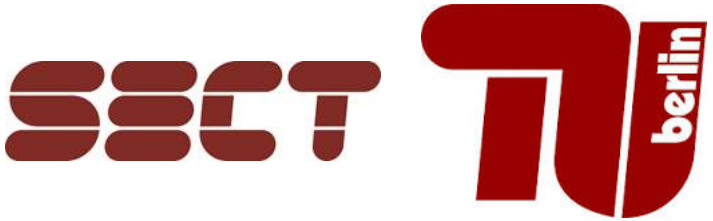
Security of PUFs: Lessons Learned after Two Decades of Research

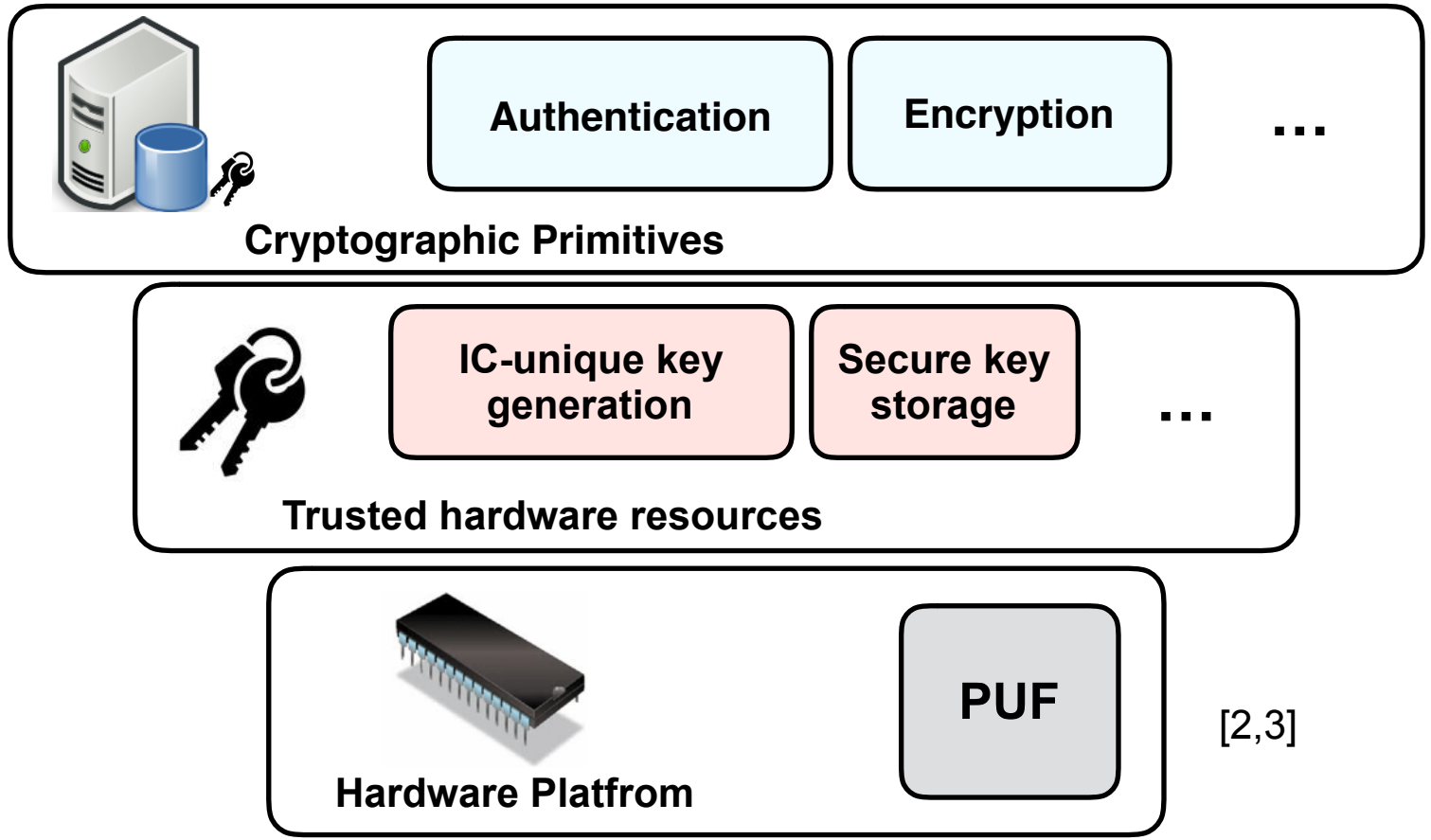
Fatemeh Ganji, Shahin Tajik, Domenic Forte, and Jean-Pierre Seifert

CHES 2019 Tutorial



Motivation: Hardware Root of Trust (RoT)

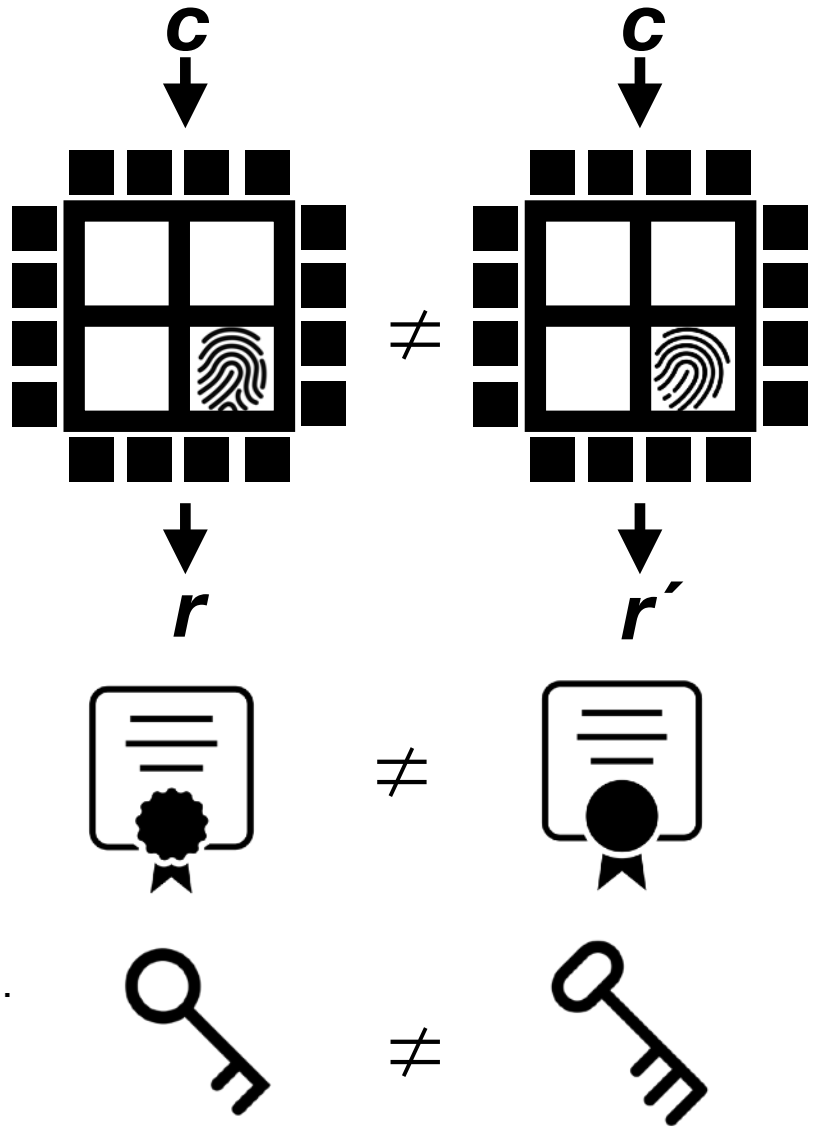
- Reliance of cryptographic protocols on secrets and random numbers
- "A root of trust is a component at a lower abstraction layer, upon which the system relies for its security." [1]



[1] Verbauwhede, , Ingrid, The need for hardware roots of trust, keynote speech, HOST 2019.
[2] Roel, Maes., 2012. Physically unclonable functions: Constructions, properties and applications. Katholieke Universiteit Leuven, Belgium.
[3] Ganji, Fatemeh. On the learnability of physically unclonable functions. Springer International Publishing, 2018.

