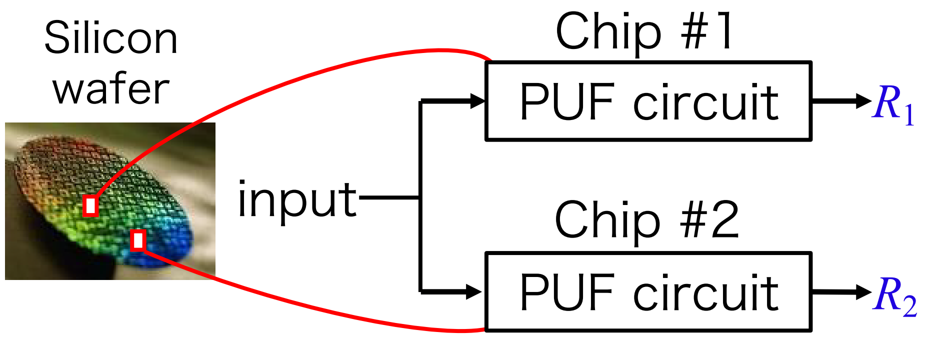
# Rejection Sampling Schemes for Extracting Uniform Distribution from Biased PUFs

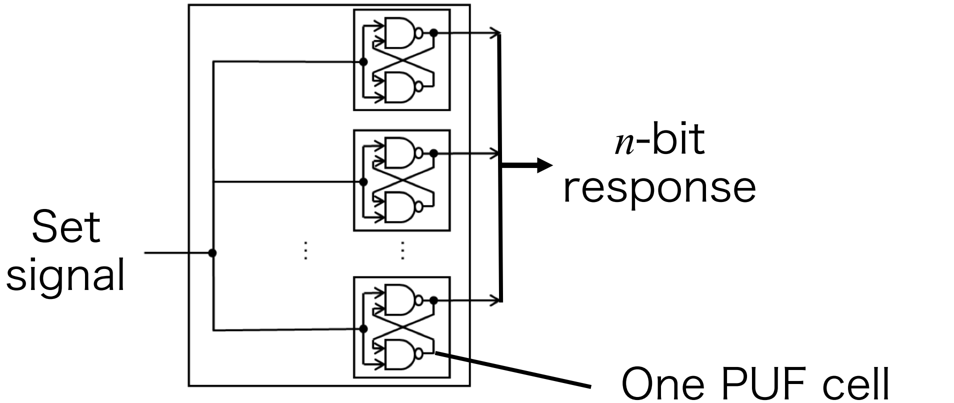
Rei Ueno, Kohei Kazumori, and Naofumi Homma
Tohoku University

# Background

- Physically unclonable functions (PUFs) play essential role for constructing secure and trustable systems
  - Generate hardware-intrinsic random number like fingerprint
  - Exploit process variations for physical unclonability and tamper evidence
- Major applications of PUF
  - Entity authentication (Strong PUF)
  - Cryptographic key generation (Weak PUF)



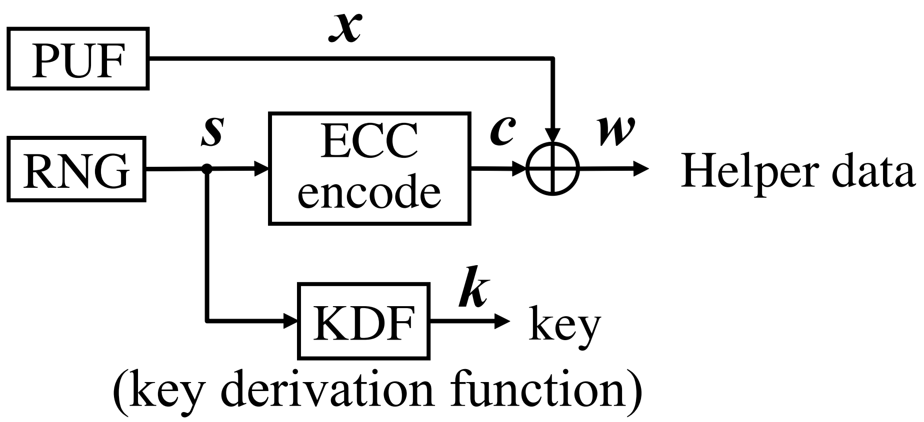

Even for <u>same input</u> and <u>same circuit construction</u>, PUF responses vary due to process variation (i.e., $R_1 \neq R_2 \neq \cdots$)



