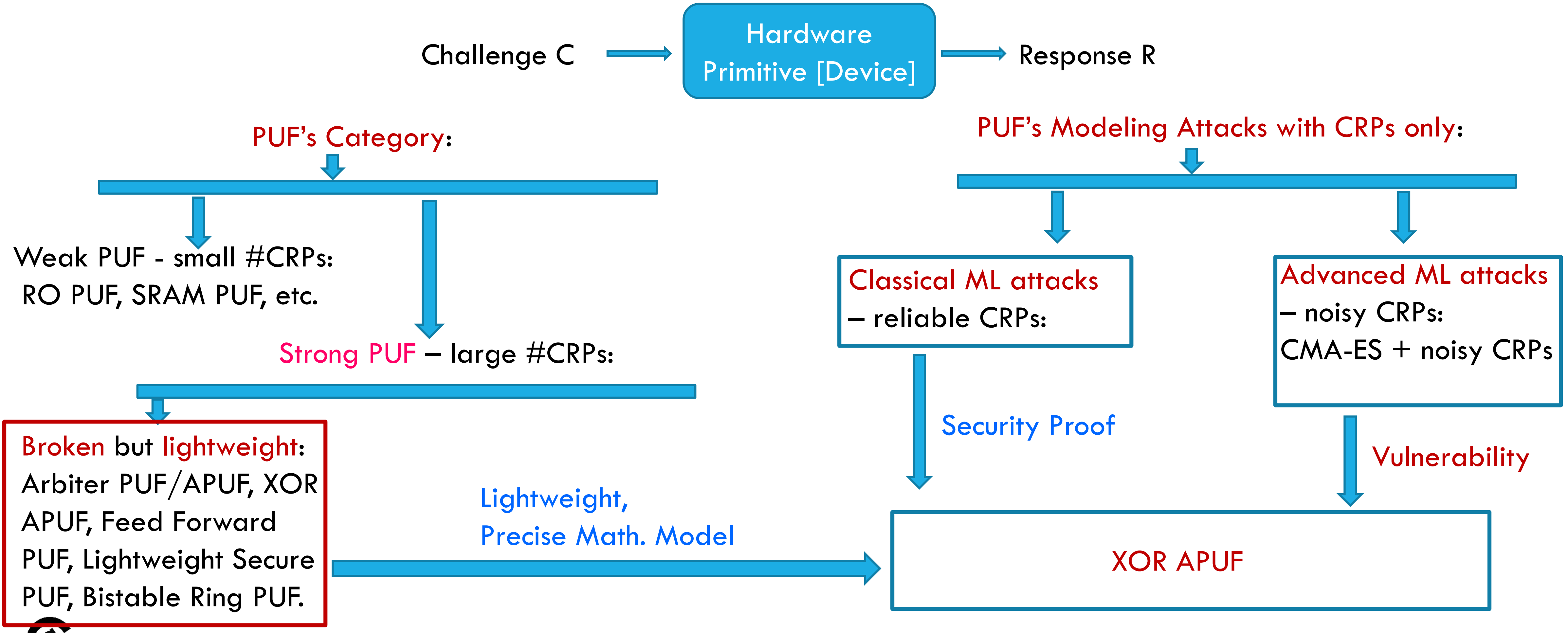
- Weak PUF - small #CRPs: RO PUF, SRAM PUF, etc.
- Strong PUF – large #CRPs:
  - **Broken** but **lightweight**: Arbiter PUF/APUF, XOR APUF, Feed Forward PUF, Lightweight Secure PUF, Bistable Ring PUF.
  - **No Security Proof:** Power Grid PUF, Clock PUF, Crossbar PUF
  - **Security Proof:** LPN PUFs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

UCONN

Concept - Overview – Motivation [6]



UCONN

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:

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

Lightweight,
Precise Math. Model

PUF's Modeling Attacks with CRPs only:

Classical ML attacks
– reliable CRPs:

Security Proof

Advanced ML attacks
– noisy CRPs:
CMA-ES + noisy CRPs

Security Proof

XOR APUF → interpose PUF (iPUF)

UCONN

Concept - Overview – Motivation [8]

Challenge C → Hardware Primitive [Device] → 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.

Lightweight,
Precise Math. Model

Design Philosophy

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

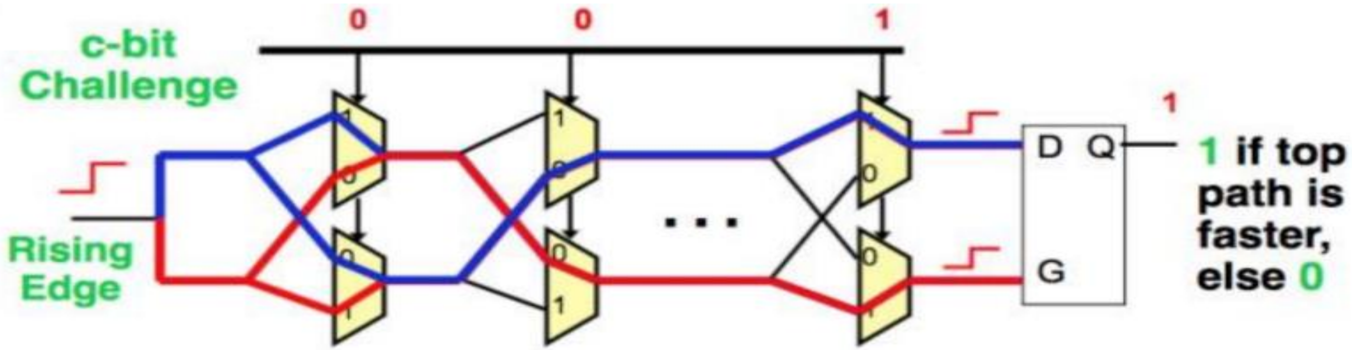
Security Proof

XOR APUF → interpose PUF (iPUF)

2. APUF- XOR APUF -iPUF

UCONN

APUF, XOR APUF and iPUF [1]



Arbiter PUF (APUF) [1]

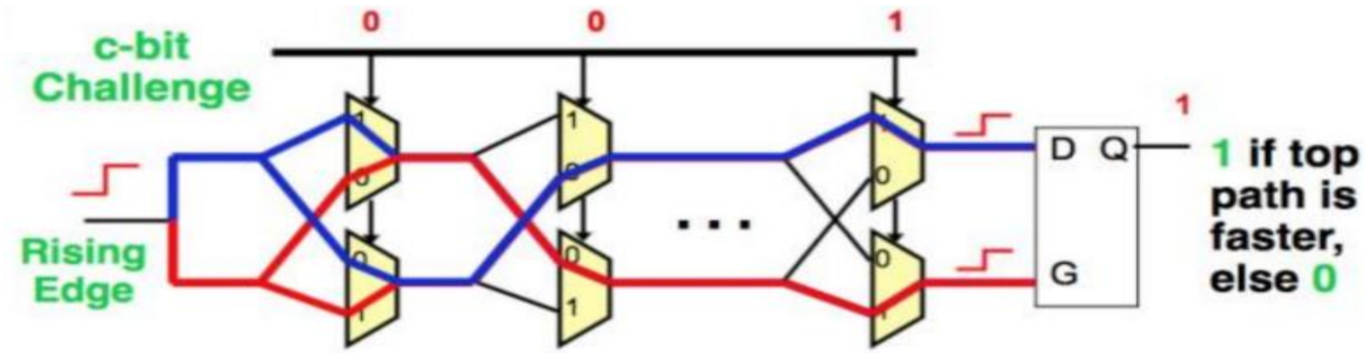
- Extremely lightweight and large number of CRPs i.e, 2^n CRPs

- Environmental noises make the PUF's outputs unreliable sometimes

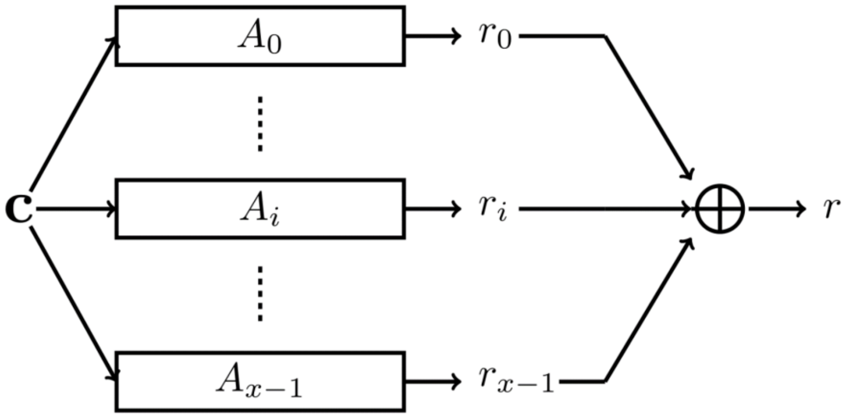
- Not secure against modeling attacks

APUF, XOR APUF and iPUF [2]

Arbiter PUF (APUF)



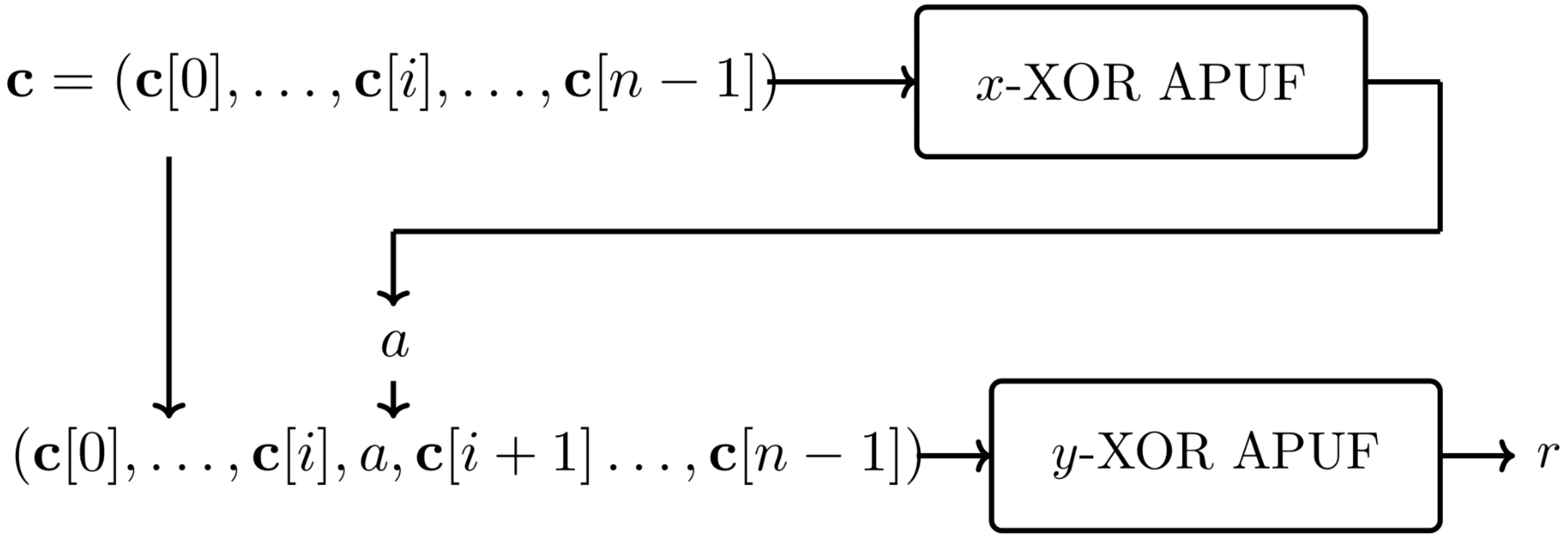
x-XOR APUF



x-XOR PUF.

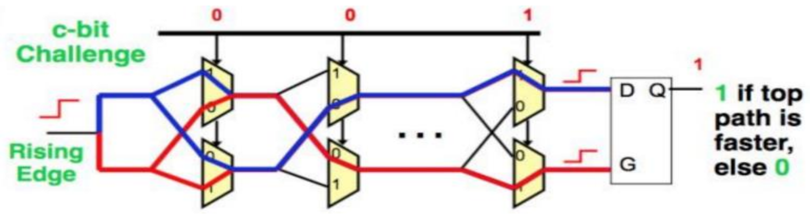
APUF, XOR APUF and iPUF [3]

The Interpose PUF / iPUF

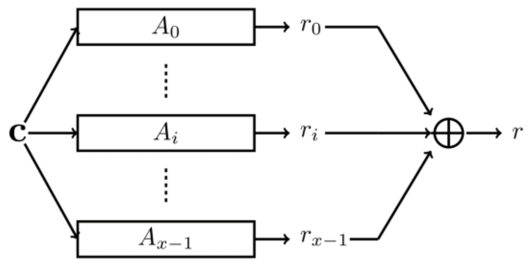


APUF, XOR APUF and iPUF [4]

Arbiter PUF (APUF)

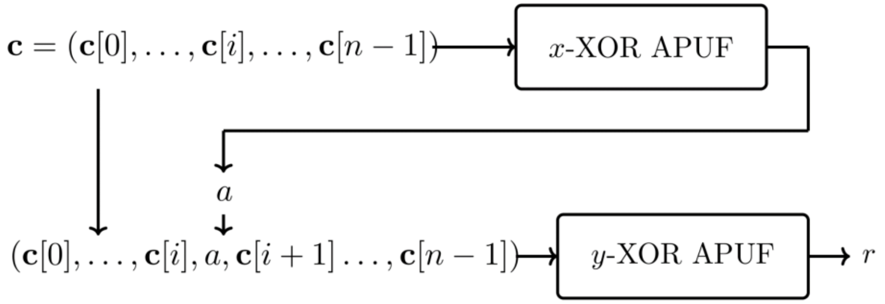


x-XOR Arbiter PUF



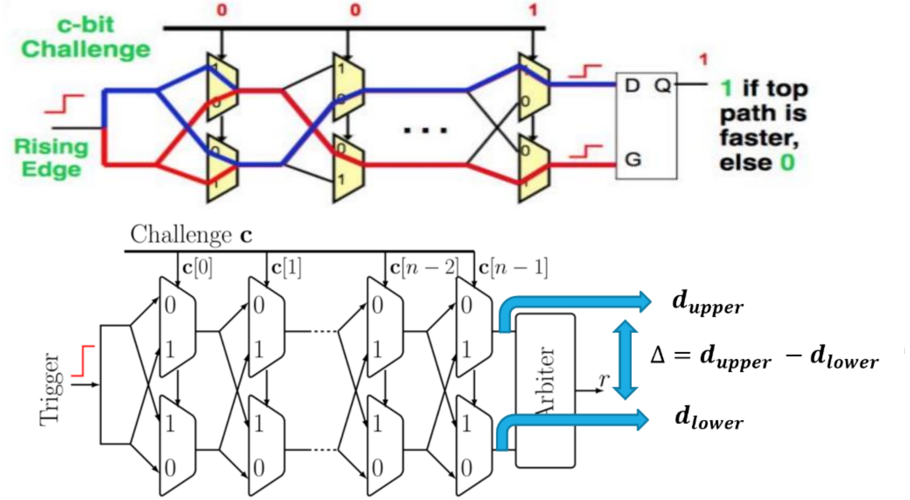
x-XOR PUF.

Interpose PUF (iPUF)



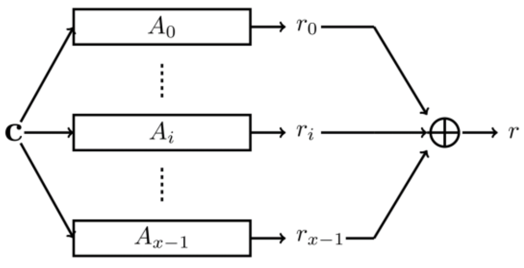
APUF, XOR APUF and iPUF [5]

Arbiter PUF (APUF)



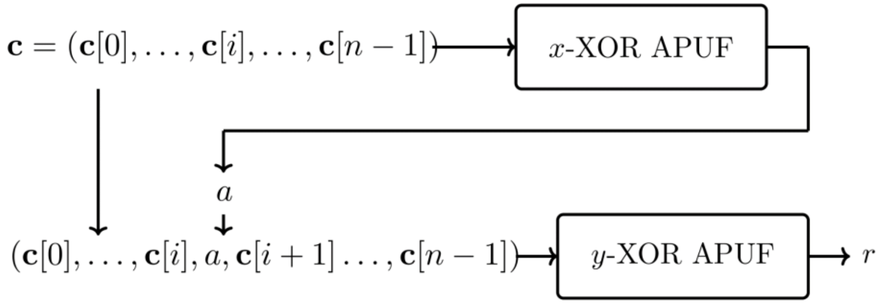
A n-stage classic Arbiter PUF with challenge $\mathbf{c} \in \{0,1\}^n$.

x-XOR Arbiter PUF



x-XOR PUF.