Phyiscal(ly) Unclonable Function (PUF)

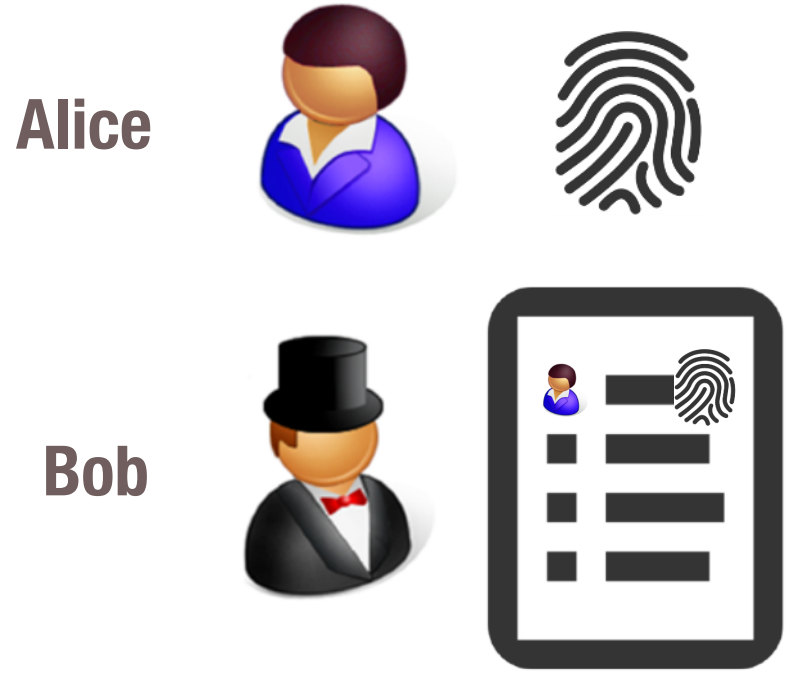
- Exploiting manufacturing process variations on different chips
- Physical entity that is embodied in a physical structure
- Easy to evaluate but hard to predict!
- Easy to make but practically impossible to duplicate
- Not a true function in a mathematical sense: one possible input >> more possible output

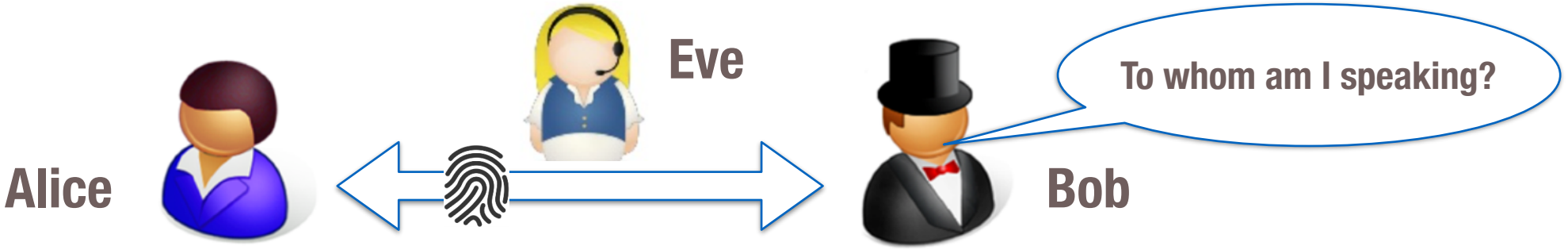


[1] Gassend, Blaise, et al. "Silicon physical random functions." Proceedings of the 9th ACM conference on Computer and communications security. ACM, 2002.
[2] Suh, G. Edward, and Srinivas Devadas. "Physical unclonable functions for device authentication and secret key generation." 2007 44th ACM/IEEE Design Automation Conference. IEEE, 2007.

Applications: Authentication & Key Generation

- PUF is used in two phases:
- **Enrolment:** A number of CRPs are collected and stored in the database (CRP database)
- **Verification:** A challenge from CRP database is applied to the PUF and the response compared with the corresponded response in data base
- Observed response close enough >> **verified!**
- Key Storing (**No key is stored actually!**)
- Key is generated when needed!





PUF Properties

[1] Maes, R and Verbauwhede, I (2008). Physically unclonable functions: A study on the state of the art and future research directions. In Towards Hardware-Intrinsic Security, (pp. 3-37). Springer

PUF Properties

1. **Evaluable:** given Π and x, it is easy to evaluate y = Π(x).

[1] Maes, R and Verbauwhede, I (2008). Physically unclonable functions: A study on the state of the art and future research directions. In Towards Hardware-Intrinsic Security, (pp. 3-37). Springer

PUF Properties

1. **Evaluable:** given Π and x, it is easy to evaluate y = Π(x).
2. **Unique:** Π(x) contains some information about the identity of the physical entity embedding Π.

[1] Maes, R and Verbauwhede, I (2008). Physically unclonable functions: A study on the state of the art and future research directions. In Towards Hardware-Intrinsic Security, (pp. 3-37). Springer

PUF Properties

1. **Evaluable:** given Π and x, it is easy to evaluate y = Π(x).
2. **Unique:** Π(x) contains some information about the identity of the physical entity embedding Π.
3. **Reproducible:** y = Π(x) is reproducible up to a small error.

[1] Maes, R and Verbauwhede, I (2008). Physically unclonable functions: A study on the state of the art and future research directions. In Towards Hardware-Intrinsic Security, (pp. 3-37). Springer

PUF Properties

1. **Evaluable:** given Π and x, it is easy to evaluate y = Π(x).
2. **Unique:** Π(x) contains some information about the identity of the physical entity embedding Π.
3. **Reproducible:** y = Π(x) is reproducible up to a small error.
4. **Unclonable:** given Π, it is hard to construct Γ≠ Π such that for all x in X: Γ(x) ≠ Π(x) up to a small error.

[1] Maes, R and Verbauwhede, I (2008). Physically unclonable functions: A study on the state of the art and future research directions. In Towards Hardware-Intrinsic Security, (pp. 3-37). Springer

PUF Properties

1. **Evaluable:** given Π and x, it is easy to evaluate y = Π(x).
2. **Unique:** Π(x) contains some information about the identity of the physical entity embedding Π.
3. **Reproducible:** y = Π(x) is reproducible up to a small error.
4. **Unclonable:** given Π, it is hard to construct Γ≠ Π such that for all x in X: Γ(x) ≠ Π(x) up to a small error.
5. **Unpredictable:** given only a set Q = {(xi,yi = Π(xi))}, it is hard to predict yc ≈ Π(xc) up to small error, for xc a random challenge such that (xc,.) ∉ Q.

[1] Maes, R and Verbauwhede, I (2008). Physically unclonable functions: A study on the state of the art and future research directions. In Towards Hardware-Intrinsic Security, (pp. 3-37). Springer

PUF Properties

1. **Evaluable:** given Π and x, it is easy to evaluate y = Π(x).
2. **Unique:** Π(x) contains some information about the identity of the physical entity embedding Π.
3. **Reproducible:** y = Π(x) is reproducible up to a small error.
4. **Unclonable:** given Π, it is hard to construct Γ≠ Π such that for all x in X: Γ(x) ≠ Π(x) up to a small error.
5. **Unpredictable:** given only a set Q = {(xi,yi = Π(xi))}, it is hard to predict yc ≈ Π(xc) up to small error, for xc a random challenge such that (xc,.) ∉ Q.
6. **One-way:** given only y and Π, it is hard to find x such that y = Π(x).