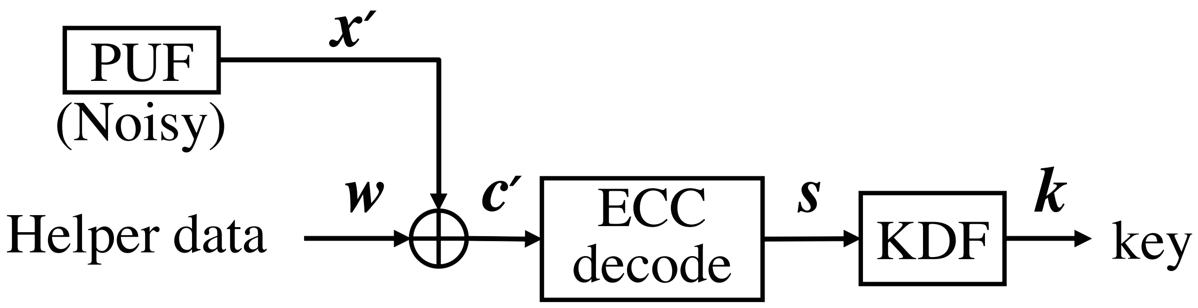

Example of PUF (Latch PUF)

# PUF-based key generation

- Fuzzy extractor (FE) is commonly used for reconstructing enrolled key from noisy PUF response




Enrollment

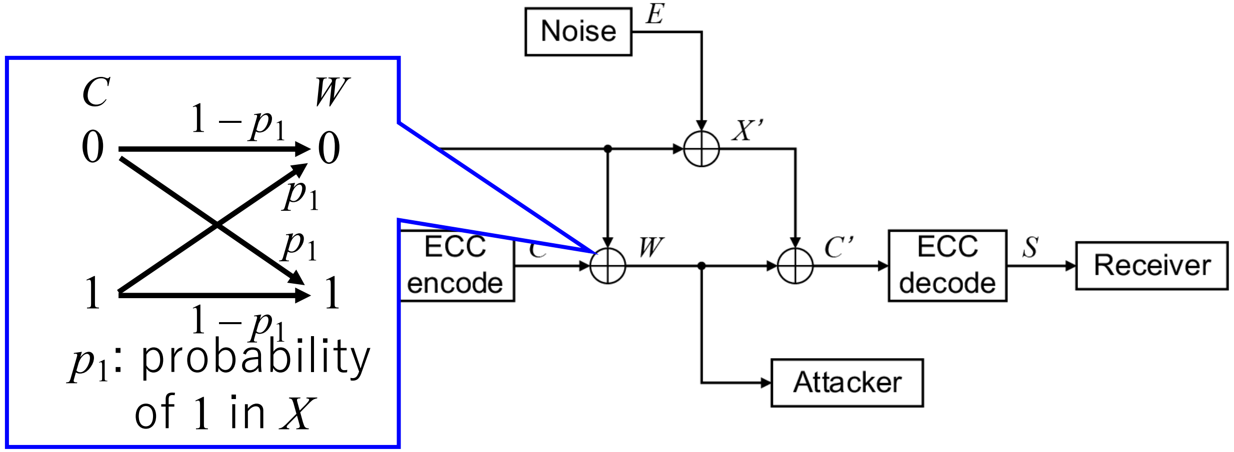



Reconstruction

- Helper data is stored in common nonvolatile memory (NVM)
  - NVM is usually non-tamper resistant, and helper data is considered public
  - We should consider conditional entropy for key generation
    - A $\sigma$-bit key generation is realized only if $\mathbb{H}(S|W) > \sigma$

# Problem of PUF bias: Entropy leakage

- If PUF response is unbiased, $\mathbb{H}(S|W) = \mathbb{H}(S)$ (i.e., seed length)
- But $\mathbb{H}(S|W)$ significantly decreases with PUF bias increase
  - $\mathbb{H}(S|W) = \mathbb{H}(S) - \boxed{\mathbb{I}(S;W)}$ Entropy leakage
  - If PUF is biased, random seed should be set longer than $\sigma$ such that $\mathbb{H}(S|W) > \sigma$
    - But required PUF size rapidly grows with PUF bias, especially when $p_1 > 0.58$