Interpose PUF (iPUF)



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

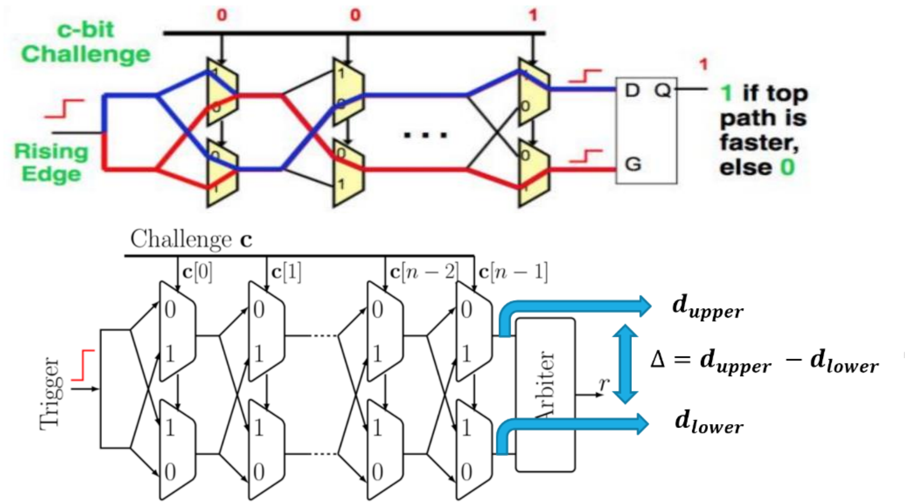
$\boldsymbol{\Phi}[i] = \prod_{j=i,\ldots,n-1}(1 - \boldsymbol{c}[j]), i = 0, \ldots, n-1\,,\ \boldsymbol{\Phi}[n] = 1$

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

APUF, XOR APUF and iPUF [6]

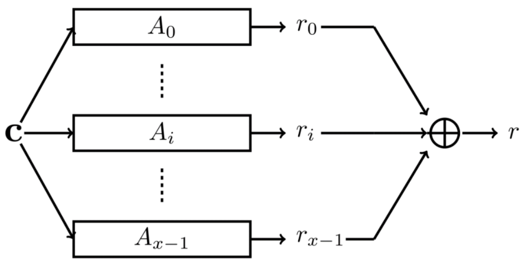
UCONN

Arbiter PUF (APUF)



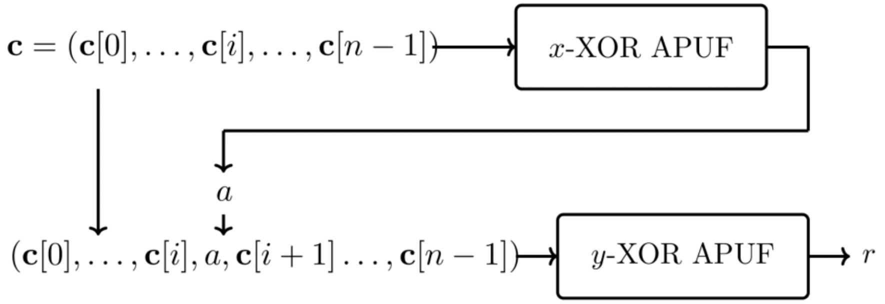
A n-stage classic Arbiter PUF with challenge $\mathbf{c} \in \{0,1\}^n$.

x-XOR Arbiter PUF



x-XOR PUF.

Interpose PUF (iPUF)



$$r_{\text{XOR APUF}} = sign\left(\prod_{i=1}^{x} w_i^{\mathrm{T}} \Phi\right)$$ [2]

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

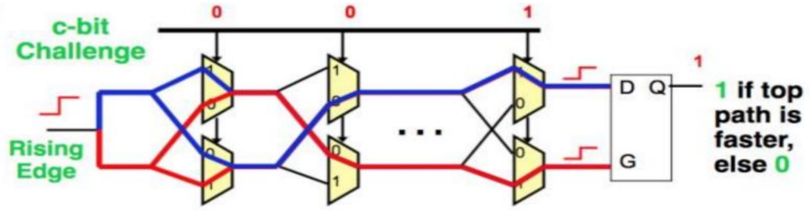
$\boldsymbol{\Phi}[i] = \prod_{j=i,\dots,n-1}(1 - \boldsymbol{c}[j]), i = 0, \dots, n-1,\ \boldsymbol{\Phi}[n] = 1$

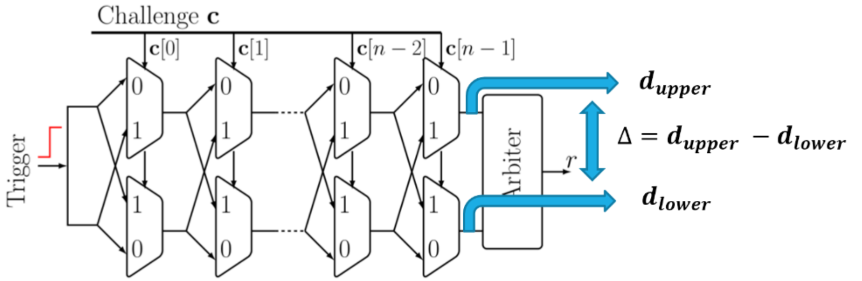
- Precise linear model
- Large CRP space
- Vulnerable to ML attacks

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

APUF, XOR APUF and iPUF [7]

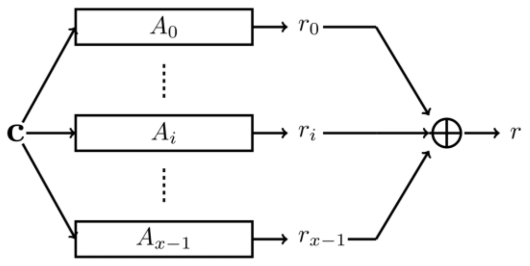
Arbiter PUF (APUF)





A n-stage classic Arbiter PUF with challenge $\mathbf{c} \in \{0,1\}^n$.

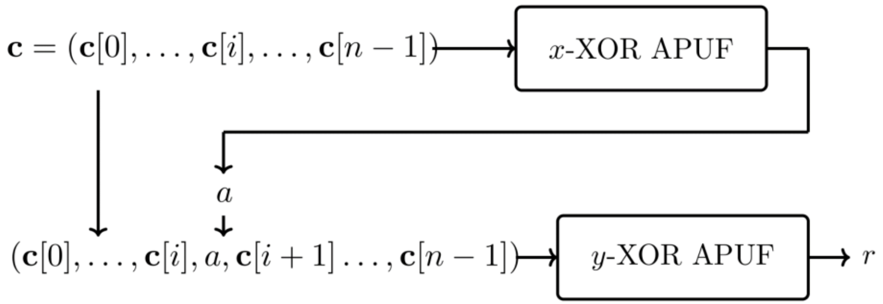
x-XOR Arbiter PUF



x-XOR PUF.

$$r_{\text{XOR APUF}} = sign\left(\prod_{i=1}^{x} w_i^{\mathrm{T}} \Phi\right)$$

Interpose PUF (iPUF)



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

if a is inserted at the middle

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

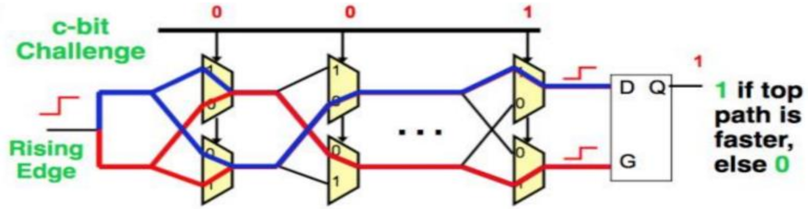
$\boldsymbol{\Phi}[i] = \prod_{j=i,\ldots,n-1}(1 - \boldsymbol{c}[j]), i = 0, \ldots, n-1\ ,\ \boldsymbol{\Phi}[n] = 1$

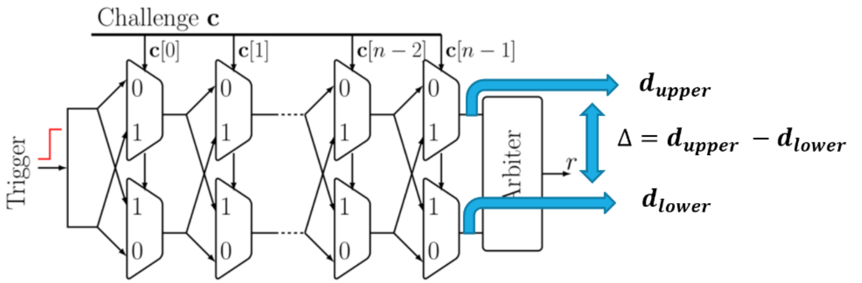
- Precise linear model
- Large CRP space
- Vulnerable to ML attacks

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

APUF, XOR APUF and iPUF [8]

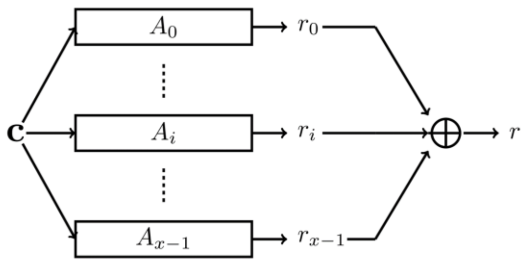
Arbiter PUF (APUF)





A n-stage classic Arbiter PUF with challenge $\mathbf{c} \in \{0,1\}^n$.

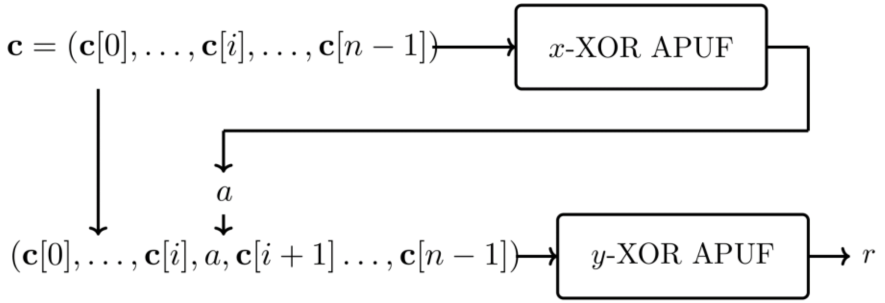
x-XOR Arbiter PUF



x-XOR PUF.

$$r_{\text{XOR APUF}} = sign\left(\prod_{i=1}^{x} w_i^{\mathrm{T}} \Phi\right)$$

Interpose PUF (iPUF)



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

if a is inserted at the middle

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

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

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

x-XOR Arbiter PUF:

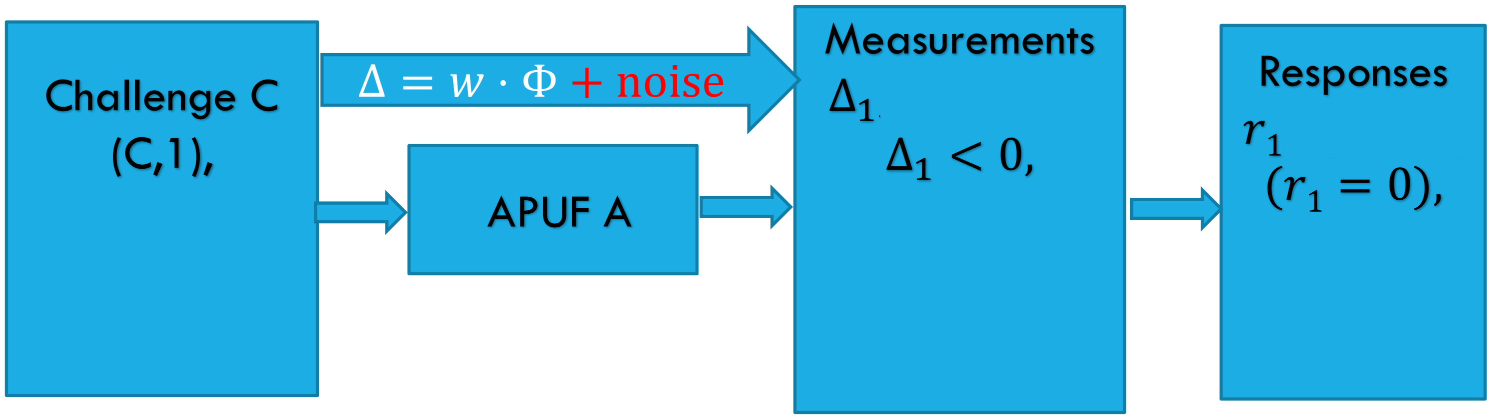
- Precise non-linear model
- Large CRP space
- Secure against classical ML
- Vulnerable to advanced ML

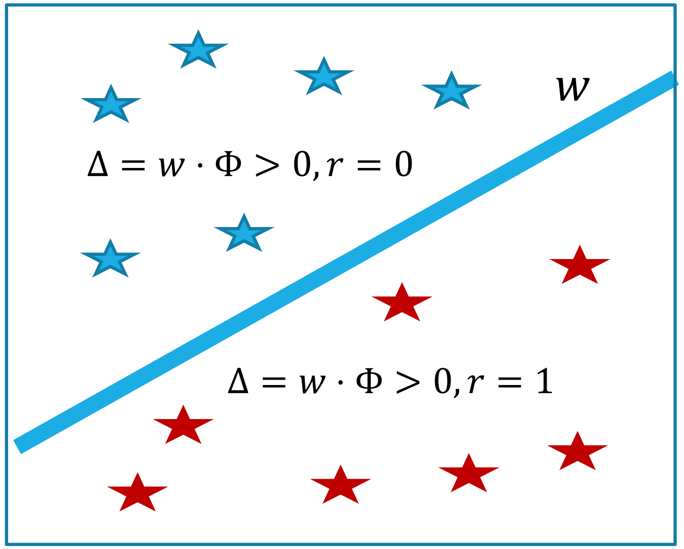
Interpose PUF (iPUF):

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

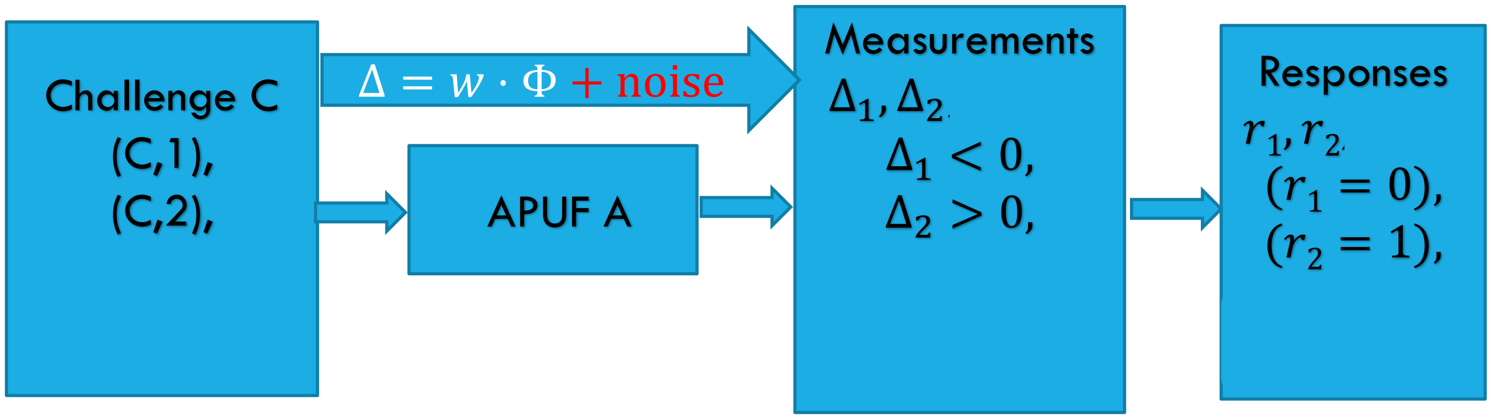
3. Short-term Reliability

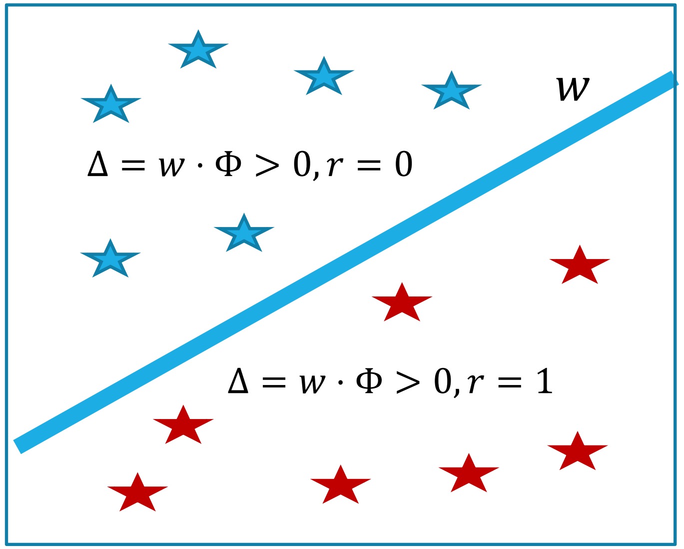
Arbiter: Repeatability – short-term Reliability [1]