[1] Maes, R and Verbauwhede, I (2008). Physically unclonable functions: A study on the state of the art and future research directions. In Towards Hardware-Intrinsic Security, (pp. 3-37). Springer

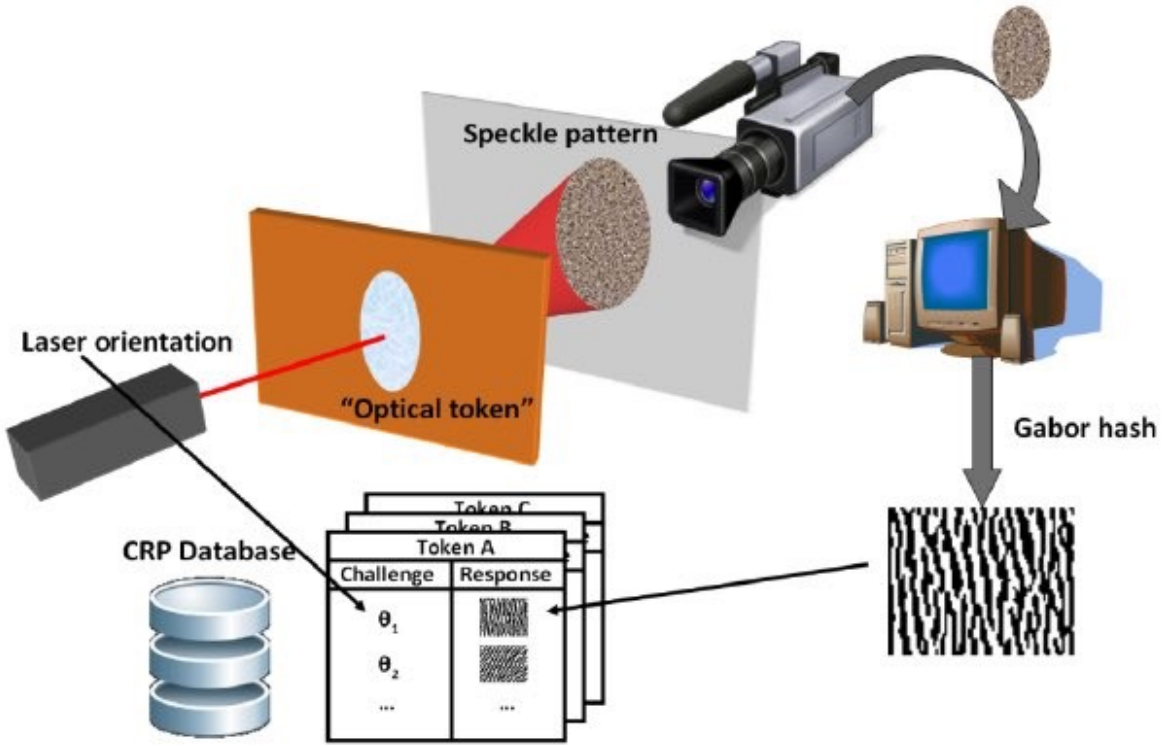
PUF Properties

1. **Evaluable:** given Π and x, it is easy to evaluate y = Π(x).
2. **Unique:** Π(x) contains some information about the identity of the physical entity embedding Π.
3. **Reproducible:** y = Π(x) is reproducible up to a small error.
4. **Unclonable:** given Π, it is hard to construct Γ≠ Π such that for all x in X: Γ(x) ≠ Π(x) up to a small error.
5. **Unpredictable:** given only a set Q = {(xi,yi = Π(xi))}, it is hard to predict yc ≈ Π(xc) up to small error, for xc a random challenge such that (xc,.) ∉ Q.
6. **One-way:** given only y and Π, it is hard to find x such that y = Π(x).
7. **Tamper evident:** altering the physical Π transforms Π→Π´ such that with high probability ∃x ∈ X: Π(x) ≠ Π´(x).

[1] Maes, R and Verbauwhede, I (2008). Physically unclonable functions: A study on the state of the art and future research directions. In Towards Hardware-Intrinsic Security, (pp. 3-37). Springer

Non-electronical PUFs

- Non-electronic constructions with PUF-like properties
- Electronic and digital techniques are used to process the PUF responses
- Example: **Optical PUF [1]**
  - The core element: Optical token with microscopic structures
  - Irradiating the token with a different laser orientations (challenge) to create a speckle pattern
  - Gabor hashing of the image to get the response

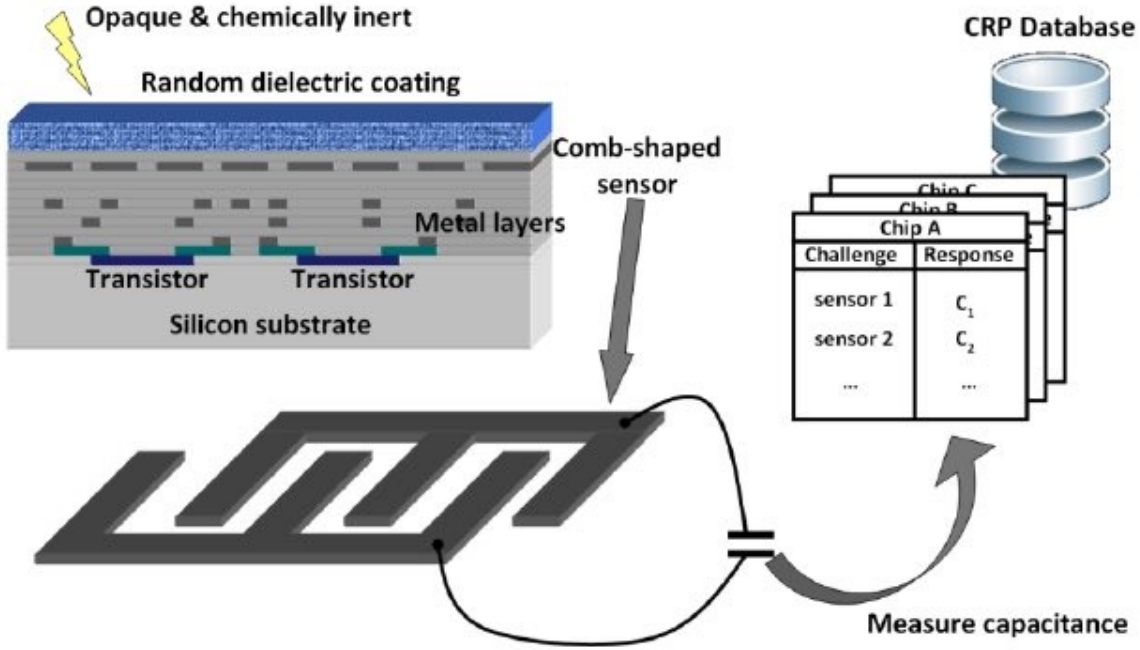


[2]

[1] Pappu, Ravikanth, et al. "Physical one-way functions." Science 297.5589 (2002): 2026-2030.
[2] Maes, R and Verbauwhede, I (2008). Physically unclonable functions: A study on the state of the art and future research directions In Towards Hardware-Intrinsic Security, (pp. 3-37). Springer

Electrical PUFs

- Electrical/Electronical PUFs (analog responses):
- PUF constructions whose basic operation consists of an analog measurement of an electric or electronic quantity
- Example: **Coating PUF [1]**
- Comb-shaped sensors in the metal layer of the IC
- Random dielectric coating sprayed on top of the sensor
- Challenge: Sensor selection
- Response: Capacitance measurement



[2]

[1] Skoric, B., Maubach, S., Kevenaar, T. A., & Tuyls, P. (2006). Information-theoretic analysis of coating PUFs. IACR Cryptology ePrint Archive, 2006, 101.