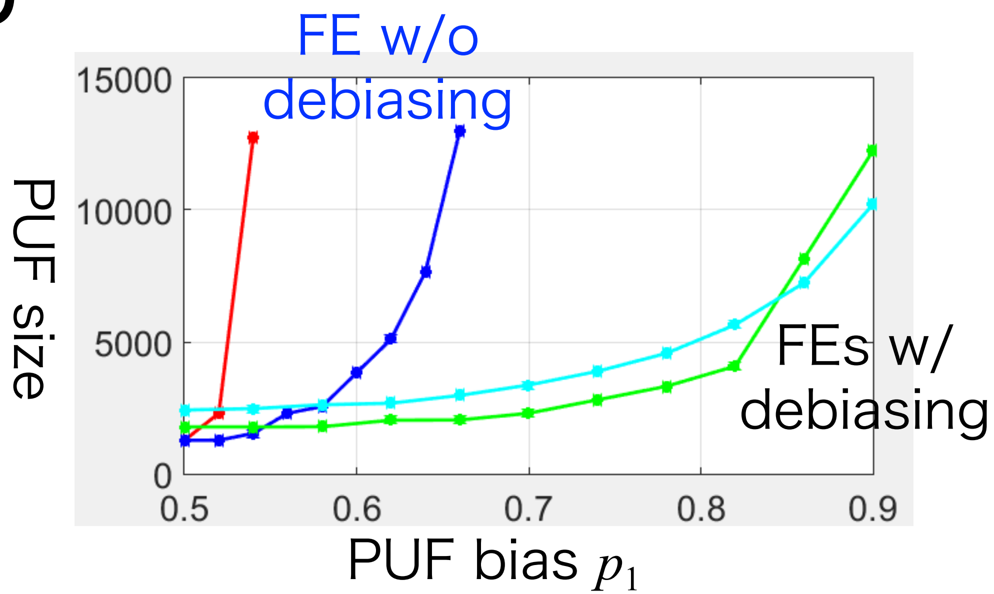

Channel diagram of FE [HO17]

PUF size required for reliable 128-bit key generation (Values are from [DGV+16])

| **PUF bias $p_1$** | **0.54** | **0.58** | **0.62** | **0.66** |
|---|---|---|---|---|
| Bit-error rate | 0.100 | 0.098 | 0.096 | 0.092 |
| PUF size | 1,530 | 2,550 | 5,100 | 13,005 |

# Debiasing

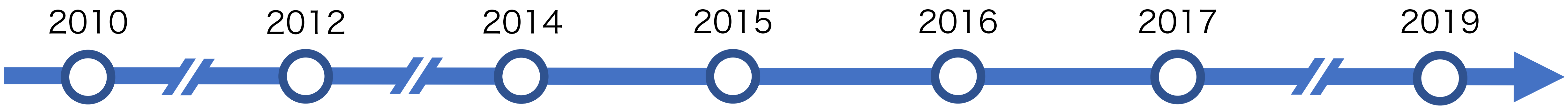
Figure is based on graph in presentation slide of [DGV+16]




- Extract unbiased bit string from biased PUF response
  - Realize secure key generation even from PUFs with nonnegligible biases
  - Efficiency has been evaluated through PUF size required for reliable 128-bit key gen.
- Example of debiasing: von Neumann corrector (VNC)
  - Values of 1 and 0 are extracted with an identical probability of $p_1 p_0$
  - Debiasing data $\boldsymbol{d}$ is used for reproducing $\boldsymbol{z}$ at reconstruction

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| PUF response $\boldsymbol{x}$: | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| | ≠ | | ≠ | | = | | = | | ≠ | |
| Debiased data $\boldsymbol{z}$: | 1 | | 0 | | | | | | 1 | |
| Debiasing data $\boldsymbol{d}$: | 1 | | 1 | | 0 | | 0 | | 1 | |

# Conventional debiasing-based FEs




**2010**

[YD10]
Index-based syndrome (IBS)

[LSHT10]
First von Neumann corrector (VNC)-based debiasing

**2012**

[HMSS12]
Generalized IBS

**2014**

[KLRW14]
Report on entropy loss in PUF-based key generation

**2015**

[MLSW15]
VNC-based FEs, first explicit solution for secure key generation from biased PUFs

**2016**

[DGV+16]
Tight bounds of min-entropy loss, motivation for debiasing

**2017**

[SUHA17]
Ternary VNC-based FEs

[S17]
Trivial debiasing

[AWSO17]
Maskless debiasing (MD)

[HO17]
Coset coding (CC)-based FE, FE is modeled as wire-tap channel

**2019**

[USH19]
Biased masking (BM)-based FE

[KW19]
Selection and balancing schemes for SRAM PUF

- Various debiasing-based FEs have been developed for improving efficiency
  - Efficient FE reduces PUF and NVM sizes
  - How far can we go?