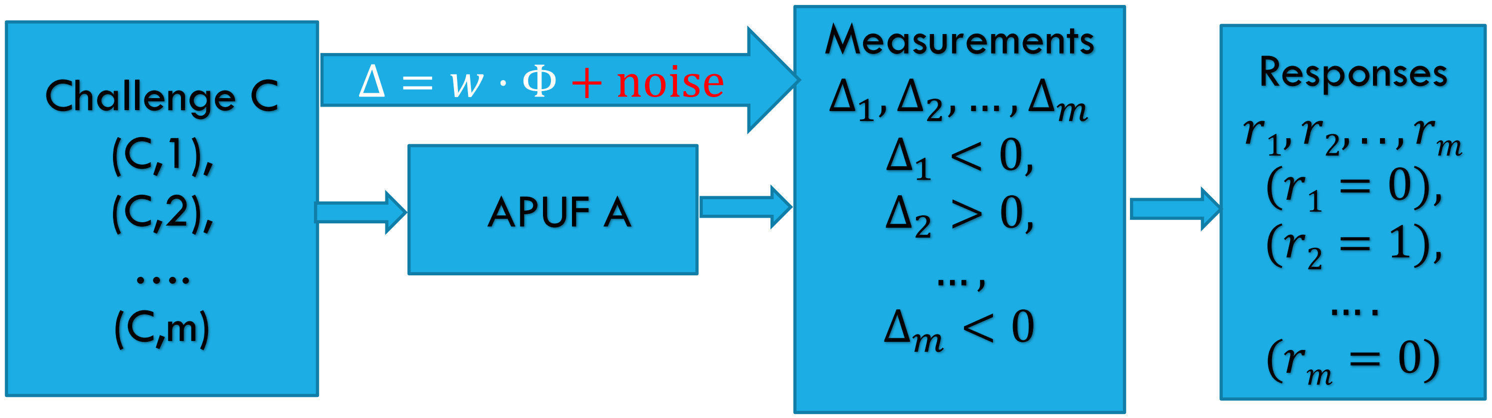


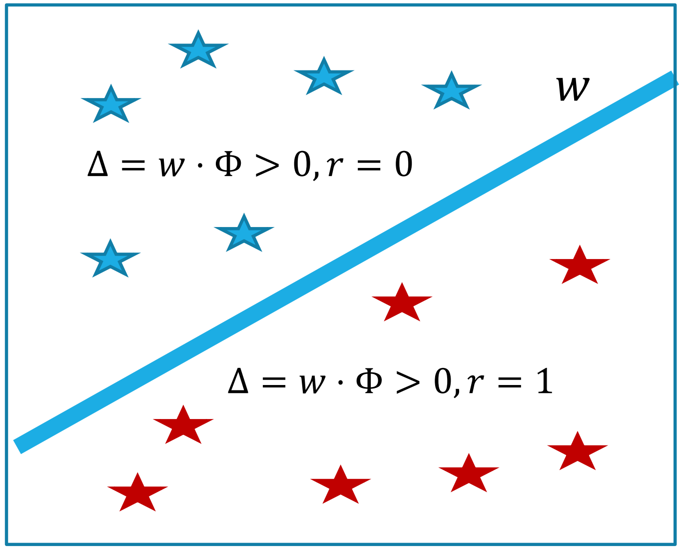
Arbiter: Repeatability – short-term Reliability [2]





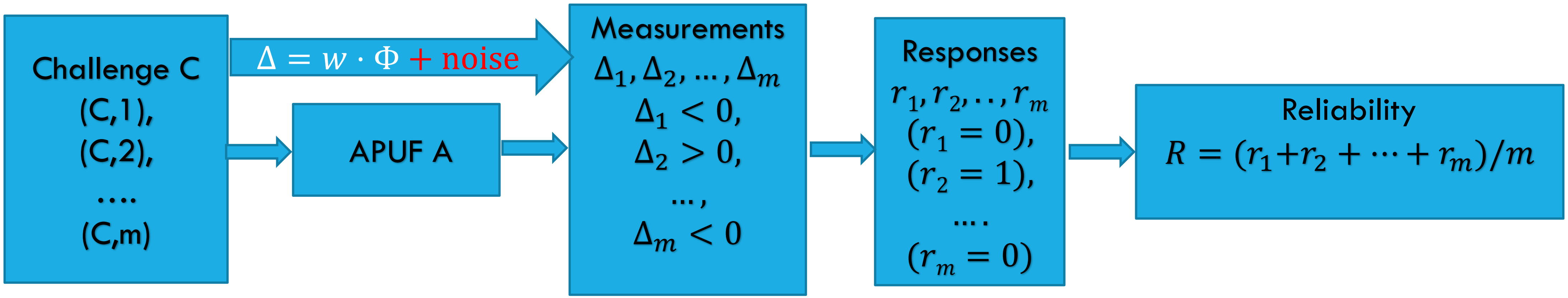
Arbiter: Repeatability – short-term Reliability [3]



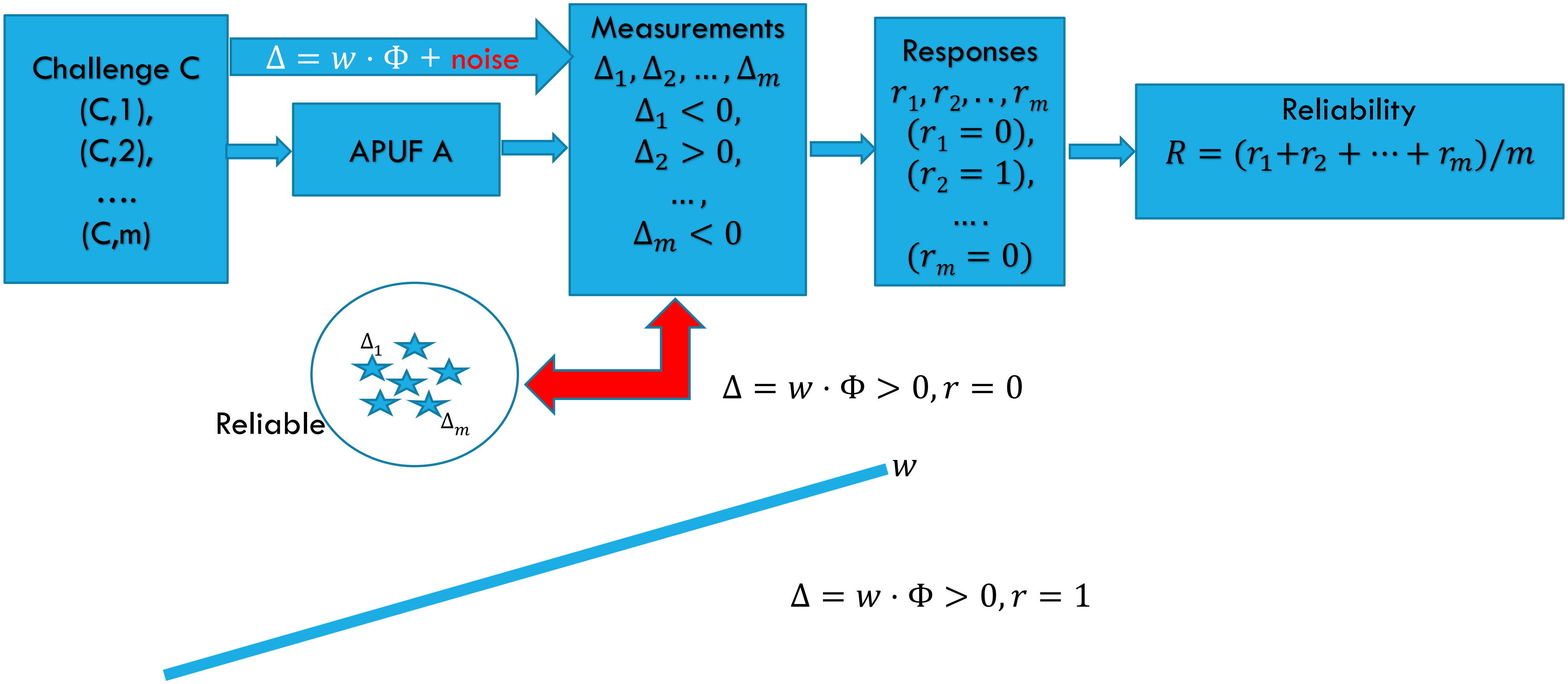


UCONN

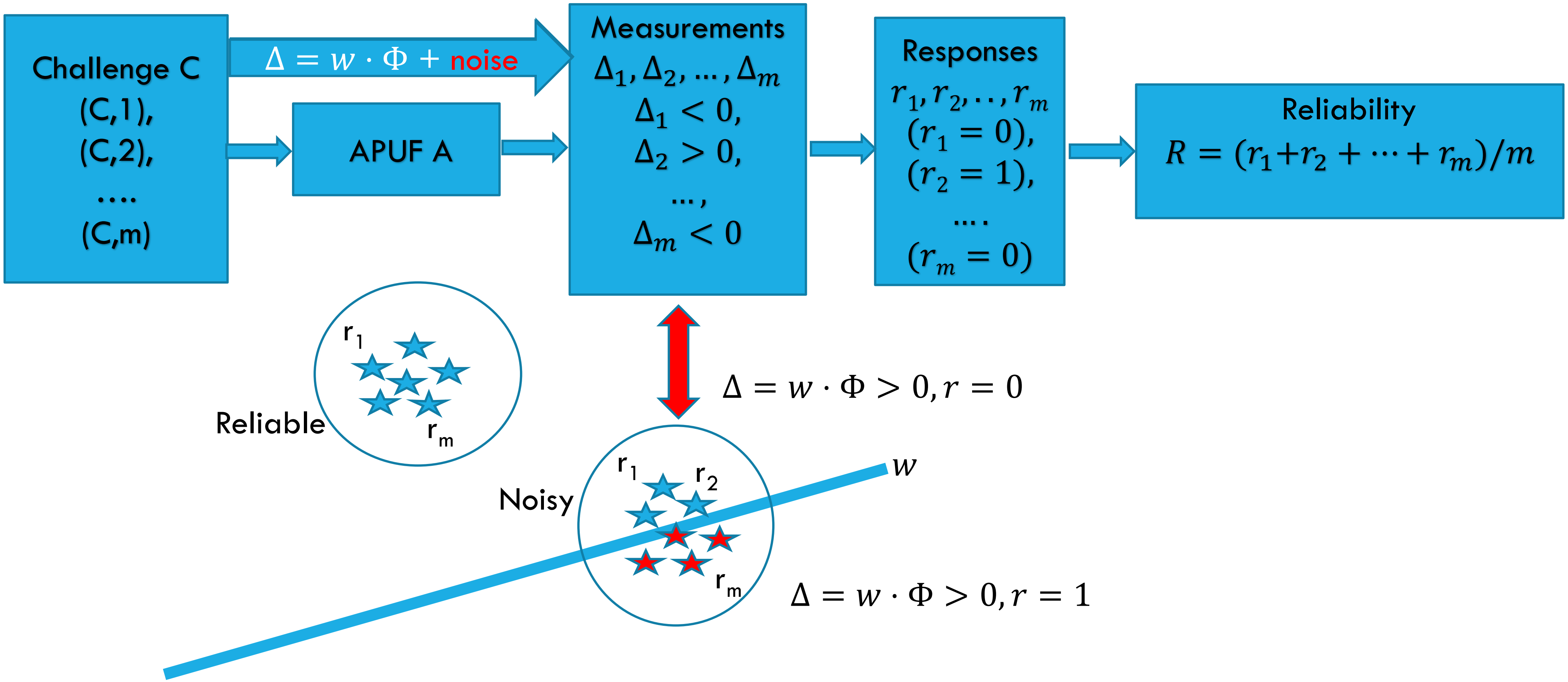
Arbiter: Repeatability – short-term Reliability [4]



Arbiter: Repeatability – short-term Reliability [5]

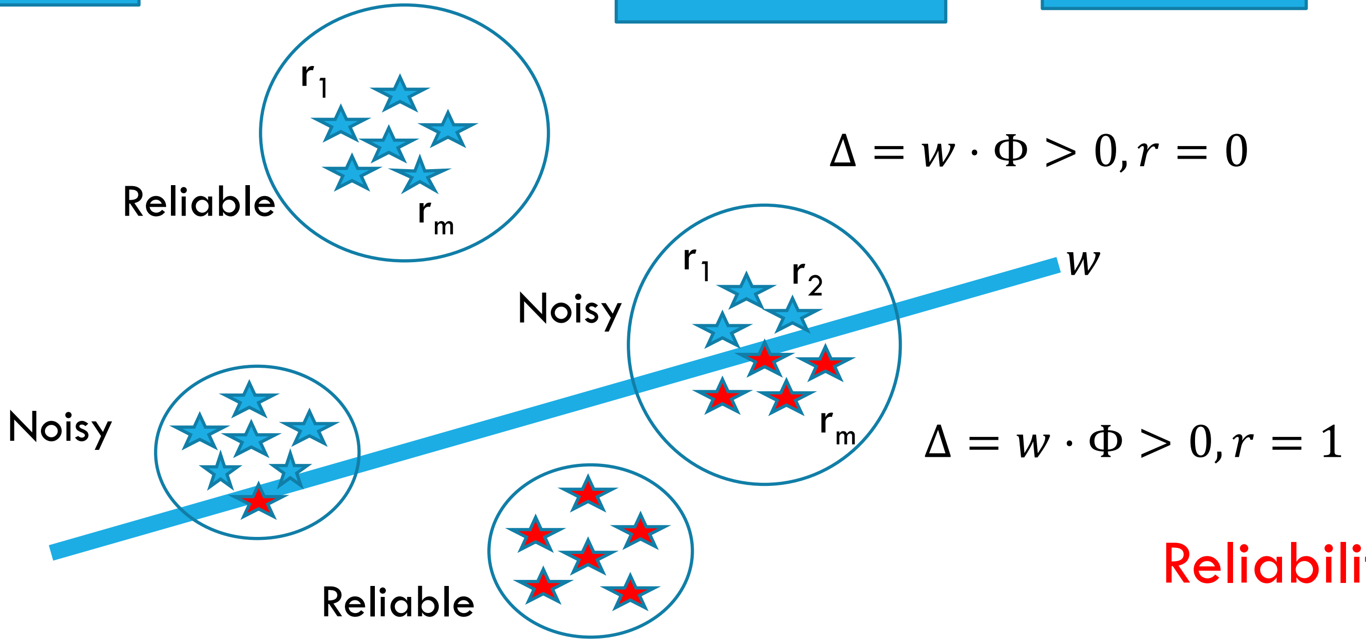


Arbiter: Repeatability – short-term Reliability [6]



Arbiter: Repeatability – short-term Reliability [7]

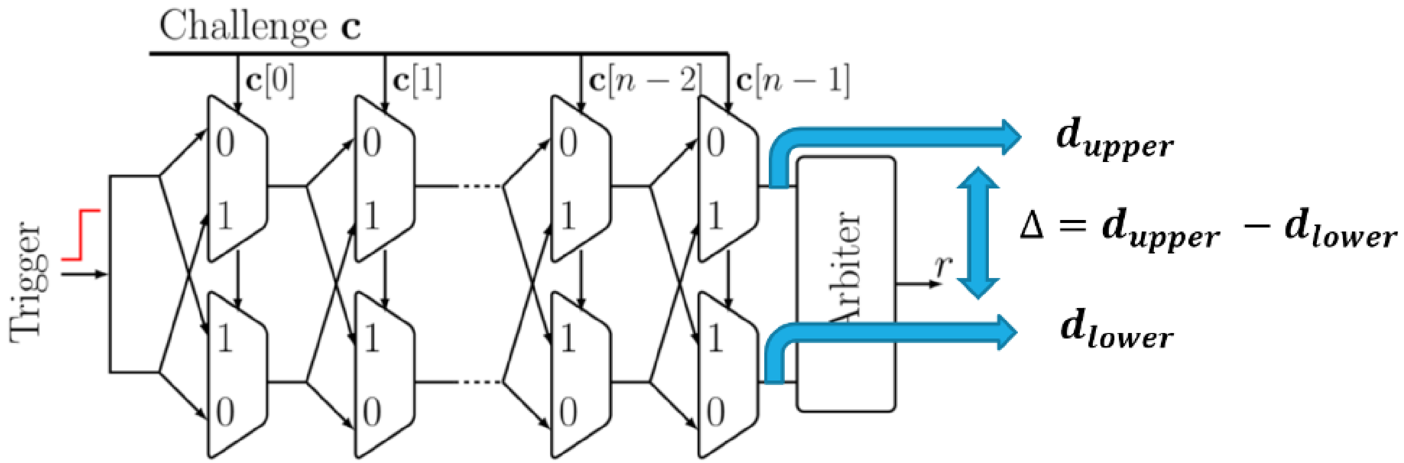
Challenge C		APUF A		Measurements		Responses		Reliability
(C,1), (C,2), …. (C,m)	$\Delta = w \cdot \Phi + \text{noise}$ →		→	$\Delta_1, \Delta_2, \dots, \Delta_m$ $\Delta_1 < 0,$ $\Delta_2 > 0,$ $\dots,$ $\Delta_m < 0$	→	$r_1, r_2, \dots, r_m$ $(r_1 = 0),$ $(r_2 = 1),$ …. $(r_m = 0)$	→	$R = (r_1 + r_2 + \dots + r_m)/m$



Reliability of C and w are related

4. Reliability based Modeling Attacks

APUF



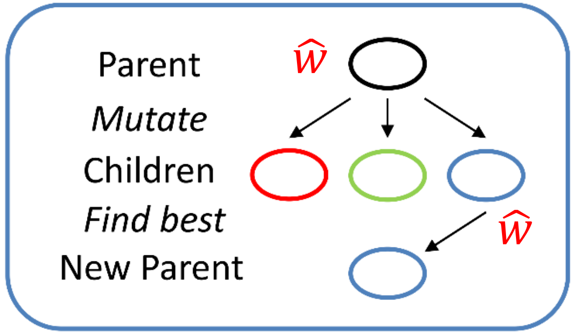
A n-stage classic Arbiter PUF with challenge $\mathbf{c} \in \{0,1\}^n$.