[2] Maes, R and Verbauwhede, I (2008). Physically unclonable functions: A study on the state of the art and future research directions In Towards Hardware-Intrinsic Security, (pp. 3-37). Springer

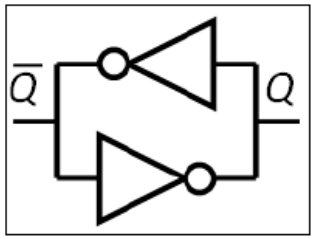
Silicon PUFs

- Digital Intrinsic Silicon PUFs [1]:
- PUF and measurement system should be fully integrated in the embedding device
- PUF should be constructible by available manufacturing process of embedding device
- Two categories based on the number of challenge-response pairs [2]:
- **Weak PUFs:** SRAM PUFs, Butterfly PUFs, Ring-Oscillator PUFs, etc.
- **Strong PUFs:** Arbiter PUFs, Bistable Ring PUFs, etc.

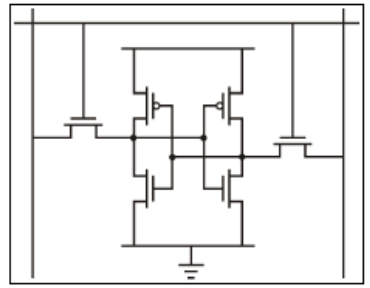
[1] Roel, M. A. E. S. "Physically unclonable functions: Constructions, properties and applications." Katholieke Universiteit Leuven, Belgium (2012).
[2] Rührmair, U., Sehnke, F., Sölter, J., Dror, G., Devadas, S., & Schmidhuber, J. (2010, October). Modeling attacks on physical unclonable functions. In Proceedings of the 17th ACM conference on Computer and communications security (pp. 237-249). ACM.

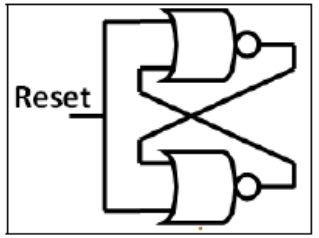
SRAM & Butterfly PUFs

- Example of Memory-based PUFs: SRAM PUFs
- Using the bistability behaviour of SRAM cells
- Bistability because of MOSFET mismatches
- **Assumption: Attacker cannot readout the SRAM or Register values!**

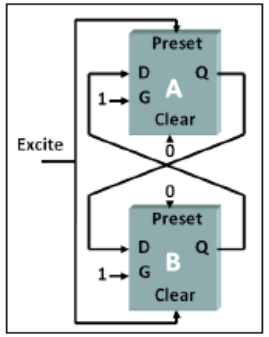


(a) Logical circuit of an SRAM (PUF) cell.


(b) Electrical circuit of an SRAM (PUF) cell in standard CMOS technology.



(c) Logical circuit of a latch (PUF) cell.

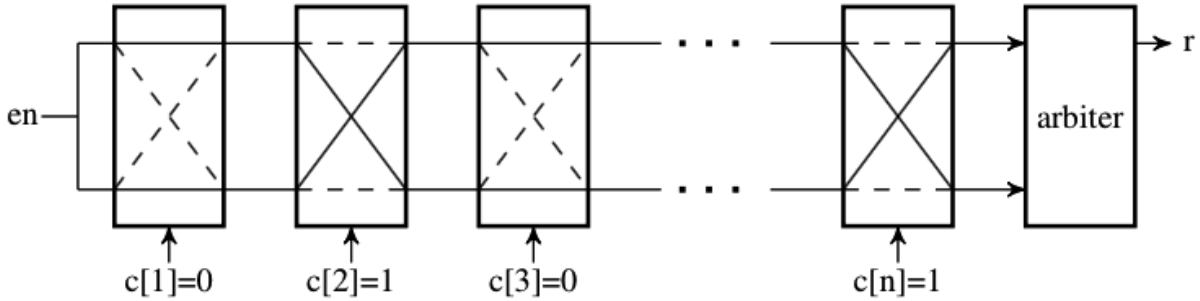


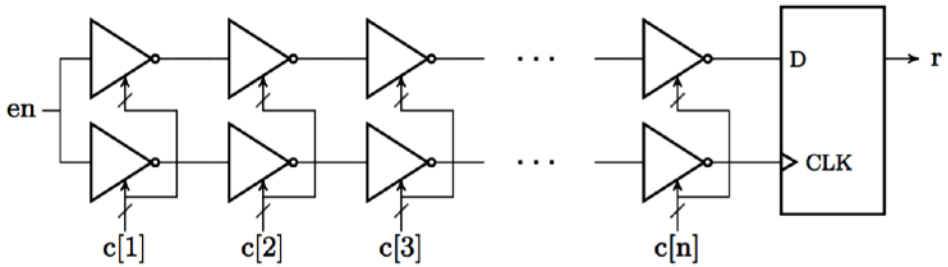
(d) Schematical circuit of a butterfly PUF cell.

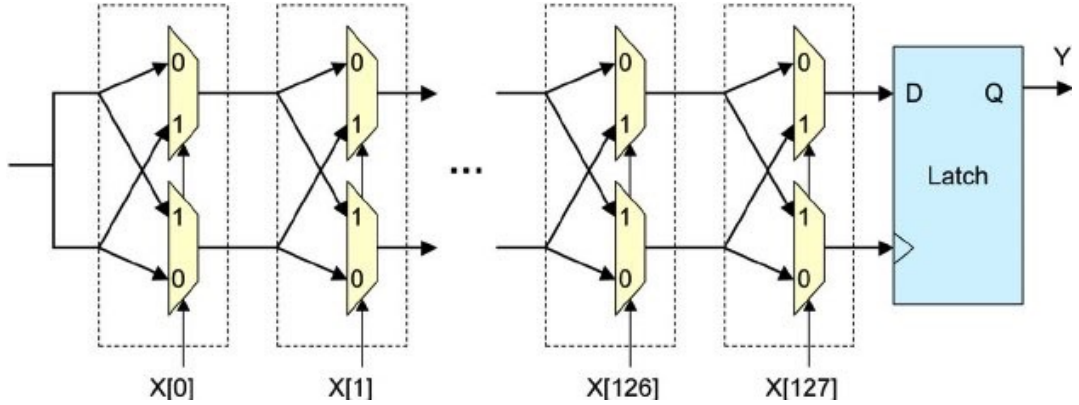
[1] [2] Maes, R and Verbauwhede, I (2008). Physically unclonable functions: A study on the state of the art and future research directions. In Towards Hardware-Intrinsic Security, (pp. 3-37). Springer
[2] Kumar, Sandeep S., et al. "The butterfly PUF protecting IP on every FPGA." 2008 IEEE International Workshop on Hardware-Oriented Security and Trust. IEEE, 2008.
[3] Roel, M. A. E. S. "Physically unclonable functions: Constructions, properties and applications." Katholieke Universiteit Leuven, Belgium (2012).