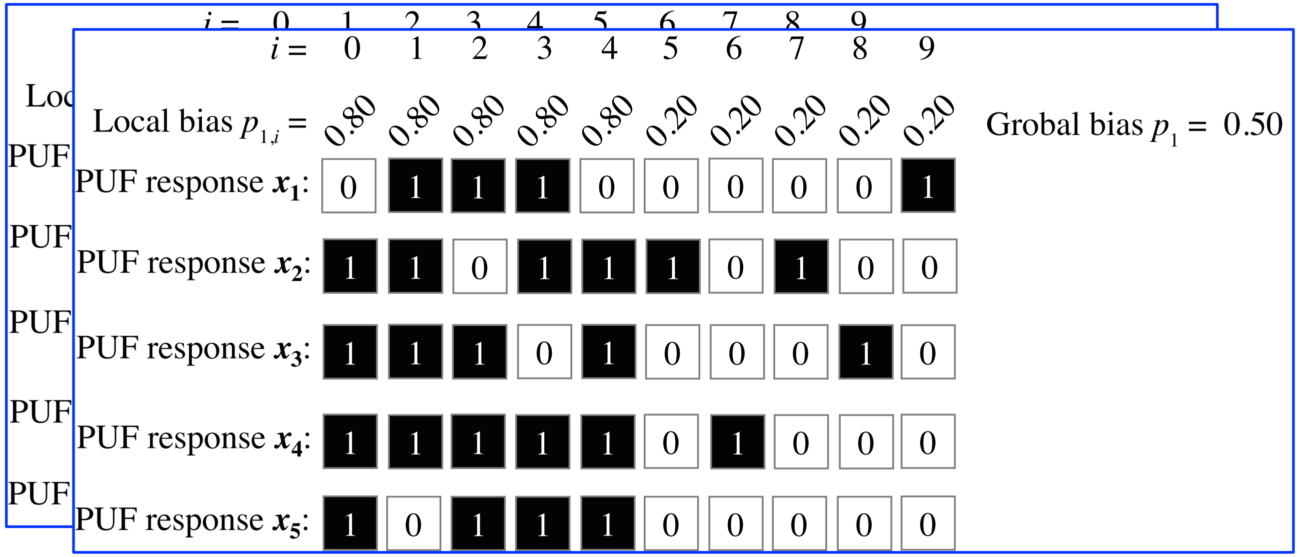
# This work

- Acceptance-or-Rejection (AR)-based FE: New debiasing scheme based on rejection sampling and FE construction
  - Extract uniform distribution with highest efficiency among conventional FEs
  - Implemented with solely an RNG at enrollment, and no critical additional operation is required at reconstruction performed on client device
  - First FE which can tolerate local biases depending on cell addresses (for example, found in some SRAM PUFs)
  - Extended to ternary PUF response for improved efficiency (see our paper)

- Performance of proposed FE is evaluated through simulation of 128-bit key generation in comparison with conventional FEs
  - AR-based FE achieves smallest PUF and/or NVM sizes (i.e., hardware cost) for various PUFs
  - At most 55% and 72% smaller PUF and/or NVM sizes than counterparts

# Bias models

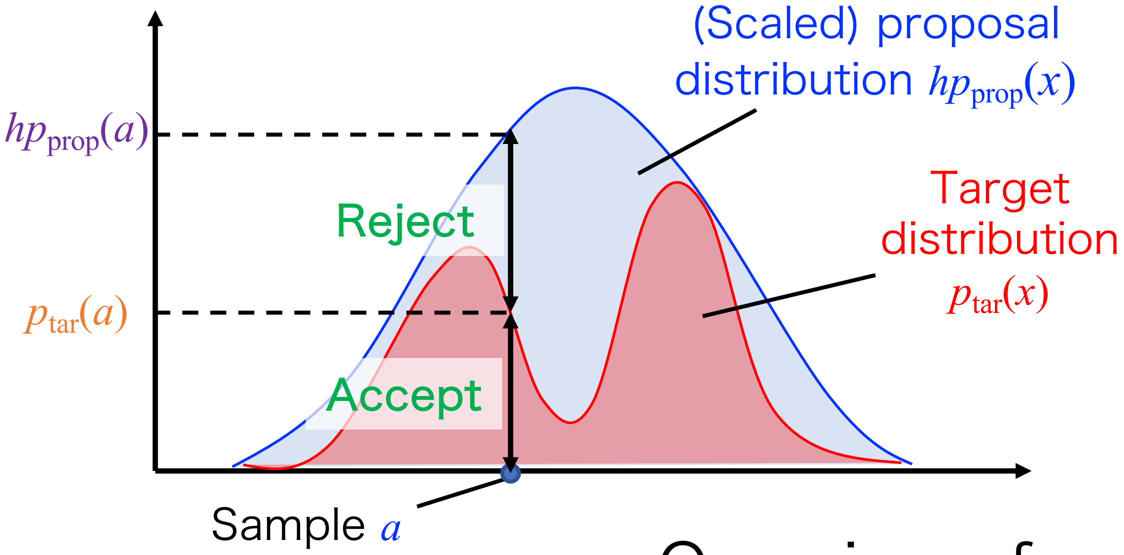
- Global bias model
  - All bits in PUF response have an identical bias of $p_1$ (with corresponding $p_0$)
  - All conventional debiasing scheme employed global bias model
- Cell-wise bias model (or local bias model)
  - Each bit has unique bias depending on cell address $i$
  - Expected value of biases are considered equal to global bias (i.e., $\mathbb{E}_i[p_{1,i}] = p_1$)