- $\Delta > 0 \rightarrow r = 1. Otherwise\ r = 0$
- $\Delta = \boldsymbol{d_{upper}} - \boldsymbol{d_{lower}} = \boldsymbol{w} \cdot \boldsymbol{\Phi}$

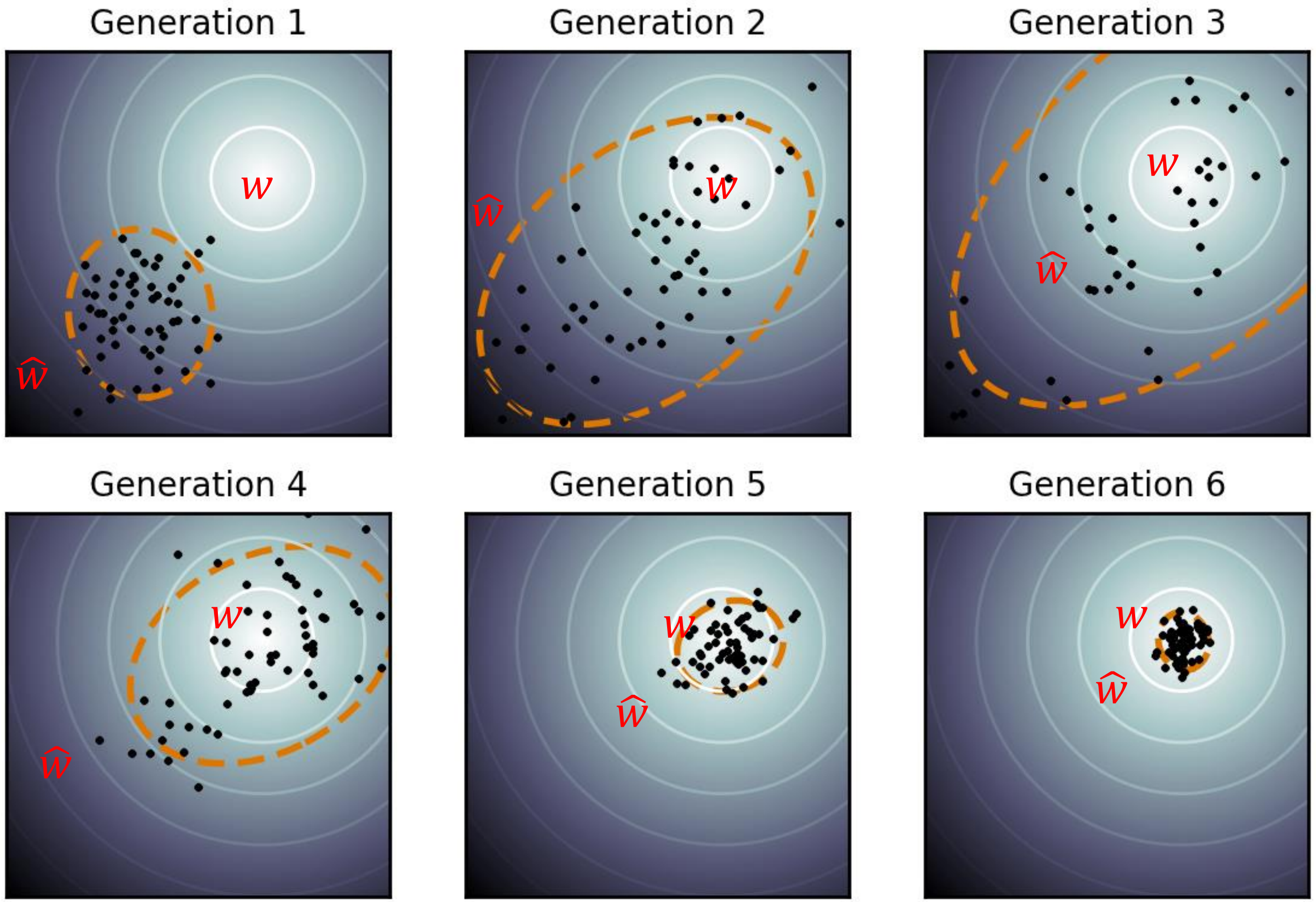
Covariance Matrix Adaptation Evolution Strategy (CMA-ES) Algorithm

$w: target$

$\hat{w}: estimator\ or\ model$

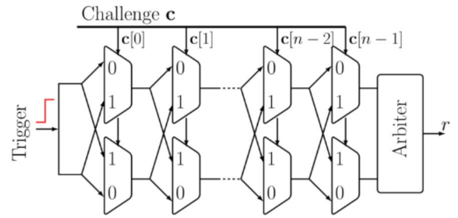


[4] $\hat{w} = w$



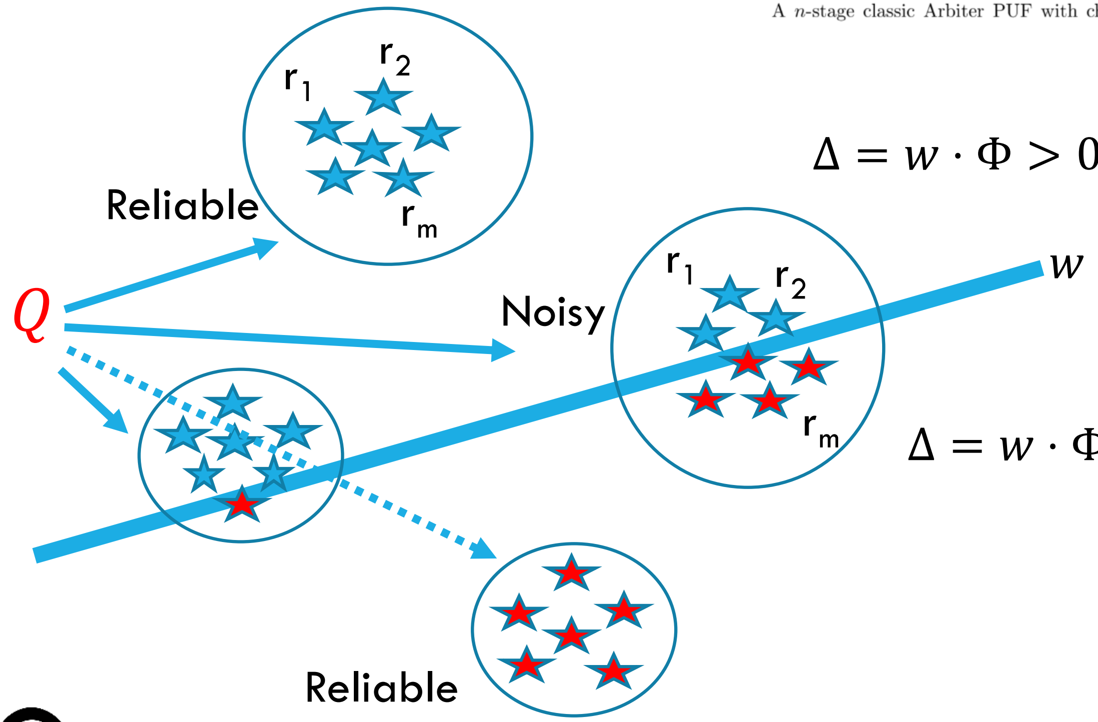
UCONN

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



A n-stage classic Arbiter PUF with challenge $\mathbf{c} \in \{0,1\}^n$.

Q: $set\ of\ CRPs$,
w: $APUF$



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

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

Q, w

Q_n : noisy challenges

Q_r : reliable challenges

Reliability-based modeling attack on APUFs using CMAES [2]

Iteration 1

Target

Q, w

Q_n : noisy

Q_r : reliable

Model

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

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

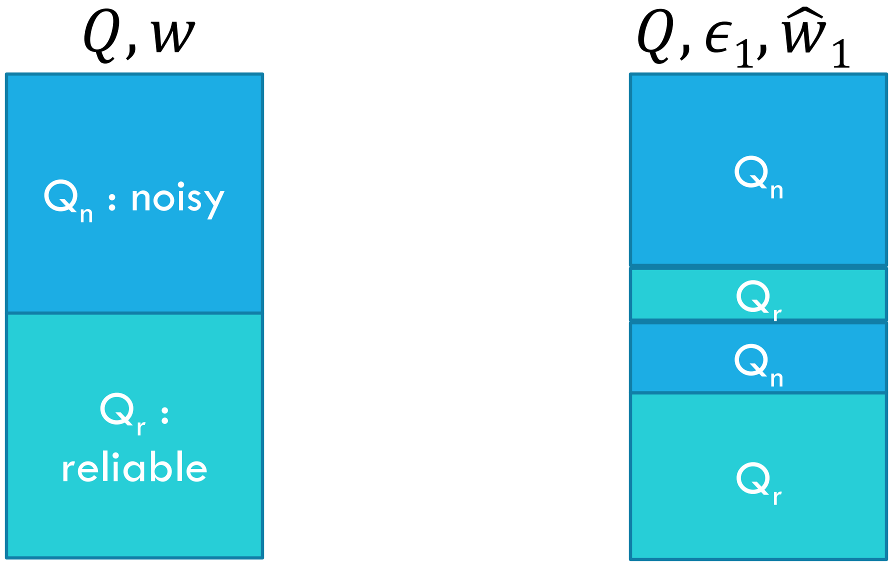
$$|\Delta| \leq \epsilon_1 \rightarrow \ \ challenge\ c\ is\ noisy$$ → Q_n : noisy

$$|\Delta| > \epsilon_1 \rightarrow \ \ challenge\ c\ is\ reliable$$ → Q_r : reliable

UCONN

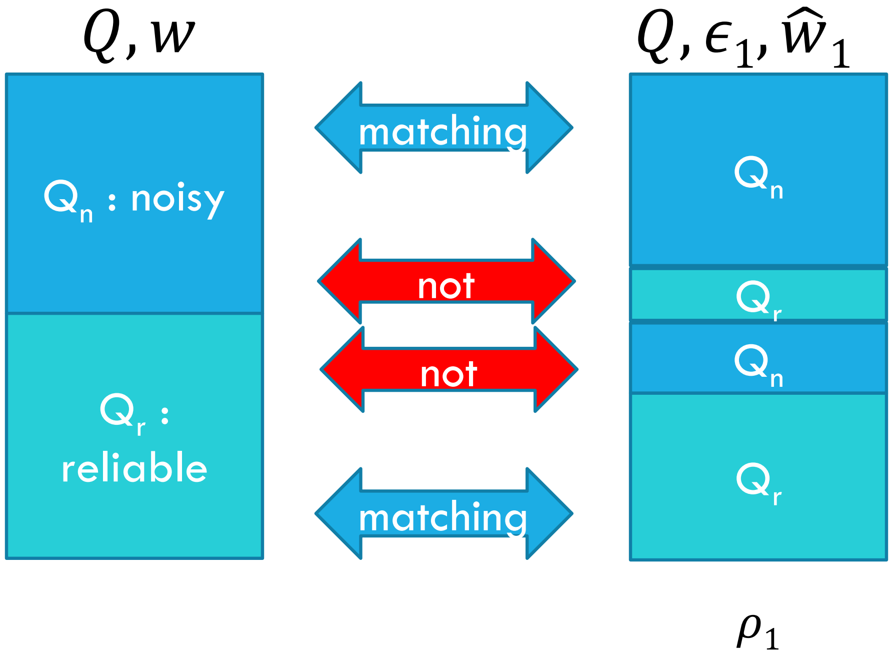
Reliability-based modeling attack on APUFs using CMAES [3]

Iteration 1



Reliability-based modeling attack on APUFs using CMAES [4]

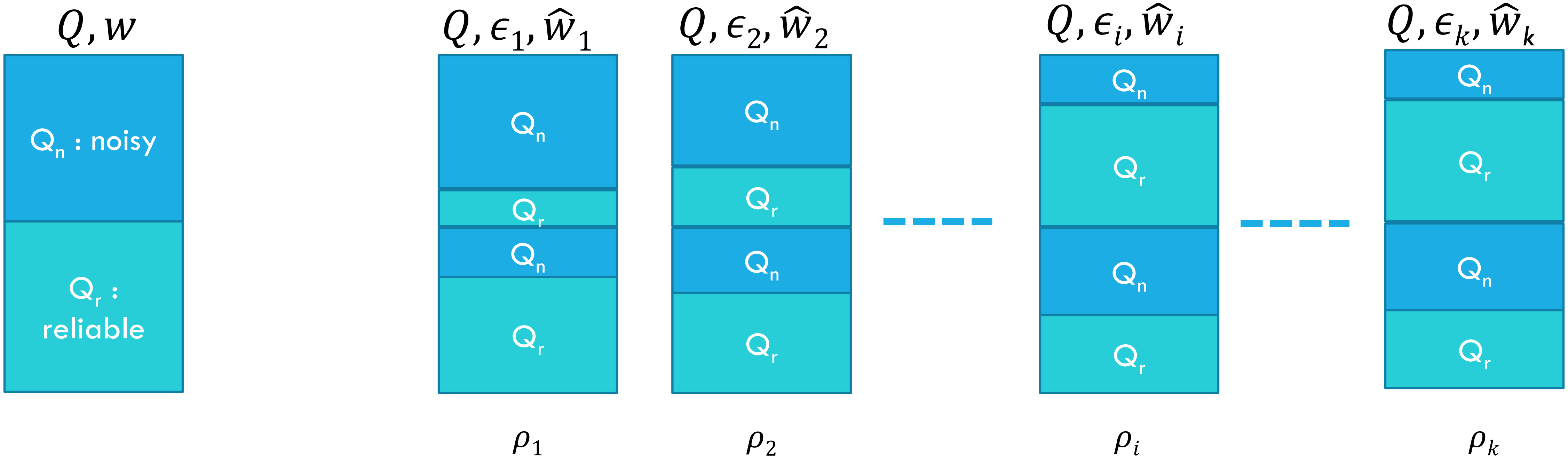
Iteration 1



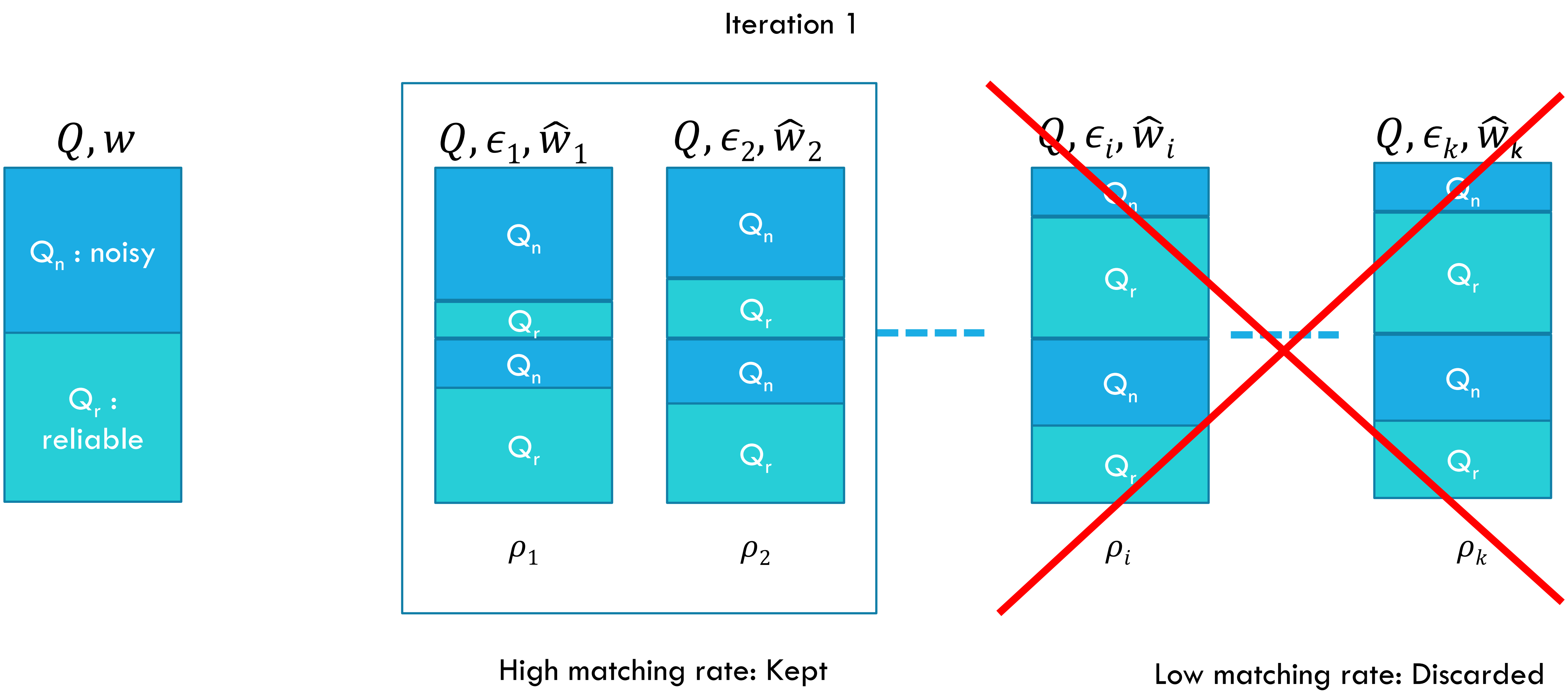
Compute the matching rate ρ_1 *between* $[Q, w]$ *and* $[Q, \epsilon_1, \hat{w}_1]$