Arbiter PUFs

- Utilizing intrinsic timing differences of 2 symmetrically designed electrical paths
- Direct or crossed paths in each stage based on challenge bit
- Binary response by the Arbiter based on arrival of first signal
- **Assumption: Attacker cannot measure individual delays!**



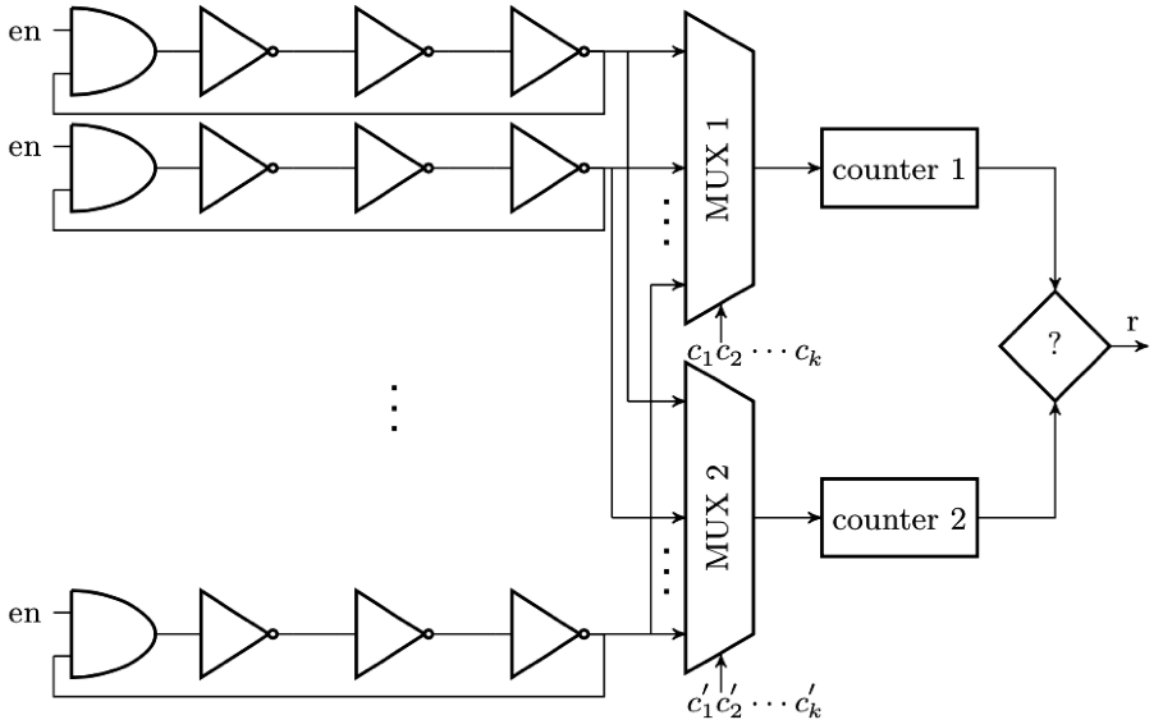




[1] Lee, J.W., Lim, D., Gassend, B., Suh, G.E., Van Dijk, M. and Devadas, S., 2004, June. A technique to build a secret key in integrated circuits for identification and authentication applications. In VLSI Circuits, 2004. Digest of Technical Papers. 2004 Symposium on (pp. 176-179). IEEE.

Ring-Oscillator PUFs

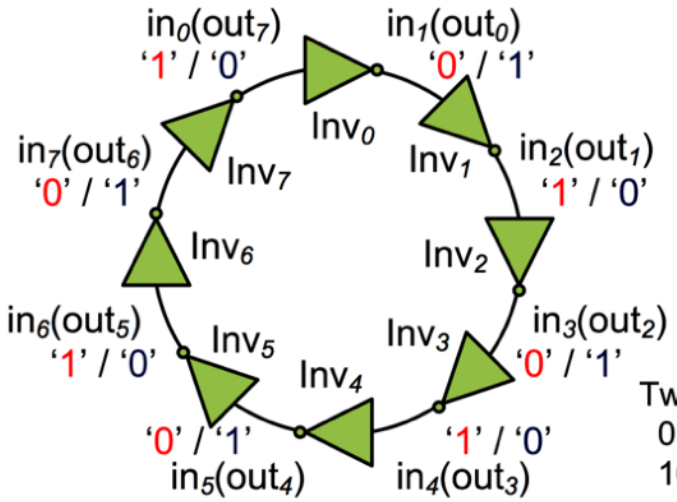
- Ring oscillators generates a clock like signal
- The frequency is partially random
- Two ROs are selected and their frequencies are compared to generate a binary response!
- **Assumption: Attacker cannot measure the ring frequencies!**

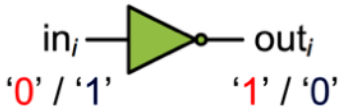


[1] Suh, G. Edward, and Srinivas Devadas. "Physical unclonable functions for device authentication and secret key generation." 2007 44th ACM/IEEE Design Automation Conference. IEEE, 2007.

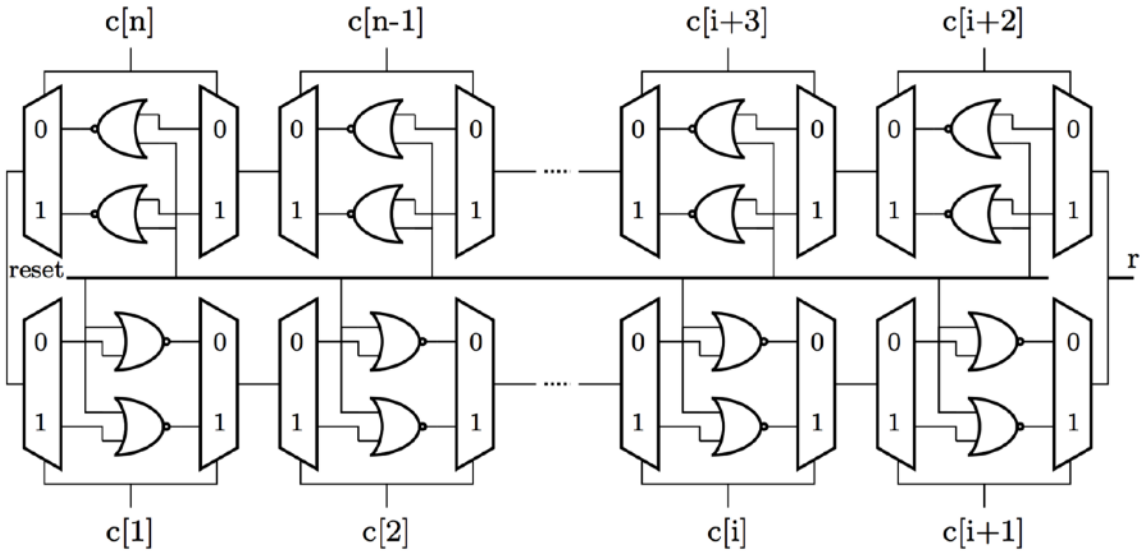
Bistable Ring PUFs

- Using bistability of inverter chains (similar to a larger SRAM cell)
- Combining 2n inverters in loop to have an exponential challenge space
- **Assumption: The exact mathematical model is not known!**





Two possible final stable states ($out_0 ... out_7$):
$01010101_{BIN} = 55_{HEX}$ ("5"-state)
$10101010_{BIN} = AA_{HEX}$ ("A"-state)



[1] Chen, Qingqing, et al. "The bistable ring PUF: A new architecture for strong physical unclonable functions." *2011 IEEE International Symposium on Hardware-Oriented Security and Trust*. IEEE, 2011.

Metrics

Metrics for evaluating the quality of PUFs (1) [1]

- **Uniqueness: how unique are the PUF responses among different chips**
  - **n-bit *response* from a PUF**

$$\frac{2}{m(m-1)} \sum_{u=1}^{m-1} \sum_{v=u+1}^{m} \frac{HD(R_u, R_v)}{n} \times 100\%$$

 # Chips

 # Diff. Chips

  - **pair-wise HDs among chips.**
  - **For a *truly random* PUF output, it should be close to 50%.**

[1] Maiti, A., Casarona, J., McHale, L. and Schaumont, P., 2010, June. A large scale characterization of RO-PUF. In *2010 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)* (pp. 94-99). IEEE.