Typical example of cell-wise-based PUF

# Rejection sampling

- Method for deriving target distribution from proposal one
  - Target distribution: Distribution which is needed, but not directly available
  - Proposal distribution: Easily available distribution




Step (1): Obtain sample $a$ from $p_{\mathrm{prop}}(x)$

Step (2): Draw random number $b$ from $[0, p_{\mathrm{prop}}(a)]$

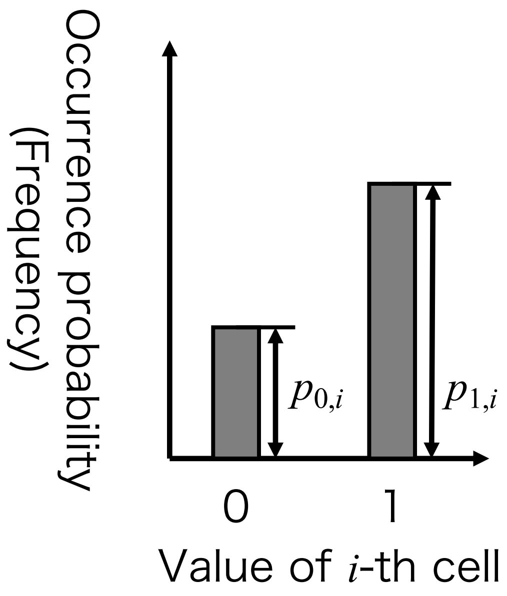
Step (3): Accept the sample if $b < p_{\mathrm{tar}}(a)$; otherwise, reject it

Overview of rejection sampling

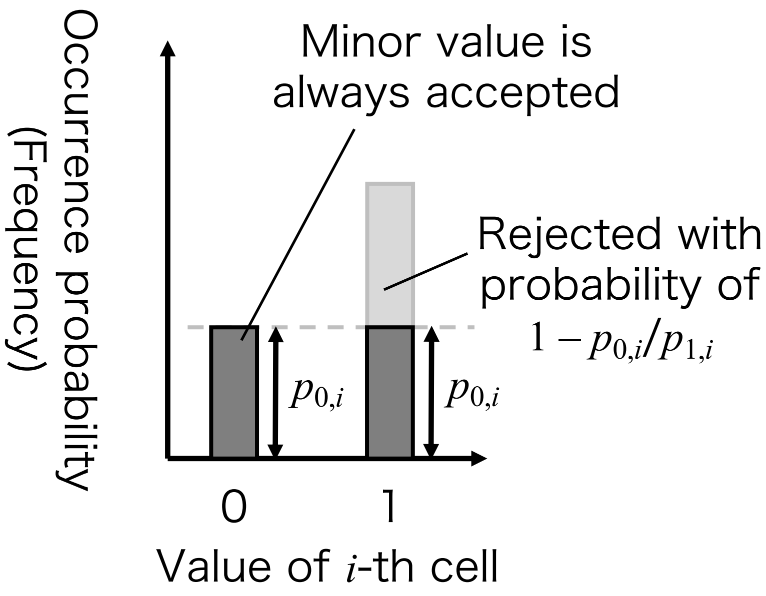
- Application to PUF debiasing
  - Target distribution: Uniform distribution
  - Proposal distribution: PUF response (i.e., $p_{1,i}$-biased Bernoulli distribution)