Reliability-based modeling attack on APUFs using CMAES [5]

Iteration 1

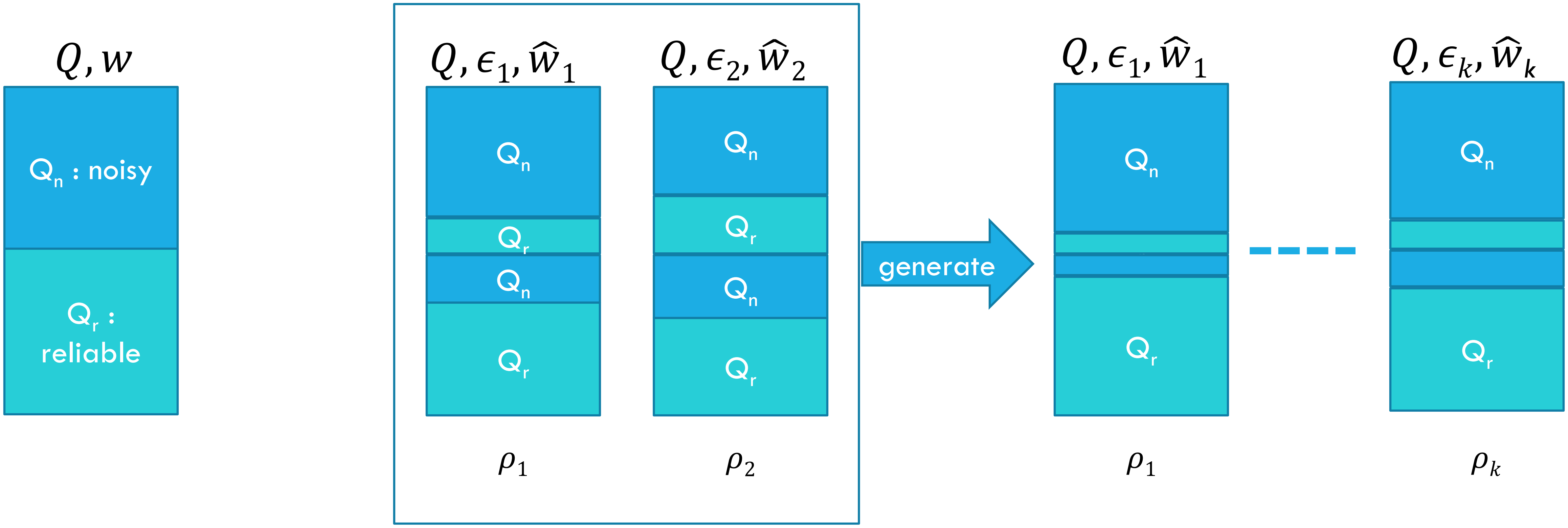


Reliability-based modeling attack on APUFs using CMAES [6]



Reliability-based modeling attack on APUFs using CMAES [7]

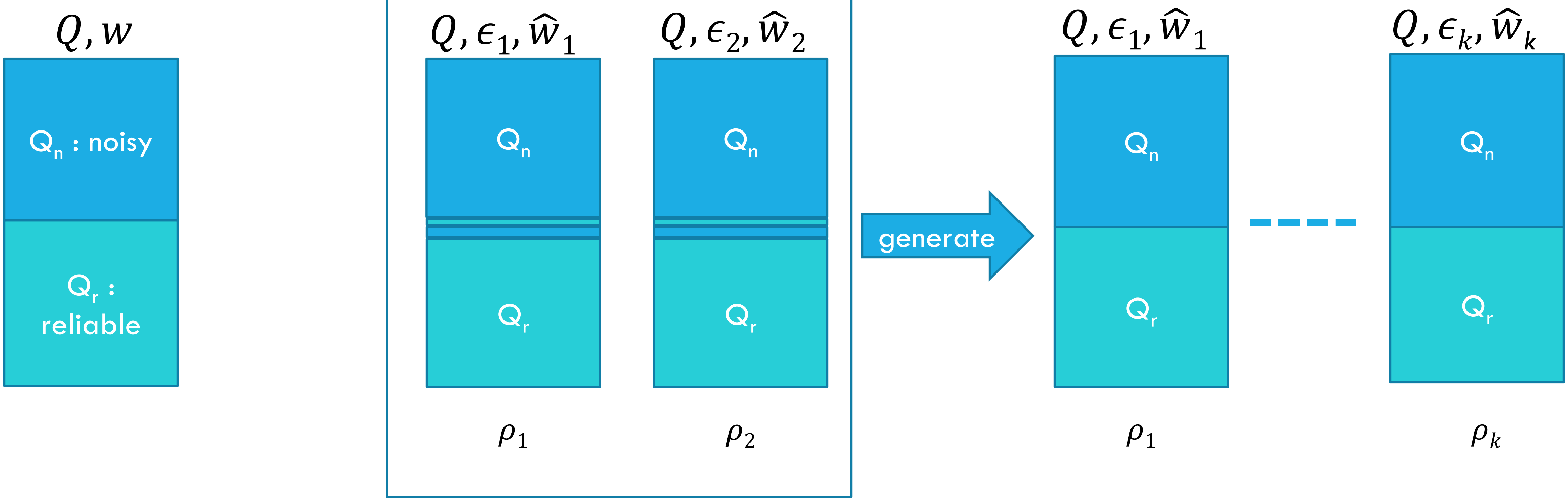
Iteration 2



High matching rate: Kept

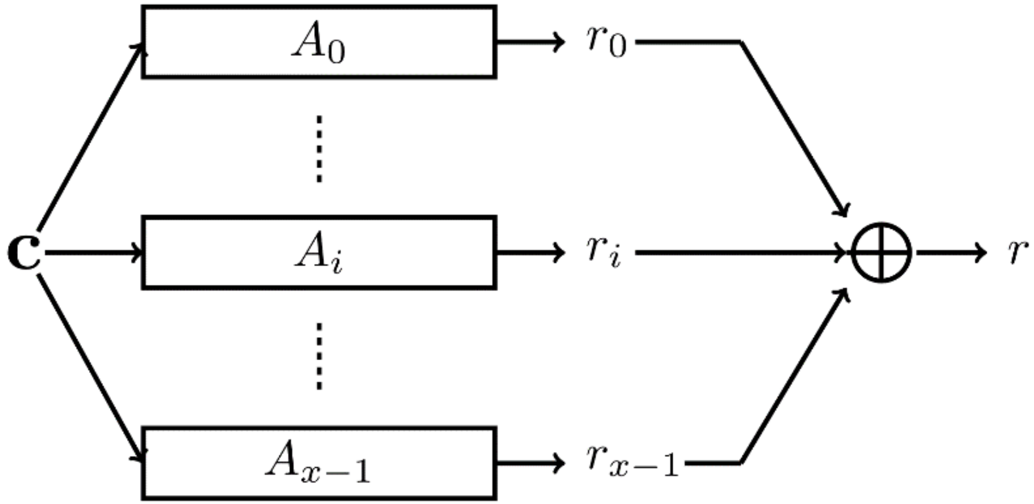
Reliability-based modeling attack on APUFs using CMAES [8]

Iteration M



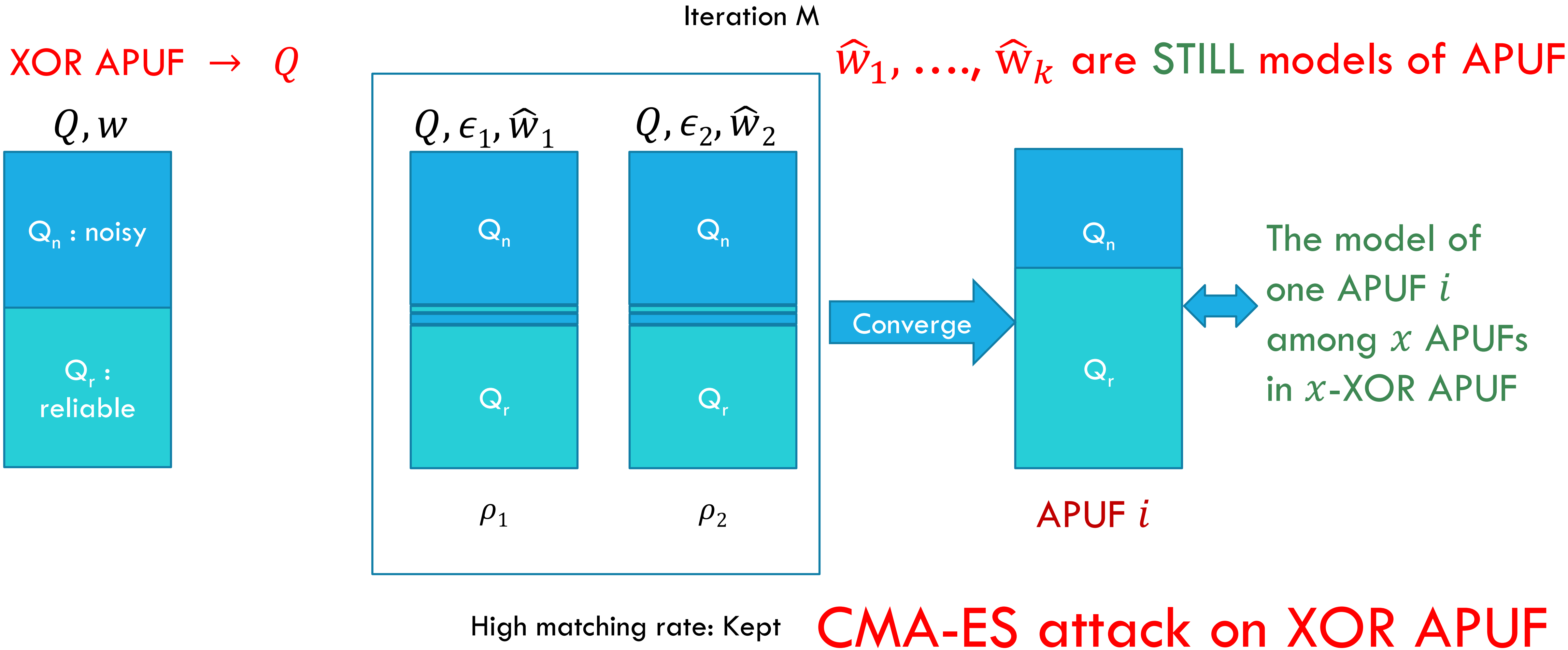
High matching rate: Kept

x-XOR APUF



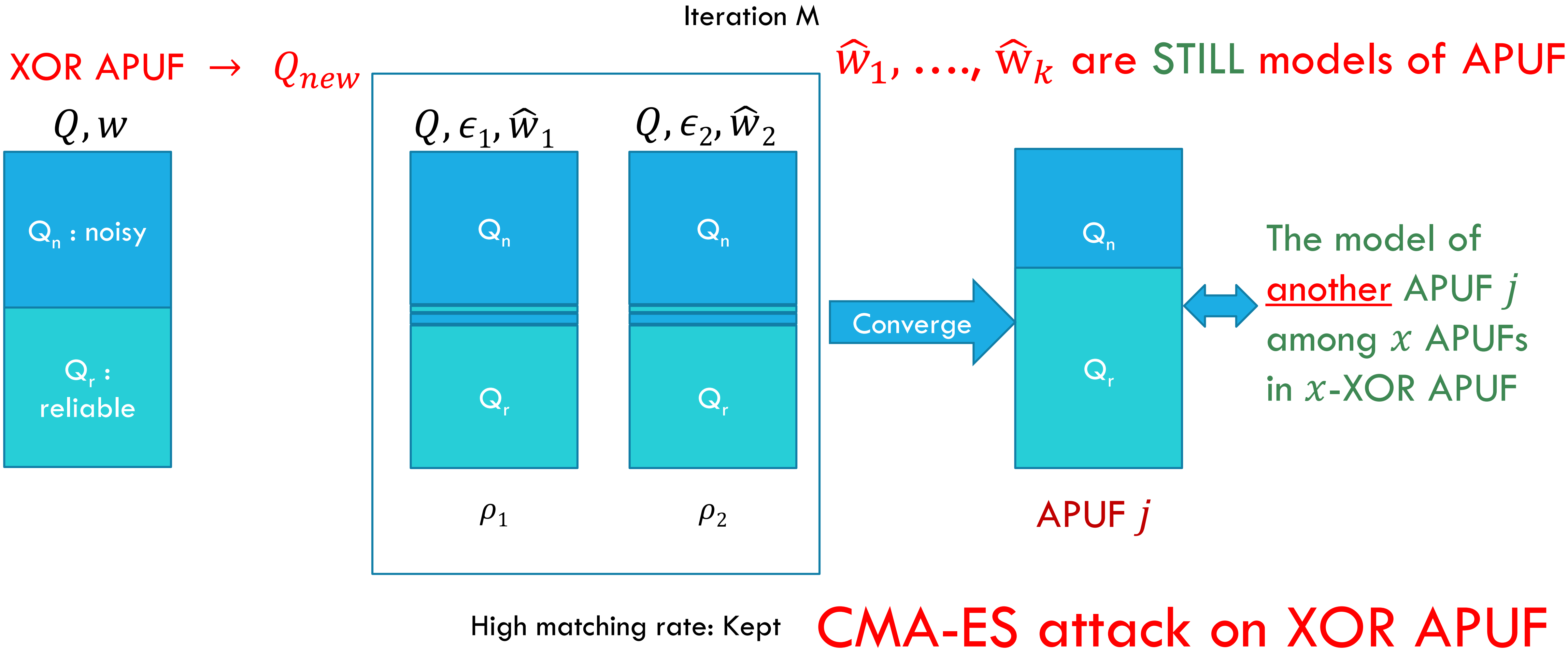
x-XOR PUF.

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



CMA-ES attack on XOR APUF

Reliability-based modeling attack on XOR APUFs using CMAES [2]



UCONN

Understanding Reliability based modeling attack on XOR PUF

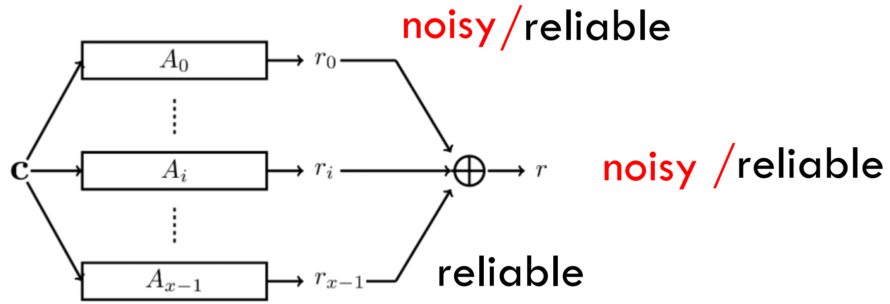
- Question 1: How does the attack on XOR PUF work?

- Question 2: How can we make the attack on XOR PUF fail?

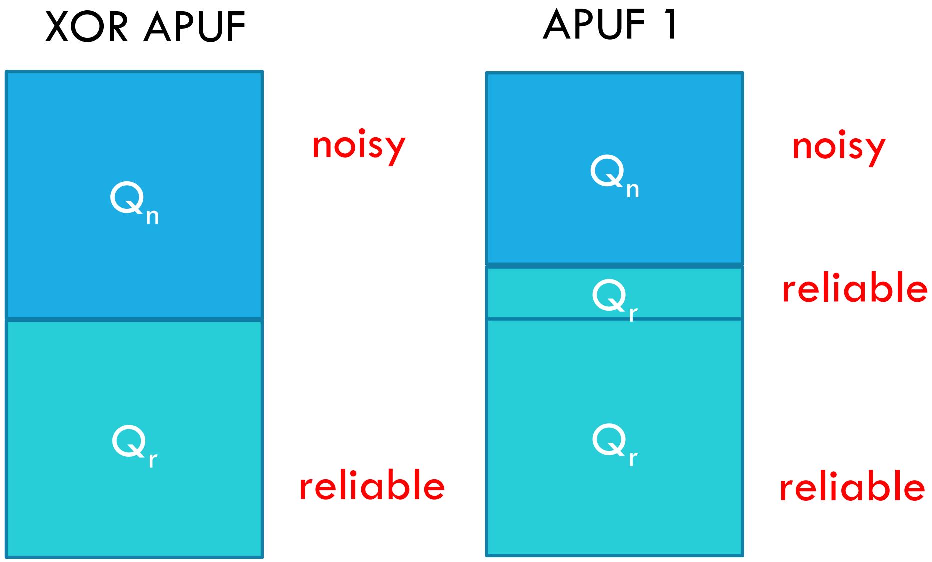
Question 1: How does the attack on XOR PUF work?