Metrics for evaluating the quality of PUFs (2) [1]

- **Uniformity:**

$$\left(\frac{1}{n}\sum_{t=1}^{n} r_{i,t}\right) \times 100\%$$

  - **It should be close to 50%.**

- **Bit-aliasing: e.g., the t-th bit has same binary value across all the chips.**

$$\left(\frac{1}{m}\sum_{i=1}^{m} r_{i,t}\right) \times 100\%$$

  - **It should be close to 50%.**

[1] Maiti, A., Casarona, J., McHale, L. and Schaumont, P., 2010, June. A large scale characterization of RO-PUF. In *2010 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)* (pp. 94-99). IEEE.

Metrics for evaluating the quality of PUFs (3) [1]

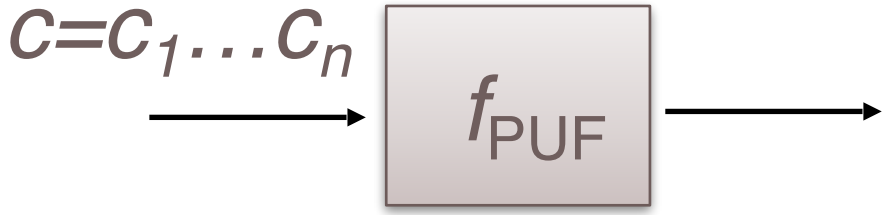
- **Reliability: Reliability quantifies the change in PUF outputs over varying operating conditions.**

- **Estimated as the average intra-die Hamming distance i.e. HD(R , R') over x samples:**

$$\frac{1}{x}\sum_{y=1}^{x}\frac{HD(R_i, R'_{i,y})}{n} \times 100\%$$

Metrics for evaluating the quality of PUFs (3) [1]

- **Reliability: Reliability quantifies the change in PUF outputs over varying operating conditions.**

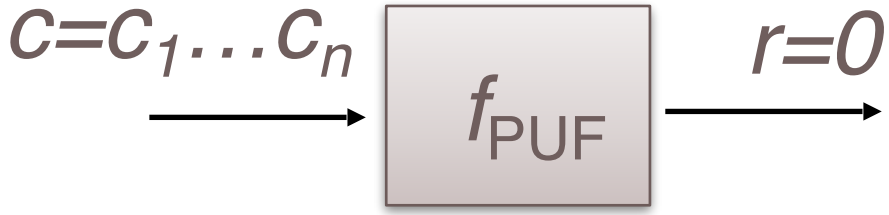


- **Estimated as the average intra-die Hamming distance i.e. HD(R , R') over x samples:**

$$\frac{1}{x}\sum_{y=1}^{x}\frac{HD(R_i, R'_{i,y})}{n} \times 100\%$$

Metrics for evaluating the quality of PUFs (3) [1]

- **Reliability: Reliability quantifies the change in PUF outputs over varying operating conditions.**

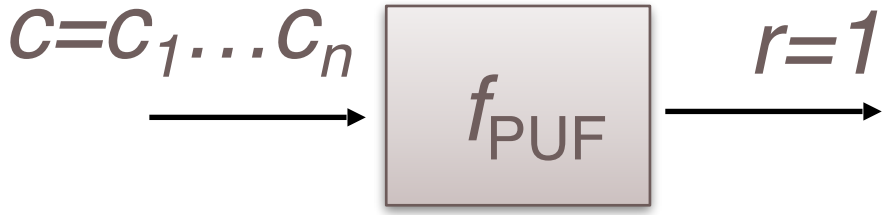


- **Estimated as the average intra-die Hamming distance i.e. HD(R , R') over x samples:**

$$\frac{1}{x}\sum_{y=1}^{x}\frac{HD(R_i, R'_{i,y})}{n}\times 100\%$$

Metrics for evaluating the quality of PUFs (3) [1]

- **Reliability: Reliability quantifies the change in PUF outputs over varying operating conditions.**

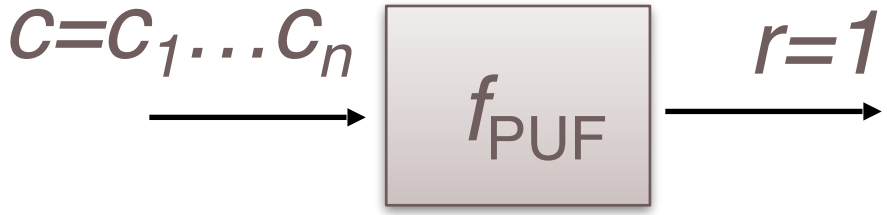


- **Estimated as the average intra-die Hamming distance i.e. HD(R , R') over x samples:**

$$\frac{1}{x}\sum_{y=1}^{x}\frac{HD(R_i, R'_{i,y})}{n}\times 100\%$$

Metrics for evaluating the quality of PUFs (3) [1]

- **Reliability: Reliability quantifies the change in PUF outputs over varying operating conditions.**

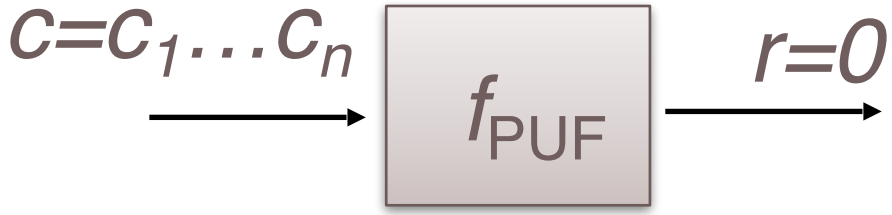


- **Estimated as the average intra-die Hamming distance i.e. HD(R , R') over x samples:**

$$\frac{1}{x}\sum_{y=1}^{x}\frac{HD(R_i, R'_{i,y})}{n}\times 100\%$$

Metrics for evaluating the quality of PUFs (3) [1]

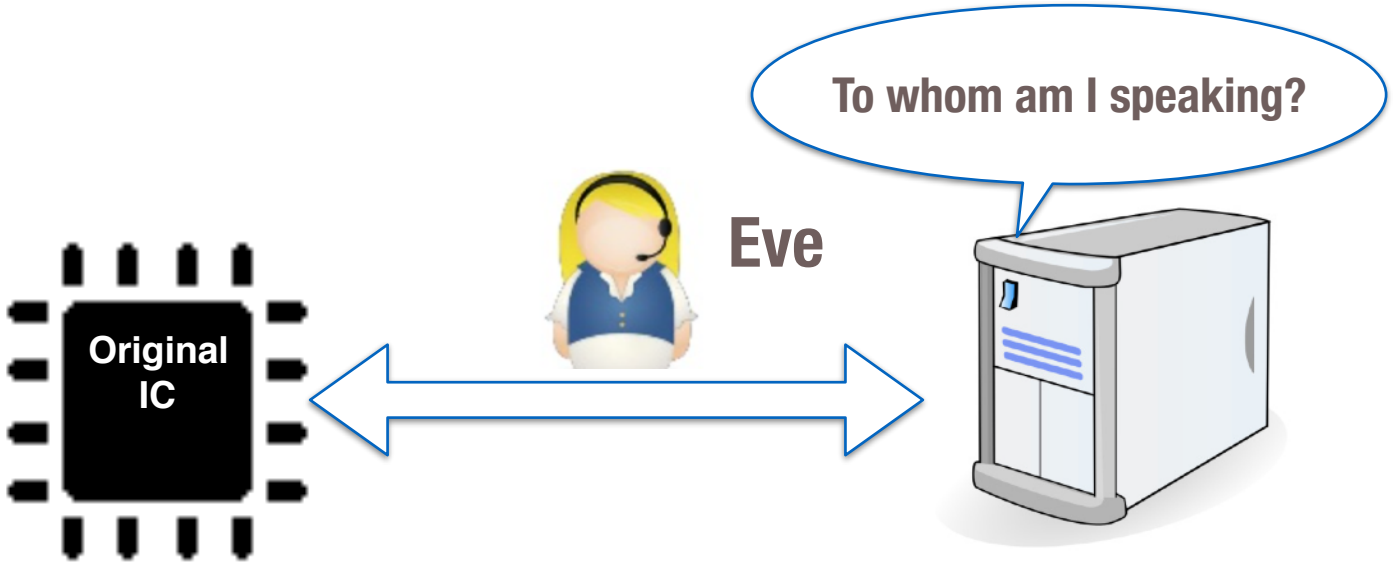
- **Reliability: Reliability quantifies the change in PUF outputs over varying operating conditions.**