# Extraction of uniform distribution from biased PUFs

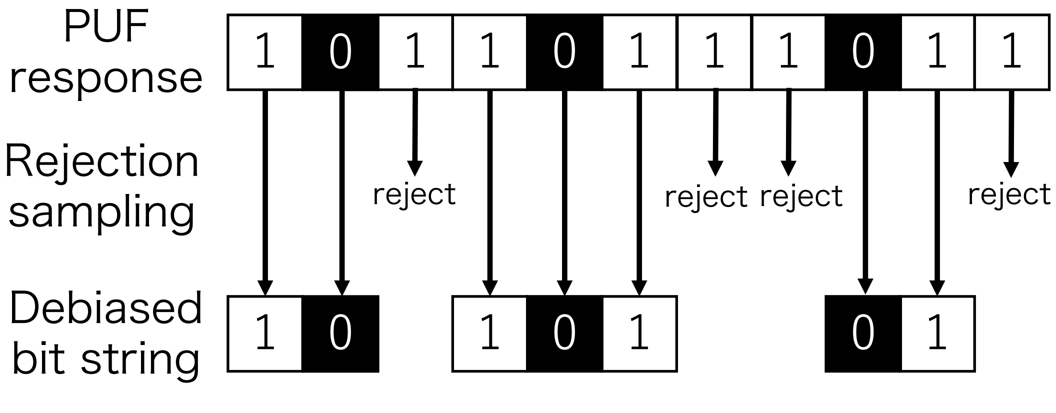
- Key idea: Bit-wise rejection sampling
  - Rejection sampling is applied to $i$-th cell with biases $p_{1,i}$, $p_{0,i}$ for all $i$
  - Expected length of debiased bit string is longer than conventional schemes




Biased PUF response
as Bernoulli distribution
(i.e., proposal distribution)



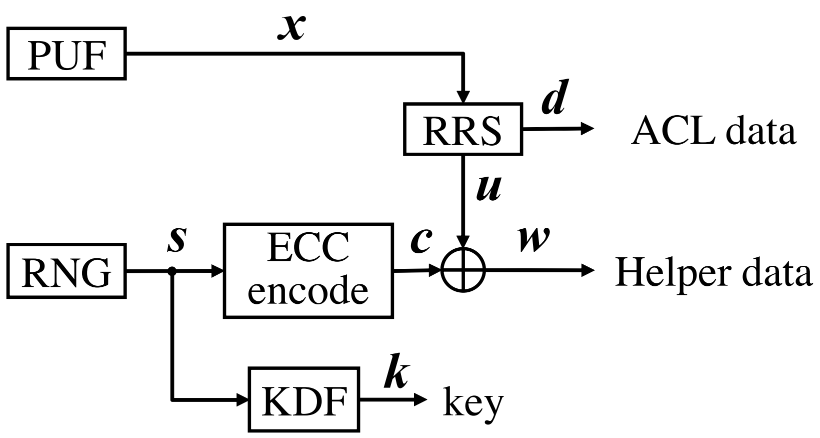

Distribution after
rejection sampling
(i.e., target distribution)



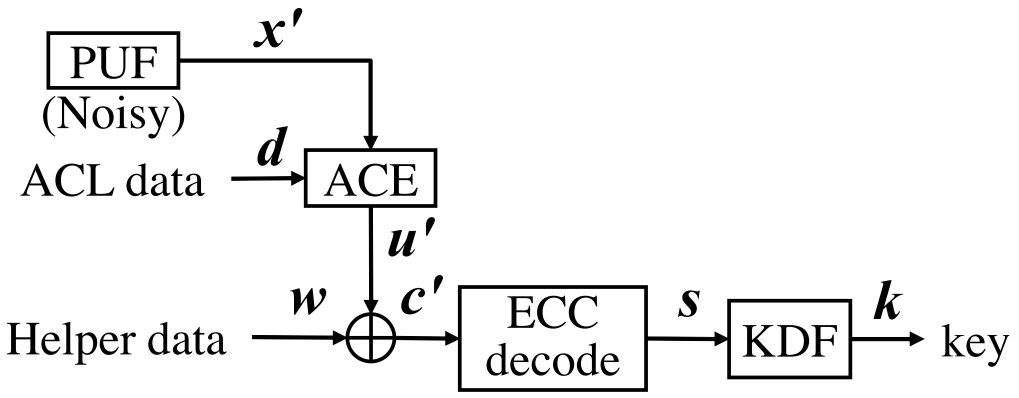

Example of debiasing ($p_{1,i} = 0.70$ for all $i$):
"0" cells are always accepted and
"1" cells are rejected (i.e., discarded) with
probability of $1 - p_{0,i}/p_{1,i} = 0.57$

# Proposed scheme: AR-based FE

- Reproducible rejection sampling (RRS) and accepted cell extraction (ACE) operations are applied to PUF response