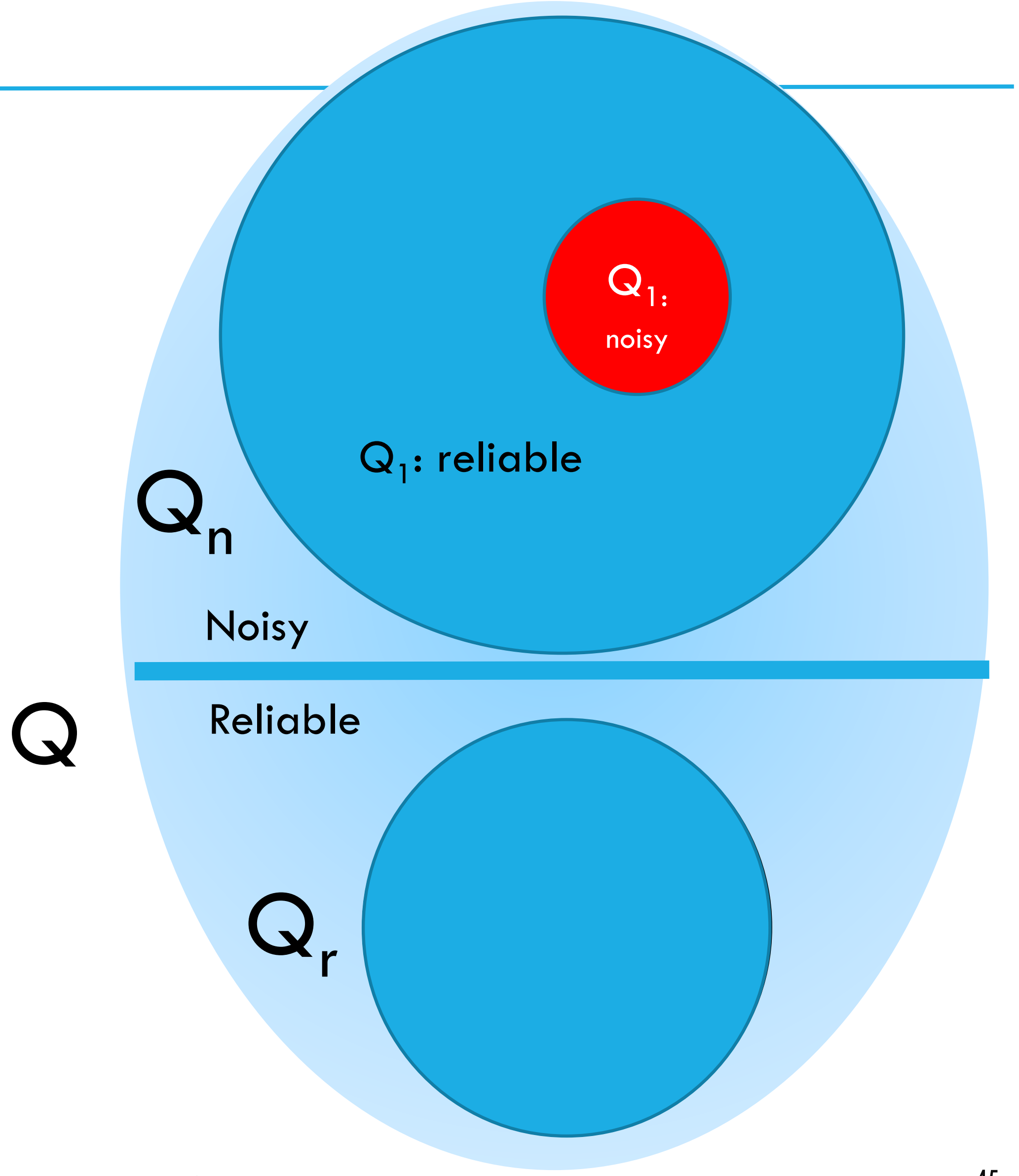
The noisy and reliable challenges in XOR PUF

UCONN



x-XOR PUF.

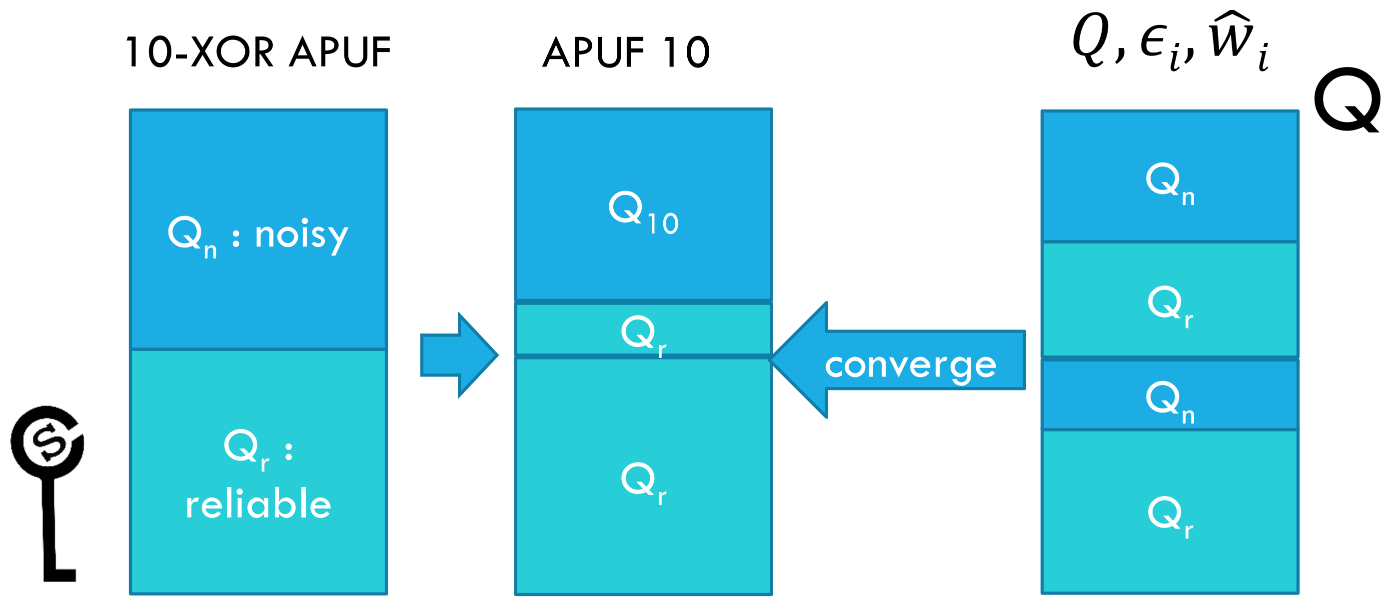


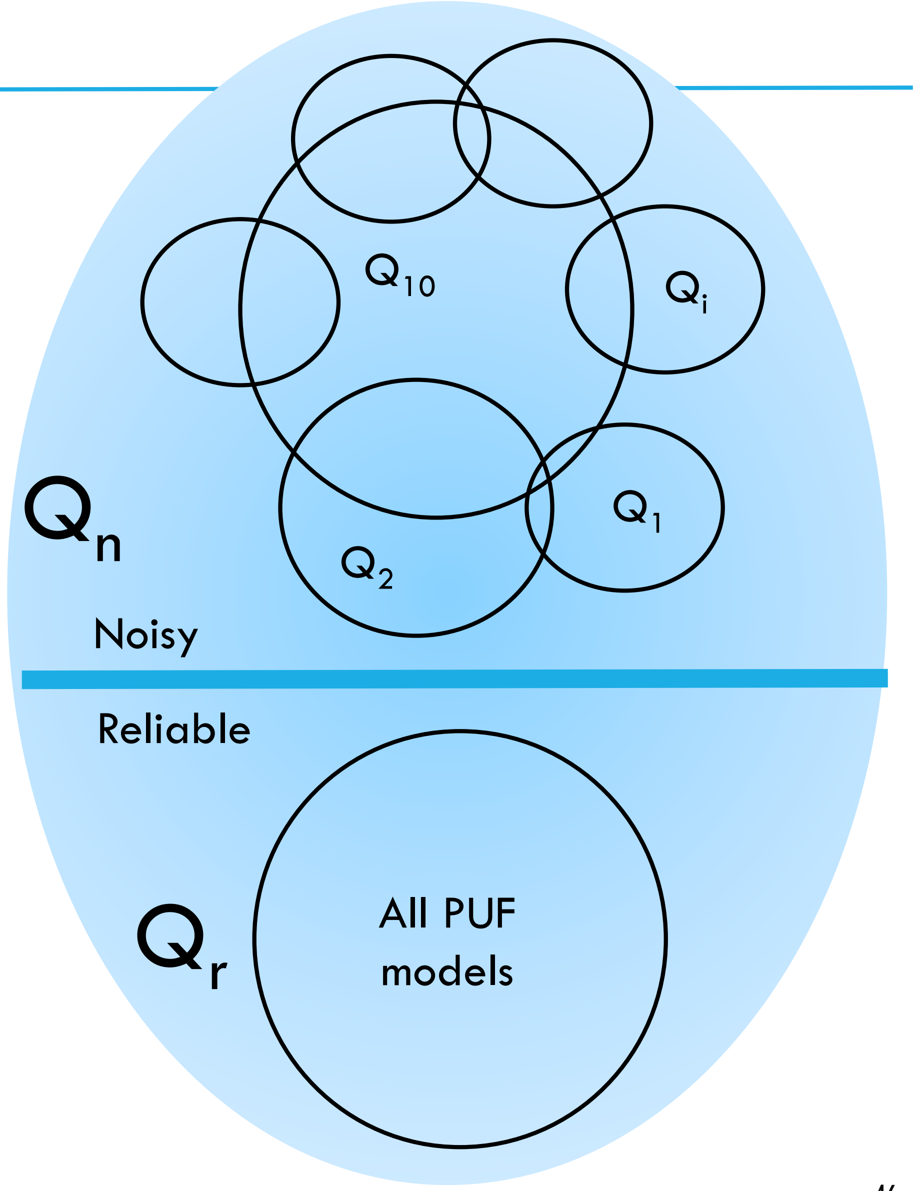


Challenge-Reliability Pairs Training Data

Key idea of the attack on XOR PUF [1]

- (1) All the models $\widehat{w}_i$ in CMA-ES are models of APUF



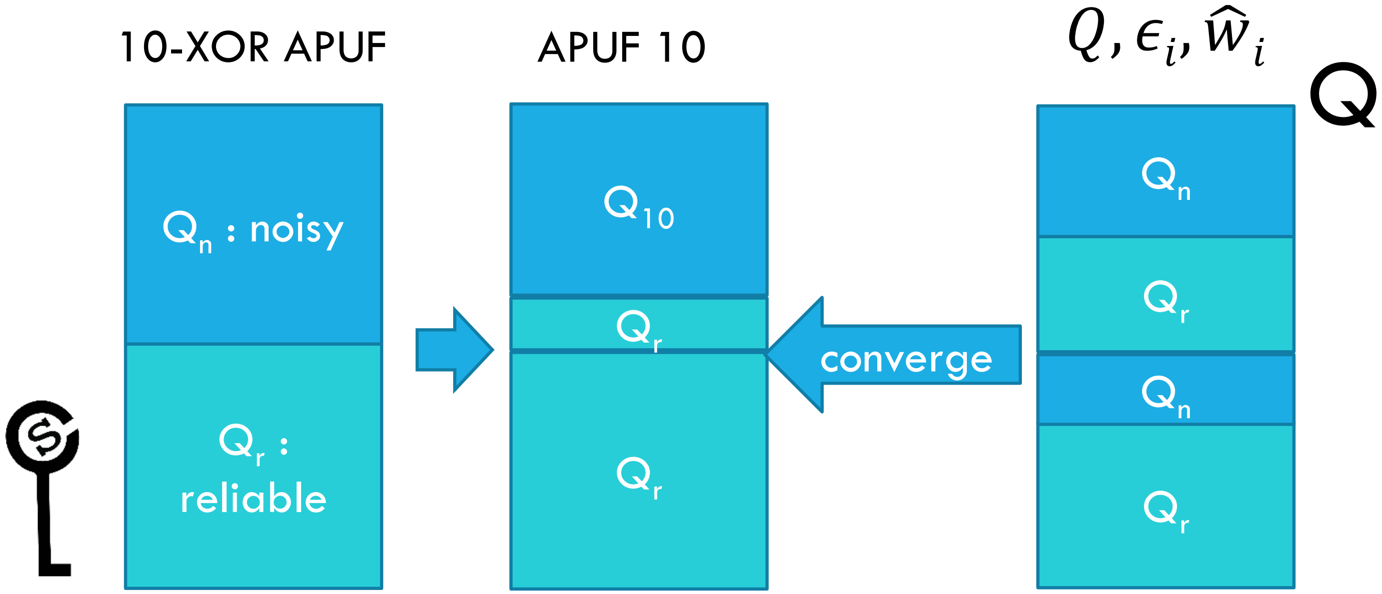


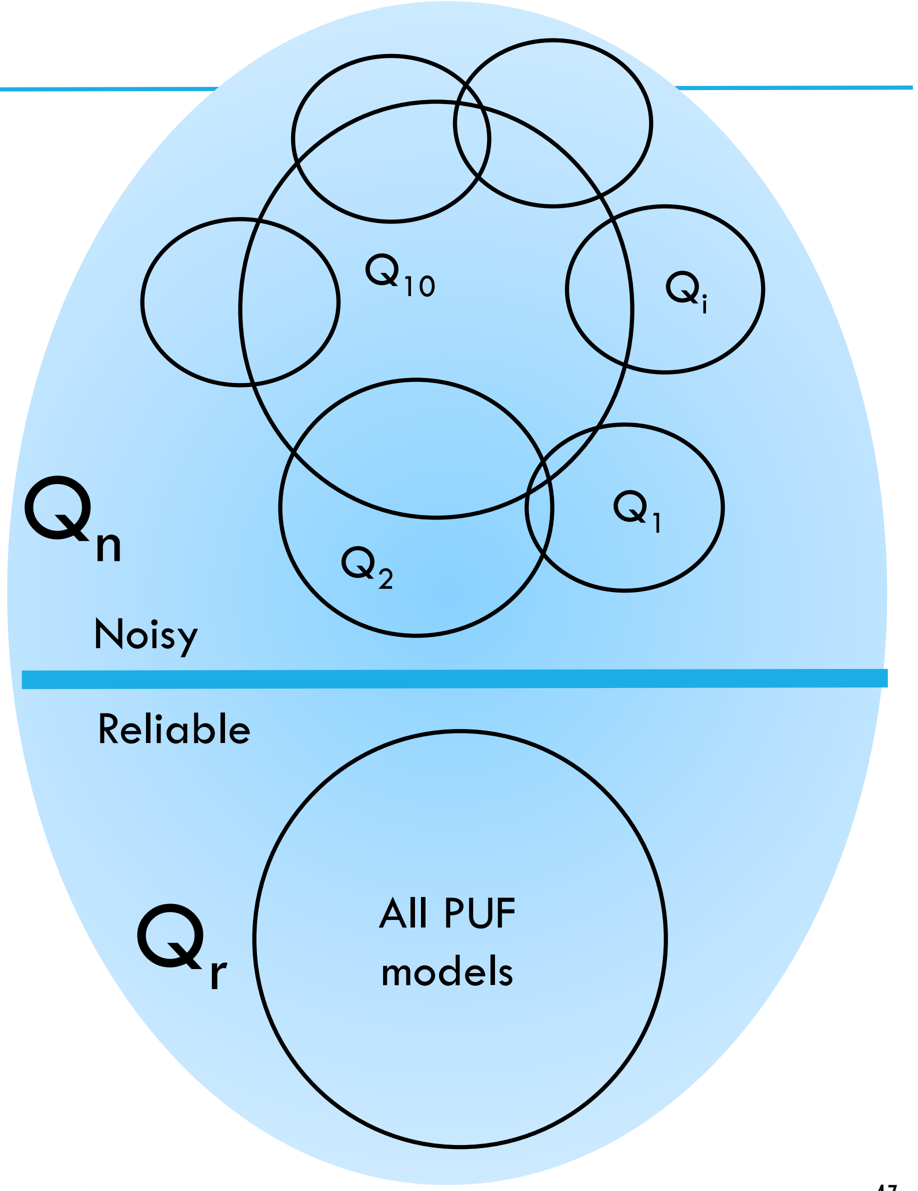
Challenge-Reliability Pairs Training Data

Key idea of the attack on XOR PUF [2]