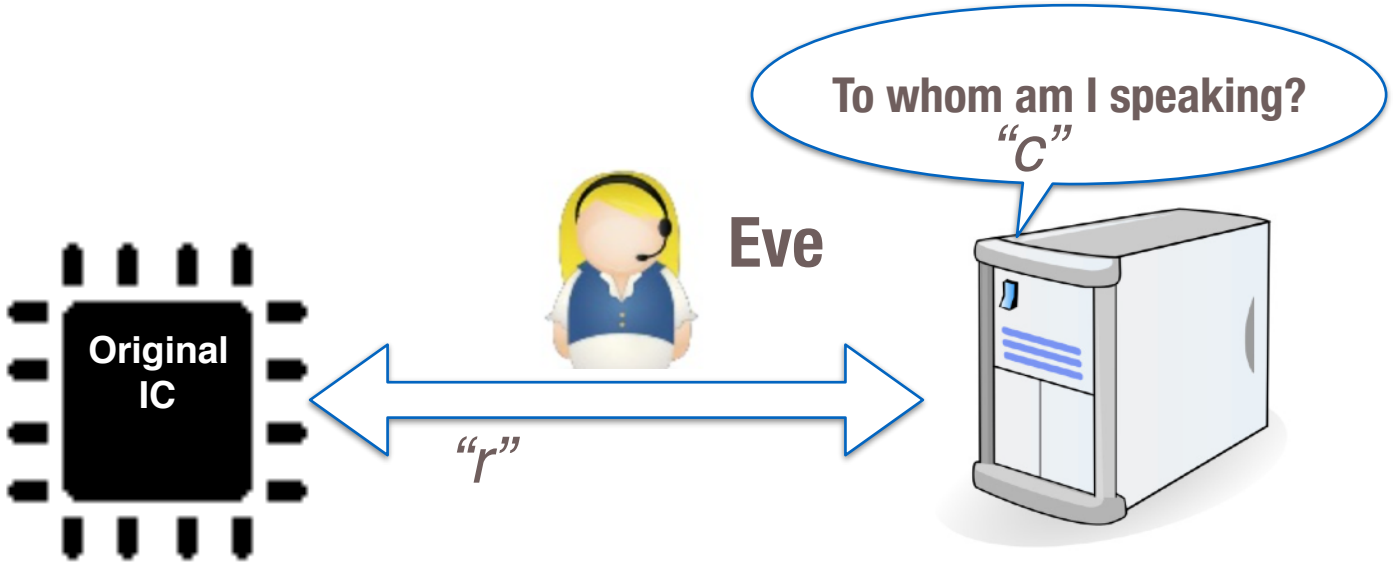
- **Estimated as the average intra-die Hamming distance i.e. HD(R , R') over x samples:**