Enrollment of AR-based FE

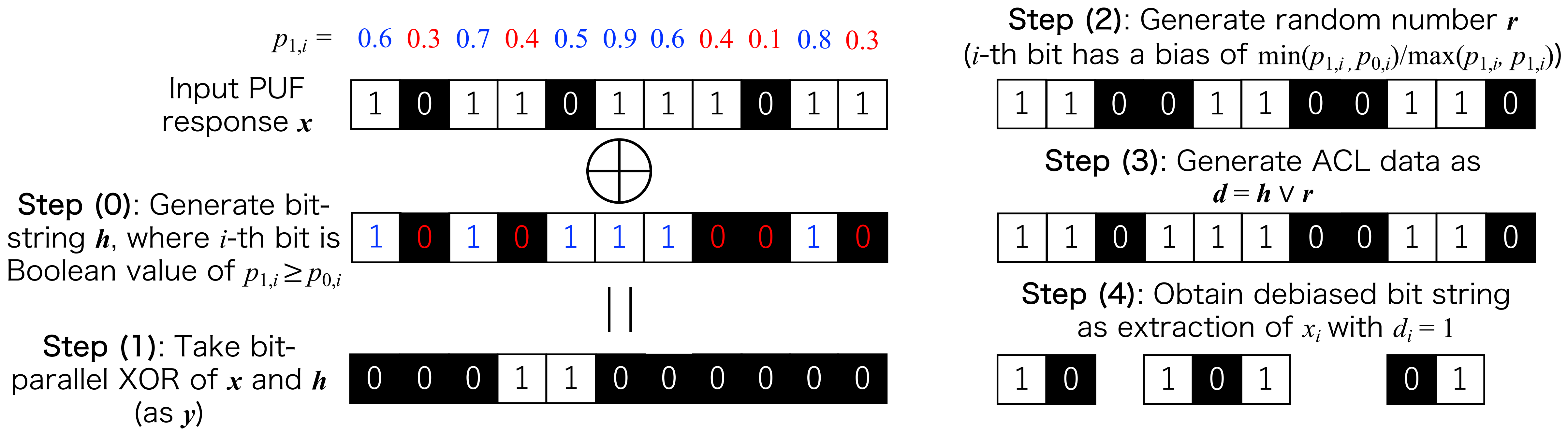



Reconstruction of AR-based FE

- RRS operation generates debiased bit string and accepted cell location (ACL) data $\boldsymbol{d}$
  - Naïve rejection sampling is not reproducible
  - ACL data enables us to reproduce debiased bits at ACL at reconstruction
  - We proved there is no entropy leakage from pair of helper and ACL data

# RRS and ACE operations—Implementation

- RRS operation performs rejection sampling with reproducibility
  - First generate ACL data $\boldsymbol{d}$, and then extract debiased bit string
  - Implemented using an RNG and bit-parallel operations in enrollment server




- ACE operation extracts bit value of cells indicated by ACL data
  - No additional computation is required in reconstruction

# AR-based FE—Features

- Security
  - No entropy leakage, and $\sigma$-bit random seed realizes $\sigma$-bit key generation
- Efficiency
  - Retained entropy via debiasing is given by $2mp_0$ (for $p_1 \geq p_0$) from $m$-bit PUF

VNC [MLSW15]: $2mp_1p_0$, (Simplest one), MD [AWSO17]: $m/\mu$ ($\mu \geq 3$ for most cases), and TD [S17]: $2mp_0 - 2$

- Reliability
  - AR-based FE may fail enrollment if length of extracted bit string is insufficient
    - PUF size should be determined such that enrollment failure rate is smaller than threshold
    - Enrollment failure rate is feasibly calculated similarly to VNC-based FEs
  - RRS and ACE operations have no impact on bit-error rate of extracted bits
    - ECC can be designed in the same way as conventional FEs
- Implementation aspects
  - RNG and bit-parallel operation at enrollment are required as main overhead
  - Reconstruction require no additional computationally-critical operations