UCONN

- (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

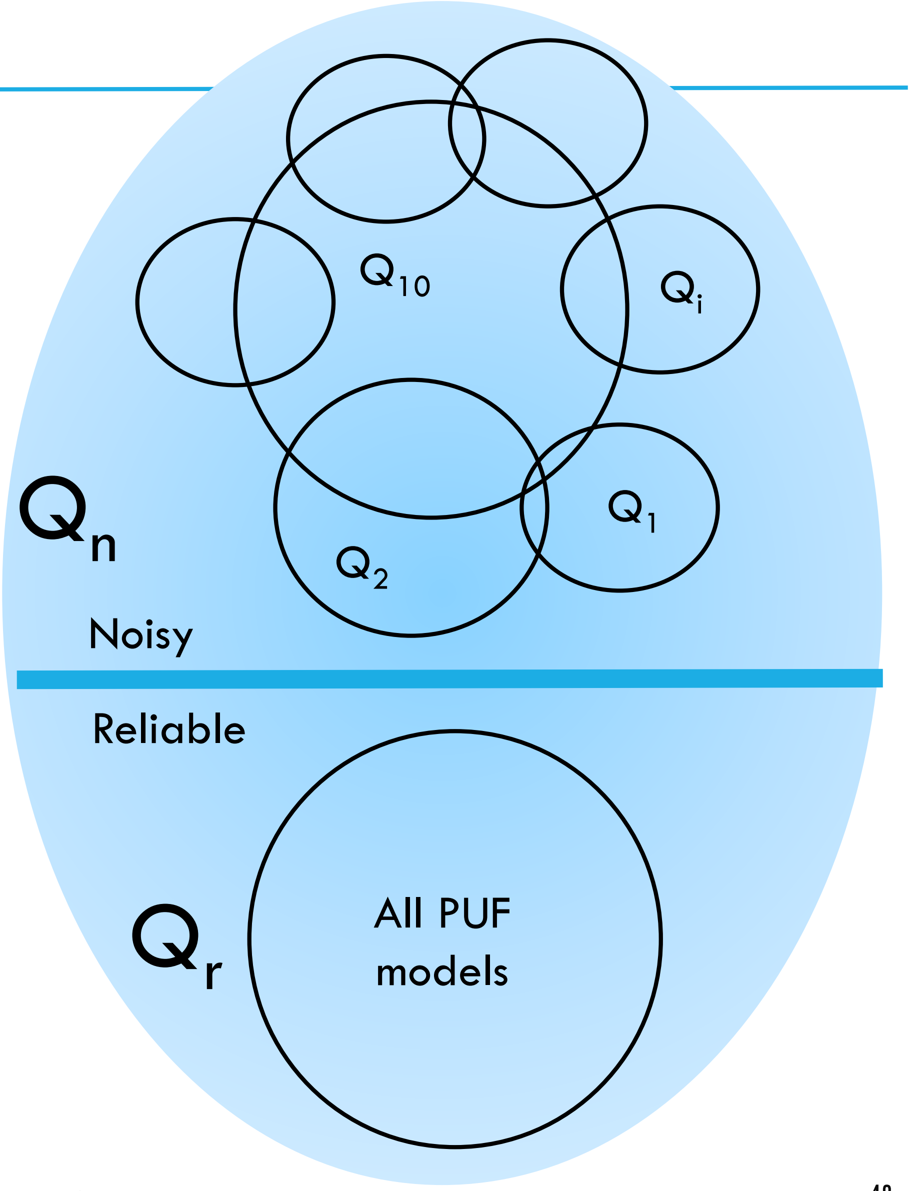




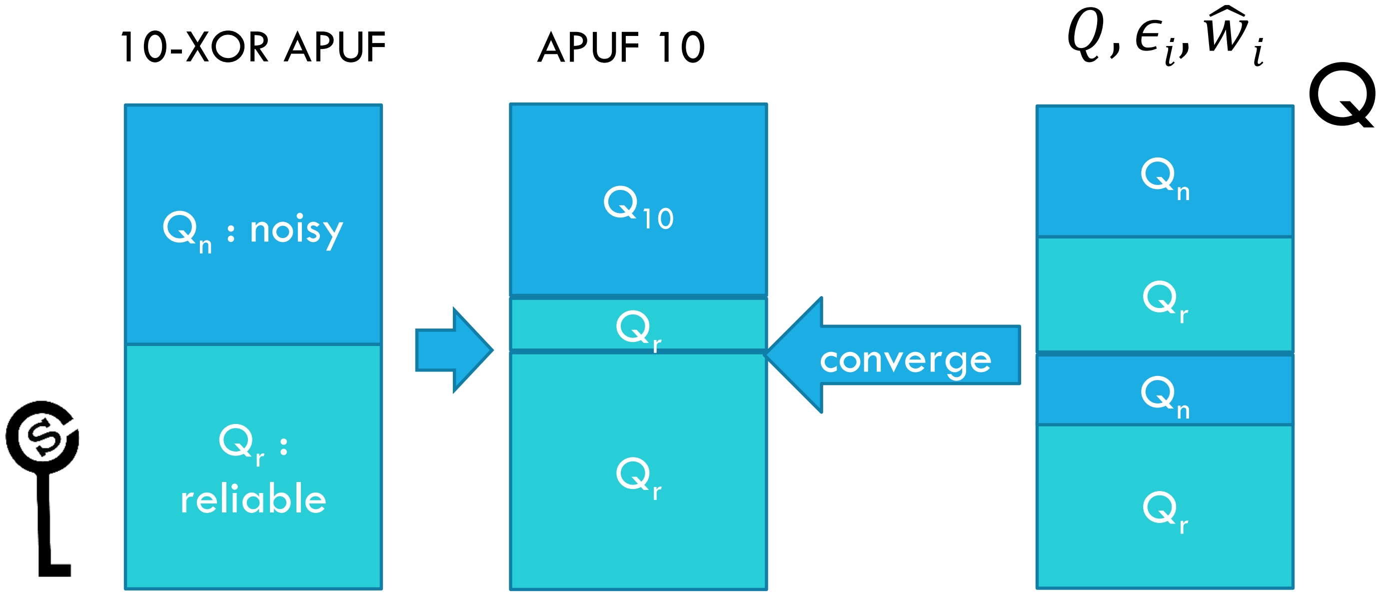
Challenge-Reliability Pairs Training Data

Key idea of the attack on XOR PUF [3]

- (1) All the models $\widehat{w}_i$ in CMA-ES are models of APUF
- (2) $\widehat{w}_i$ can only converge to an APUF instance
- (3) CMA ES maximizes the matching Q of $\widehat{w}_i$ and Q of XOR APUF

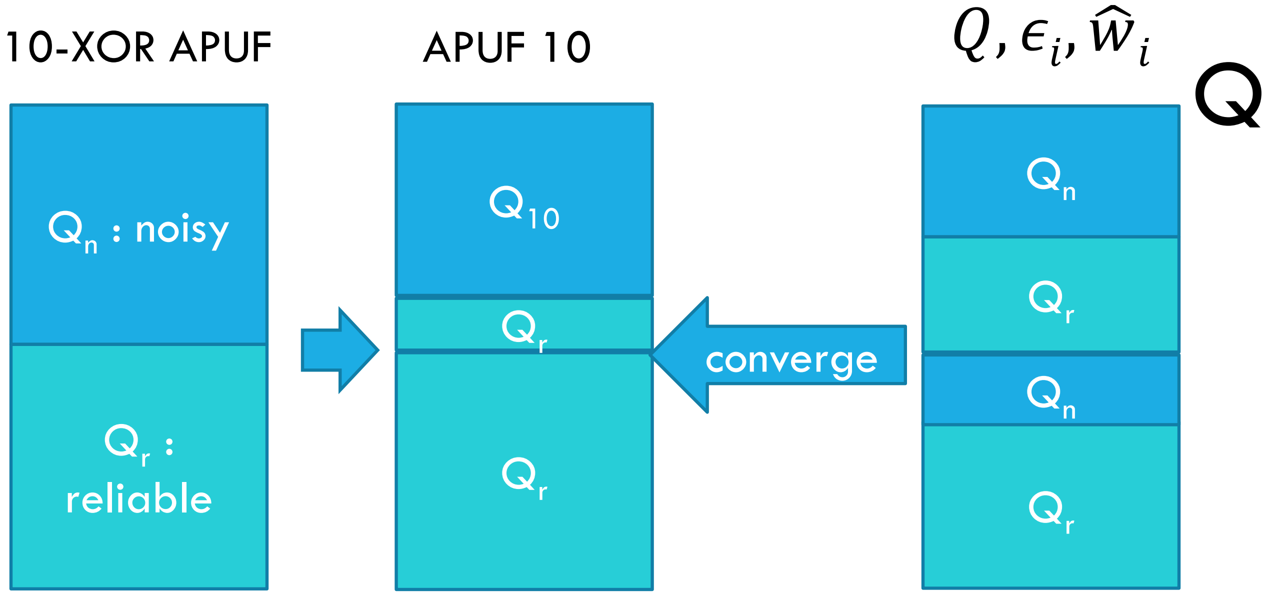


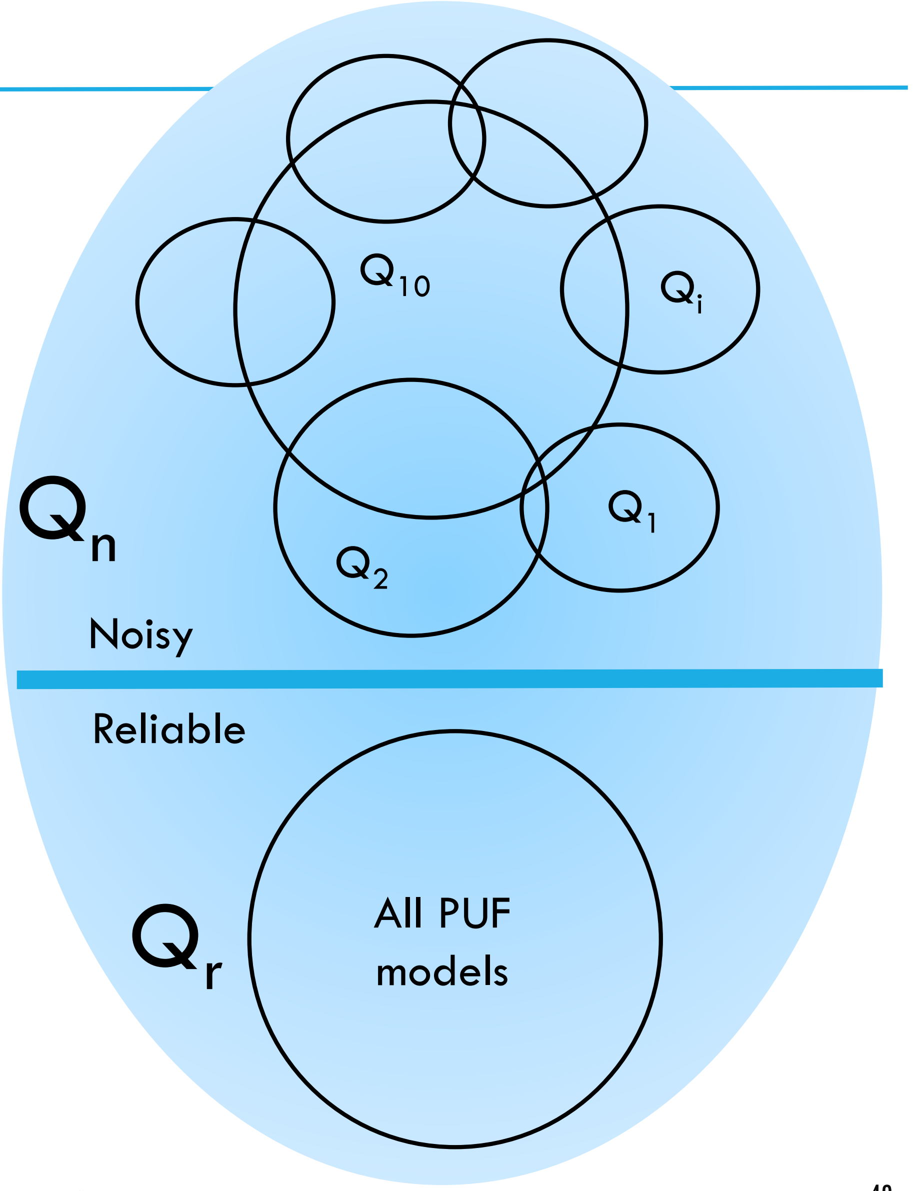
Challenge-Reliability Pairs Training Data



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.



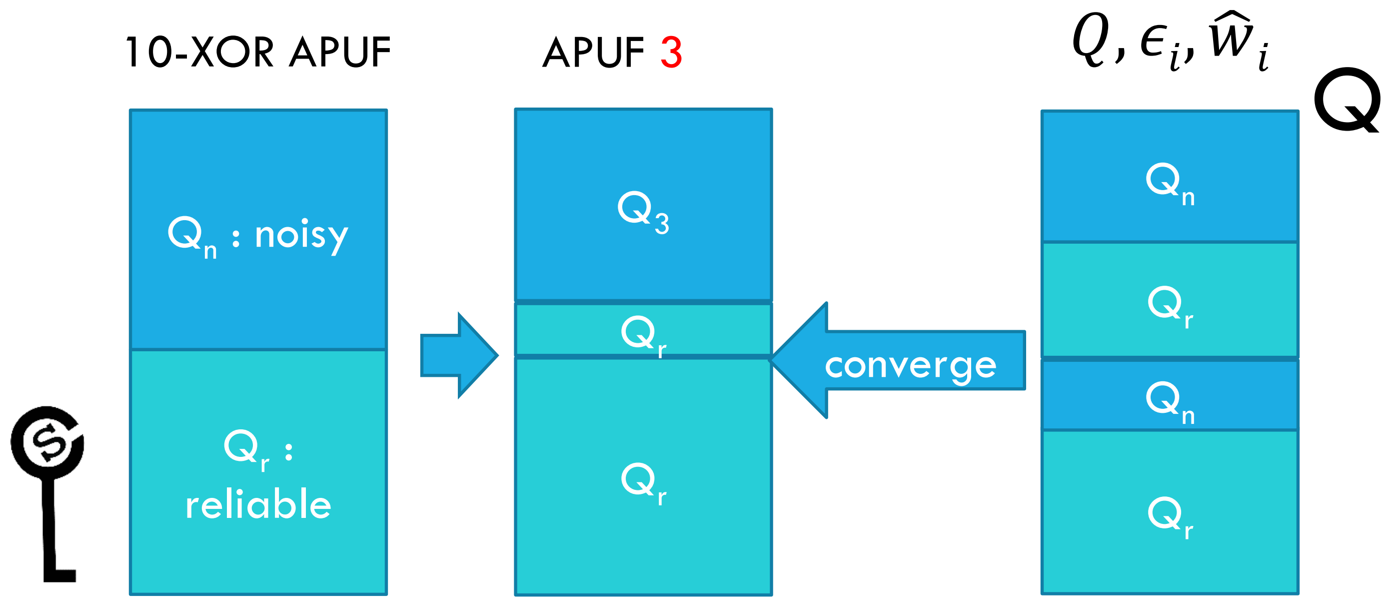


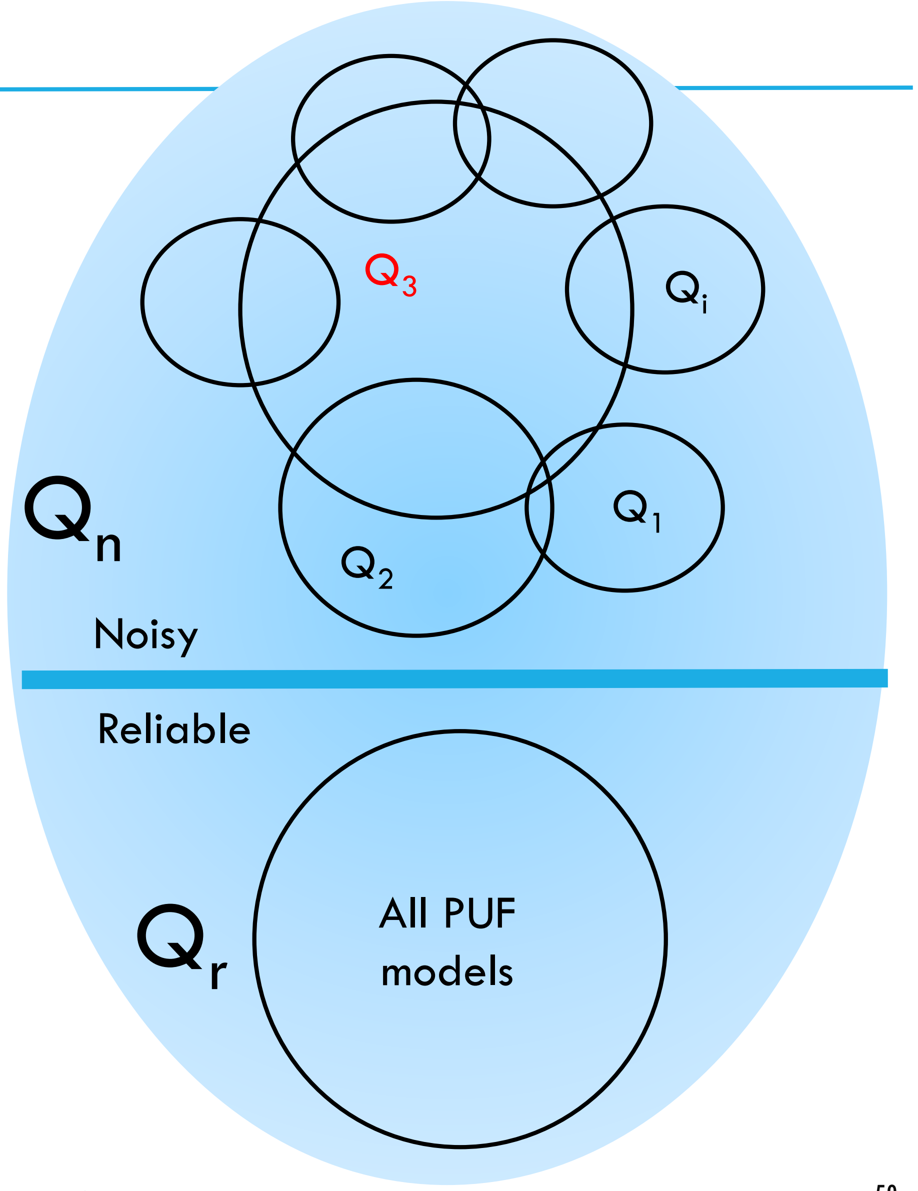
Challenge-Reliability Pairs Training Data

Key idea of the attack on XOR PUF [5]

UCONN

- Changing Q makes Q_3 largest



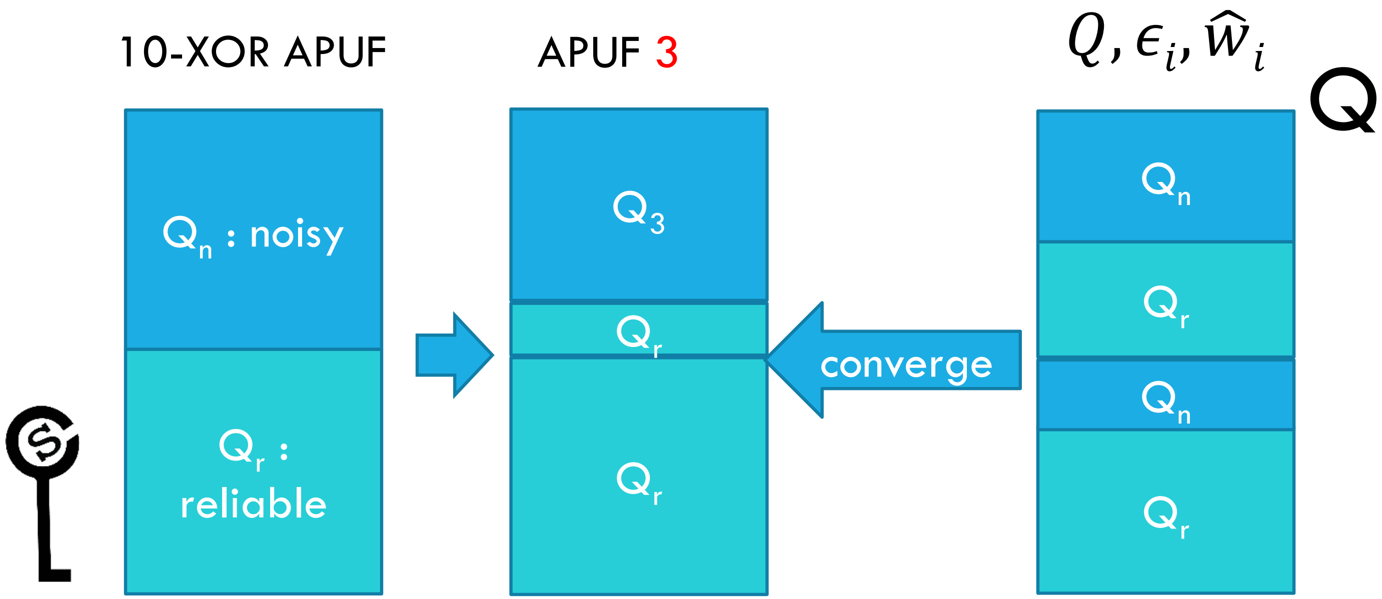


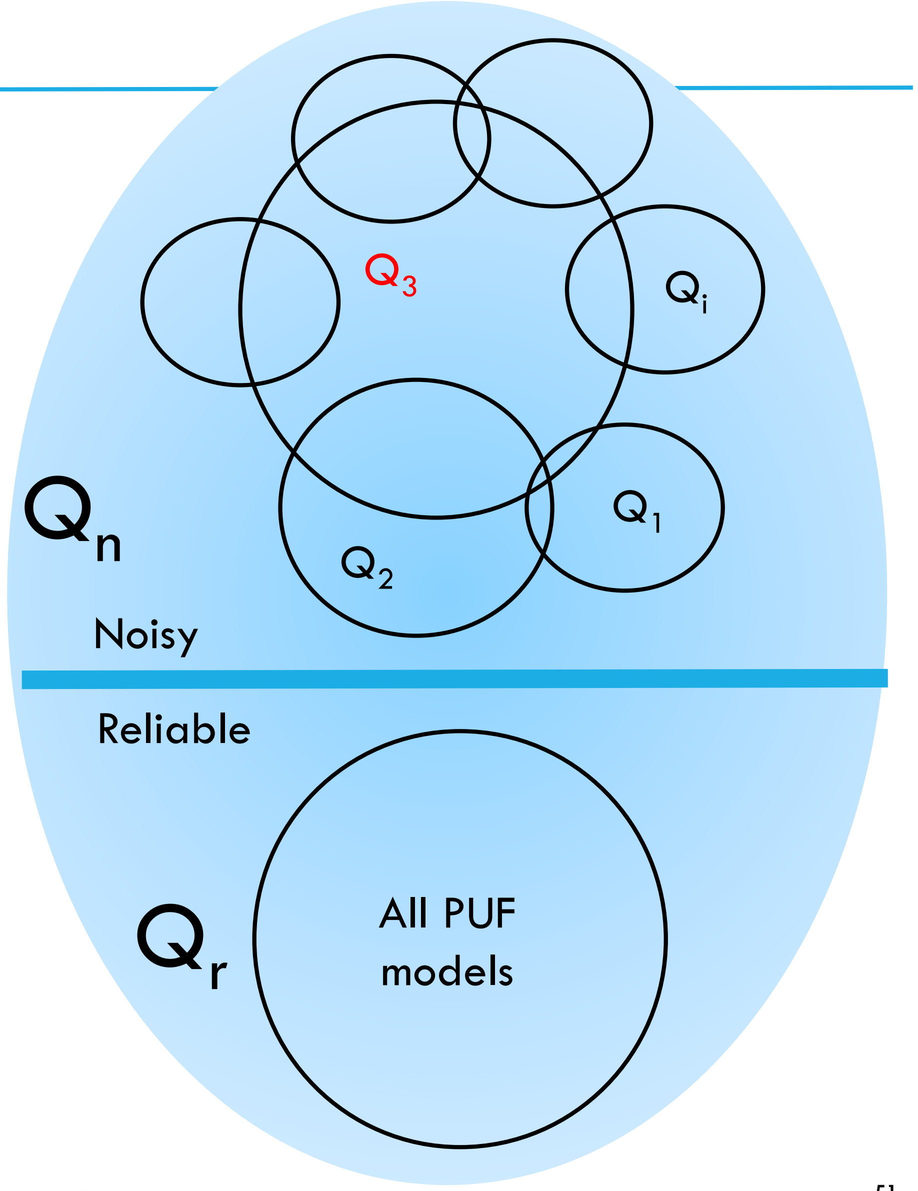
Challenge-Reliability Pairs Training Data

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



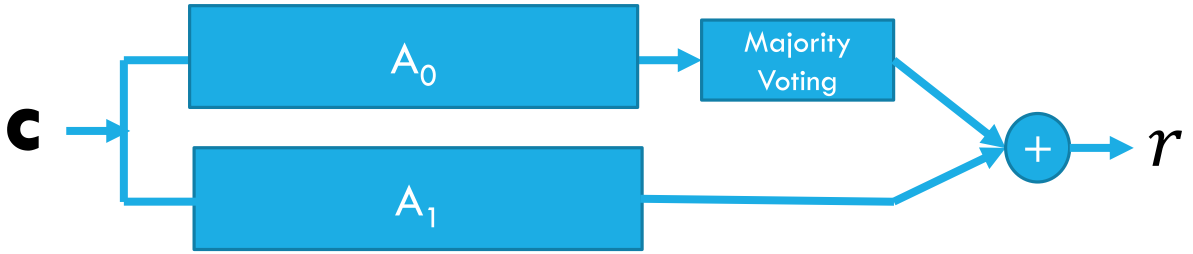


Challenge-Reliability Pairs Training Data

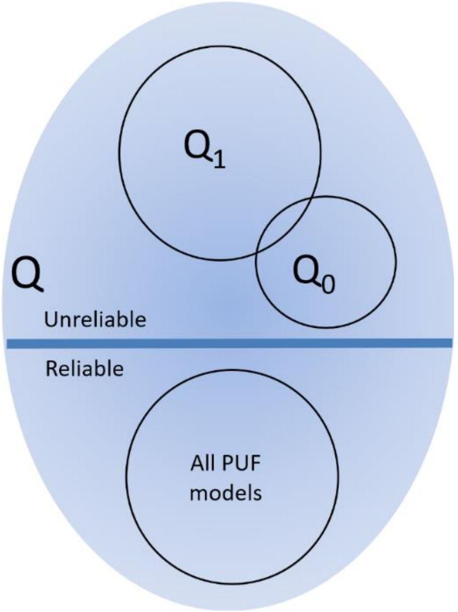
Question 2: How to make the attack on XOR PUF fail?

UCONN

Attack fails



2-XOR APUF with majority voting circuit at A_0



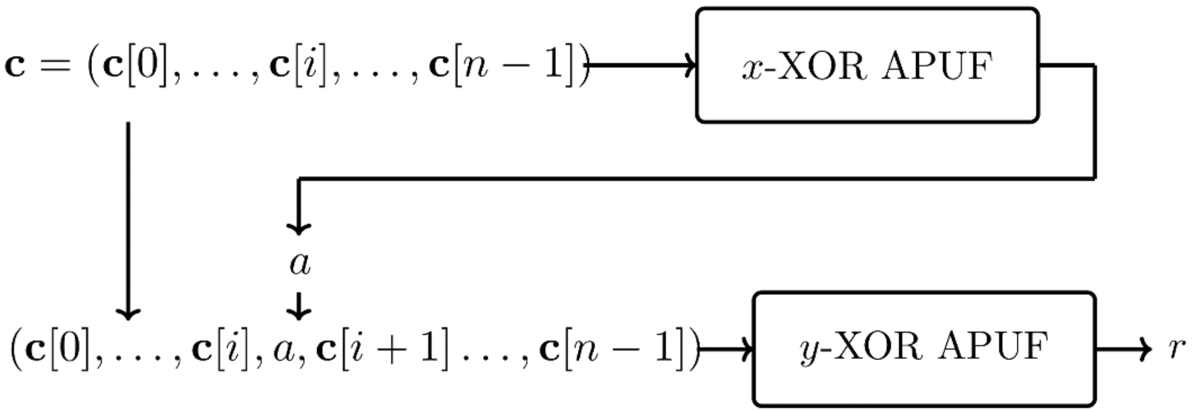
CMA ES never converges to APUF A_0 and always converges to APUF A_1 when majority voting mechanism in use.

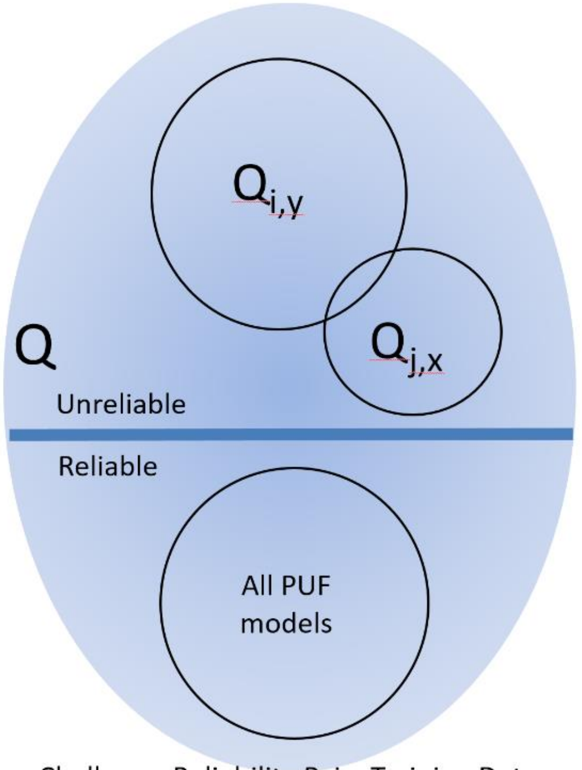
5. Interpose PUF (iPUF) – Reliability based modeling attack resistance

UCONN

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

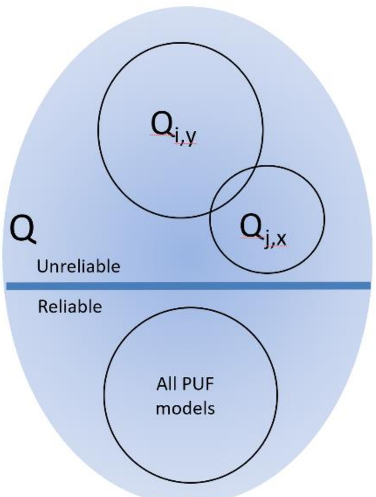




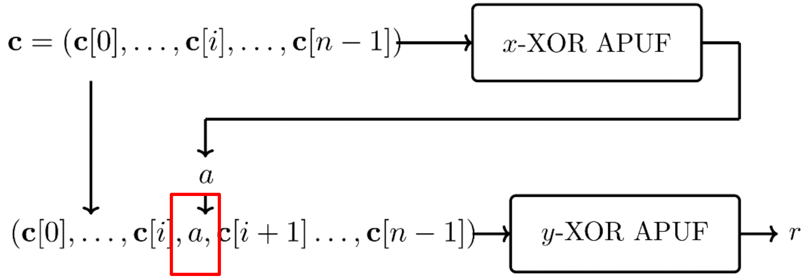
Challenge-Reliability Pairs Training Data

Security of iPUF wrt Reliability-based modeling attack [2]

- Reason 2: to attack APUFs at y-XOR APUF, the adversary needs to compute Δ. But compute Δ is infeasible because the output of x-XOR PUF (a) is not known.



Challenge-Reliability Pairs Training Data



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

Cannot compute $\Phi(c)$ or Δ

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

$|\Delta| \leq \epsilon_1 \rightarrow$ *challenge c is noisy* → Q_n : noisy

$|\Delta| > \epsilon_1 \rightarrow$ *challenge c is reliable* → Q_r : reliable

$$\Delta = \Delta(n-1) = \mathbf{w} \cdot \mathbf{\Phi}^{\mathsf{T}}$$

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

$$\mathbf{\Phi}[n] = 1$$

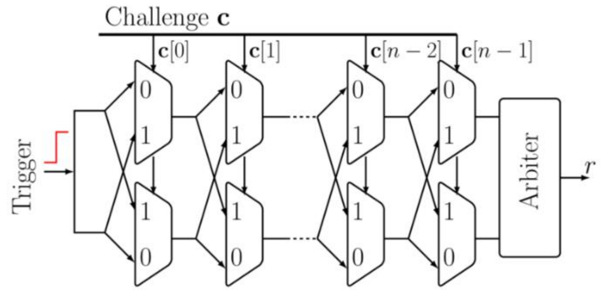
UCONN

Other Contributions

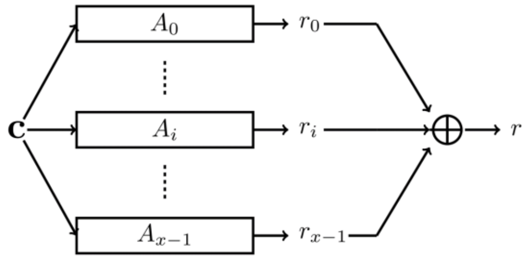
- Theoretical
  - Enhanced Reliability based Modeling Attacks on APUF and XOR APUFs
  - Proved Logistic Regression on XOR APUF is the best attack
  - Proved Logistic Regression on iPUF is not applicable
- Engineering
  - Implemented APUF, XOR, and iPUF on FPGA
  - Studied good and bad FPGA-implemented APUF based PUF
  - All source codes available online: https://github.com/scluconn/DA_PUF_Library/
- Detailed tutorial online: https://www.youtube.com/playlist?list=PLK5NNs4GceLQw7bOEHSdZOwHlmSF1zvSW

6. Conclusion

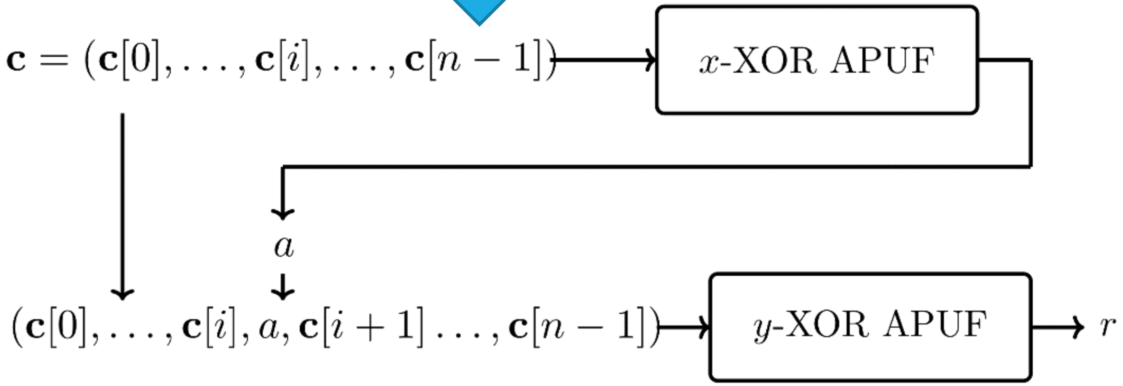
- We explain how the reliability-based modeling attack on XOR PUF works
- We propose a new lightweight PUF design (iPUF) which is secure against the state-of-the art of modelling attacks.



A n-stage classic Arbiter PUF with challenge $\mathbf{c} \in \{0,1\}^n$.



x-XOR PUF.



UCONN

Literature

1. https://slideplayer.com/slide/3927633/
2. Cryptanalysis of electrical PUFs via machine learning algorithms – Master Thesis of Jan Solter
3. The Gap Between Promise and Reality: On the Insecurity of XOR Arbiter PUFs CHES, September 16 th , 2015, Georg T. Becker
4. https://en.wikipedia.org/wiki/CMA-ES

Thank you for your attention!

and any questions?