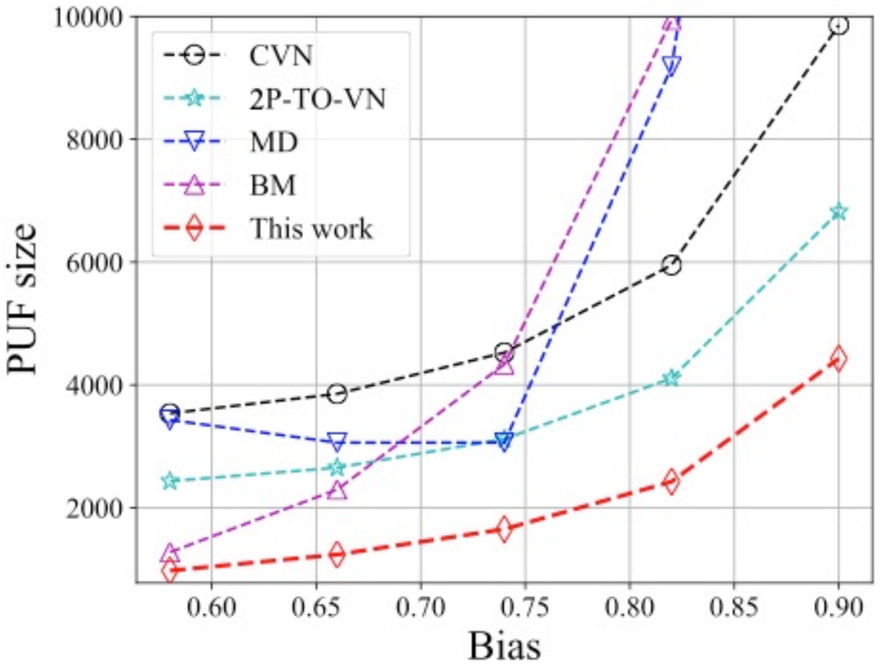
# Performance evaluation

- Simulate 128-bit key generation to evaluate PUF and NVM sizes (i.e., hardware cost) for various biases and bit-error rates
  - PUF bias: 0.58—0.90
  - Bit-error rate: 0.025—0.100
  - ECC: BCH-repetition concatenate code
    - BCH codes with length of 7, 15, 31, 63, 127, and 255 are considered
  - Enrollment and reconstruction failure rates are set less than $10^{-6}$
  - Compared to VNC-, MD-, and BM-based FEs herein [MLSW15, AWSO17, USH19]
    - See our paper for comparison with other conventional FEs

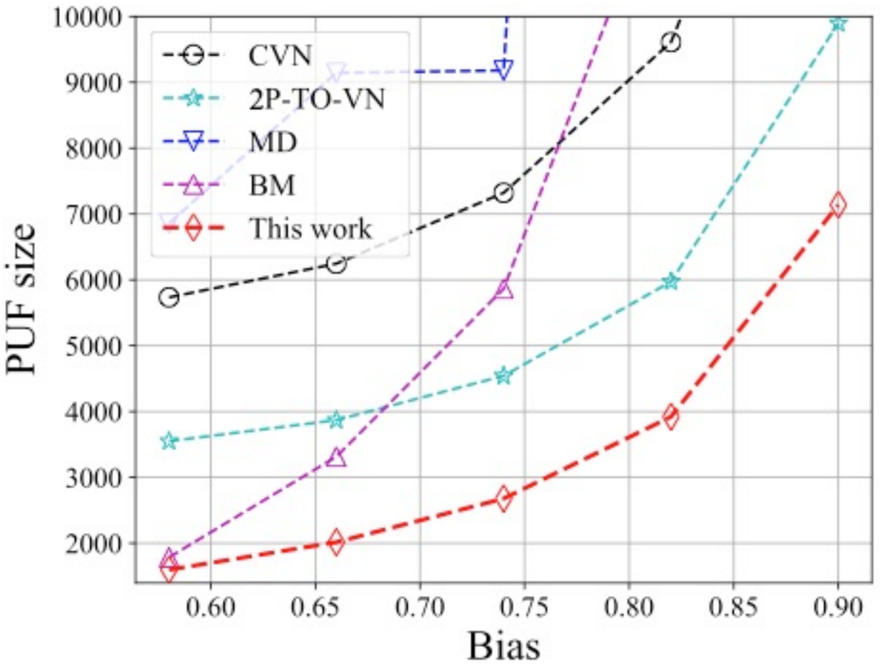
[MLSW15] R. Maes et al., Secure key generation from biased PUFs, *CHES 2015*.
[AWSO17] A. Aysu et al., A new maskless debiasing method for lightweight physical unclonable function, *HOST 2017*.
[USH19] R. Ueno et al., Tackling biased PUFs through biased masking: A debiasing method for efficient fuzzy extractor, *IEEE TC*, 2019.

# Evaluation result

Bit-error rate = 0.050




Bit-error rate = 0.100




- AR-based FE achieves highest efficiency for most biases and bit-error rates
  - At most 55% smaller PUF size
  - NVM size is basically consistent with PUF size

# Concluding remarks

- We present AR-based FE which extracts uniform distribution from biased PUFs based on rejection sampling
  - Implemented using RNG and bit-parallel operations on enrollment server
  - Client device with PUF requires no computational overhead
  - First debiasing scheme applicable to PUFs with local biases
  - Simulation of 128-bit key generation shows that AR-based FE has higher efficiency for most biases and bit-error rates than conventional FEs, and achieves at most 55% and/or 72% smaller PUF and NVM sizes respectively
  - Extended to ternary PUF response for improved efficiency (see our paper)
    - More efficient for many PUFs than counterparts (i.e., ternary VNC-based FEs and C-IBS)
- Future works
  - Real-world implementation and evaluation of key generation system based on AR-based FE
  - Extension of AR-based FE for secure reuse of PUF