$$\frac{1}{x}\sum_{y=1}^{x}\frac{HD(R_i, R'_{i,y})}{n}\times 100\%$$

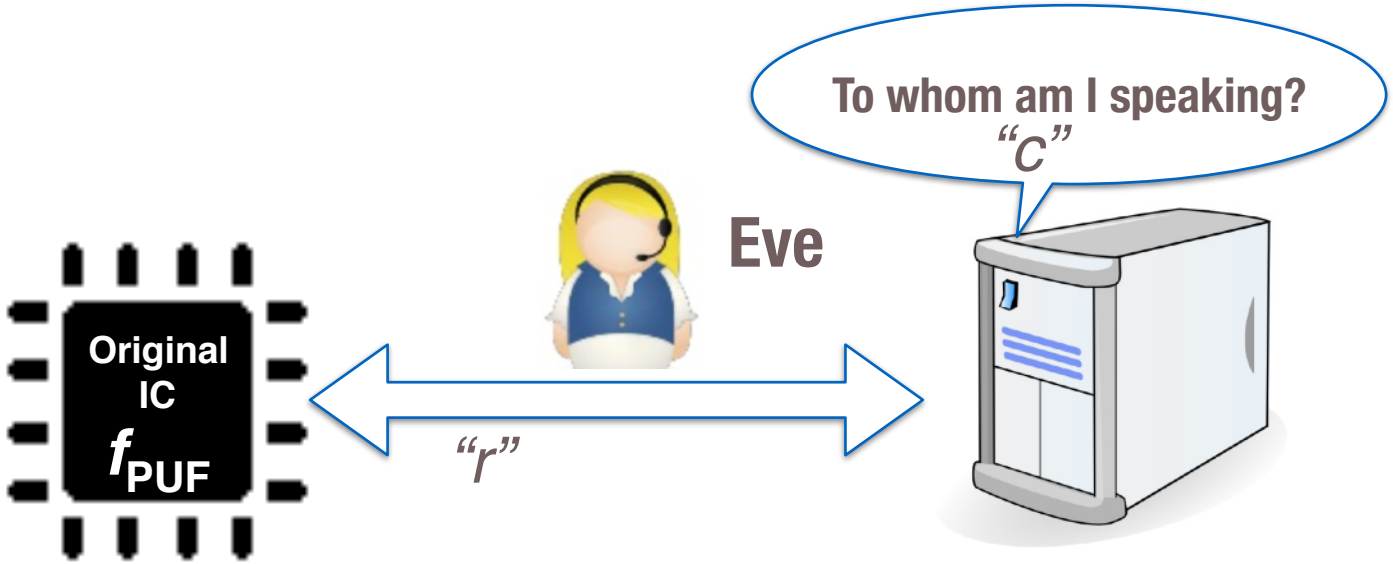
Fuzzy behavior



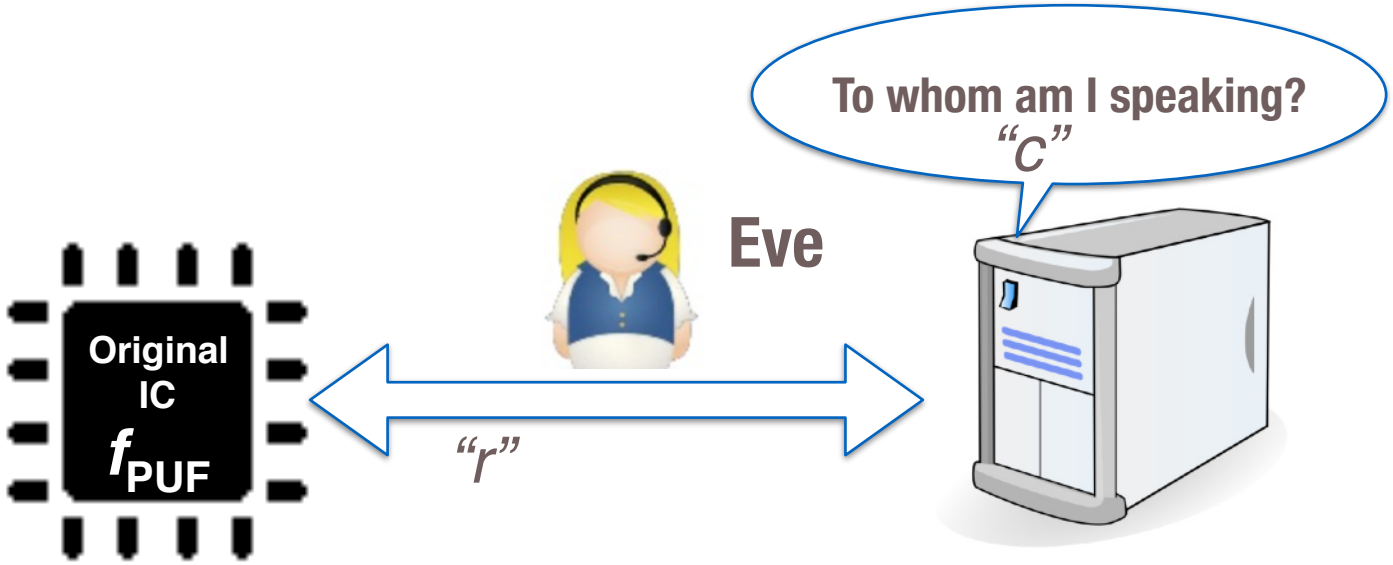
Fuzzy behavior



Fuzzy behavior

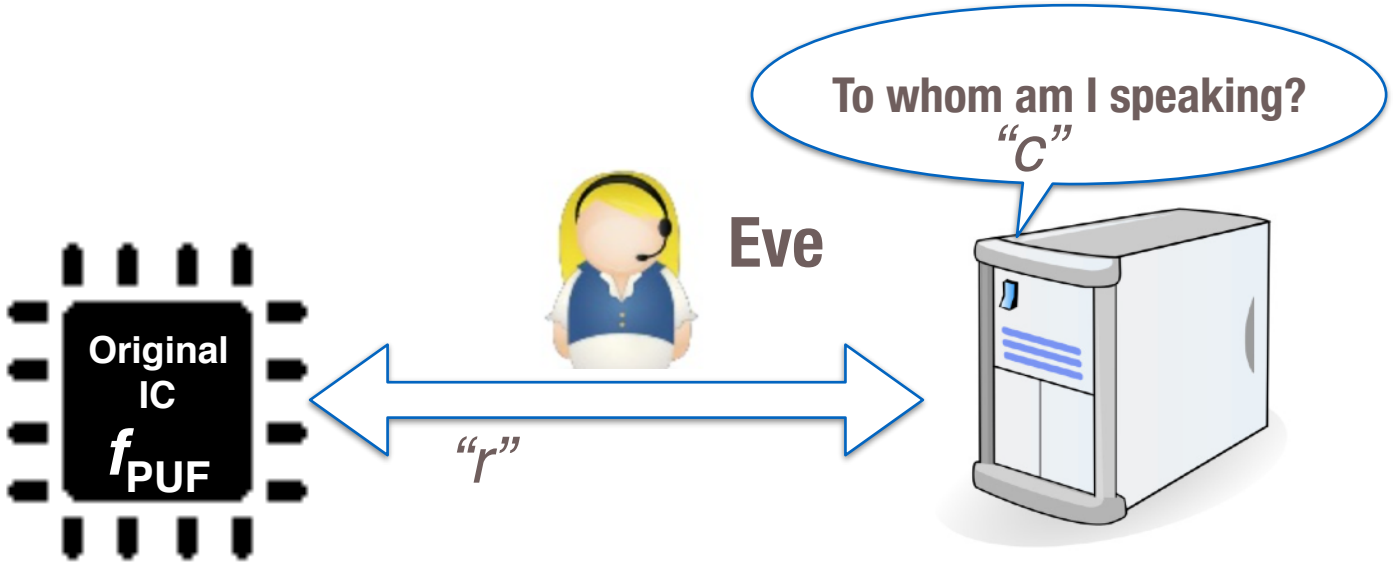


Fuzzy behavior



- r is fuzzy
  - it is not entirely uniformly distributed
  - it is not perfectly reproducible when measured multiple times

Fuzzy behavior



- r is fuzzy
  - it is not entirely uniformly distributed
  - it is not perfectly reproducible when measured multiple times
- Due to the physical nature:
  - random physical processes that introduce entity-specific features during manufacturing are typically not uniformly distributed
  - the response evaluation mechanisms of a PUF construction are subject to physical noise and environmental conditions

False Acceptance, False Rejection, and Equal Error Rates

False Acceptance, False Rejection, and Equal Error Rates

- During the identification phase of a PUF-based identification system
  - response of an entity is checked against a list of enrolled responses
  - when an enrolled response is found whose distance to the presented response is smaller or equal to the identification threshold, then the entity is identified as the matching entry in the list.

Fuzzy identification system based on such a identification threshold is not 100% reliable

Machine Learning Attacks
What can be learned by a designer?

Learnability and security

Source: tagxedo.com

[1] Rivest, R.L., 1991, November. Cryptography and machine learning. In International Conference on the Theory and Application of Cryptology (pp. 427-439). Springer, Berlin, Heidelberg.

Learnability and security

- **Machine learning and cryptanalysis sharing same notions and concerns**

Source: tagxedo.com

[1] Rivest, R.L., 1991, November. Cryptography and machine learning. In International Conference on the Theory and Application of Cryptology (pp. 427-439). Springer, Berlin, Heidelberg.

Learnability and security

- **Machine learning and cryptanalysis sharing same notions and concerns**

  - **Wishes of a cryptanalyst: "breaking" some cryptosystem to find the secret key used by the users**

Source: tagxedo.com

[1] Rivest, R.L., 1991, November. Cryptography and machine learning. In International Conference on the Theory and Application of Cryptology (pp. 427-439). Springer, Berlin, Heidelberg.

Learnability and security

- **Machine learning and cryptanalysis sharing same notions and concerns**

  - **Wishes of a cryptanalyst: "breaking" some cryptosystem to find the secret key used by the users**

Source: tagxedo.com

[1] Rivest, R.L., 1991, November. Cryptography and machine learning. In International Conference on the Theory and Application of Cryptology (pp. 427-439). Springer, Berlin, Heidelberg.

Learnability and security

- **Machine learning and cryptanalysis sharing same notions and concerns**
  - **Wishes of a cryptanalyst: "breaking" some cryptosystem to find the secret key used by the users**
- **Similar to the problem of "learning an unknown function": good cryptography can provide examples of classes of functions that are hard to learn [1].**

Source: tagxedo.com

[1] Rivest, R.L., 1991, November. Cryptography and machine learning. In International Conference on the Theory and Application of Cryptology (pp. 427-439). Springer, Berlin, Heidelberg.

Learnability and security

- **Machine learning and cryptanalysis sharing same notions and concerns**
  - **Wishes of a cryptanalyst: "breaking" some cryptosystem to find the secret key used by the users**
- **Similar to the problem of "learning an unknown function": good cryptography can provide examples of classes of functions that are hard to learn [1].**

Source: tagxedo.com

[1] Rivest, R.L., 1991, November. Cryptography and machine learning. In International Conference on the Theory and Application of Cryptology (pp. 427-439). Springer, Berlin, Heidelberg.

“In cryptography, the major goal is to “prove” security under the broadest possible definition of security, [...].
[...] , in the typical paradigm it is shown that there is no polynomial-time [learning] algorithm that can “break” the security of the system. ”

[1] Rivest, R.L., 1991, November. Cryptography and machine learning. In International Conference on the Theory and Application of Cryptology (pp. 427-439). Springer, Berlin, Heidelberg.

An example of their differences: Exact vs. approximate inference

An example of their differences: Exact vs. approximate inference

- In the practical cryptographic domain: a “total break” is needed, i.e., the attacker determines the unknown secret key.
  - Typically, it is not possible to well approximate the set of possible cryptographic functions.
  - The theoretical cryptography: definitions of security excluding even approximate inference by the cryptanalyst. Such theoretical definitions and corresponding results are thus applicable to derive results on the difficulty of (even approximately) learning, as we will see.

An example of their differences: Exact vs. approximate inference

- In the practical cryptographic domain: a “total break” is needed, i.e., the attacker determines the unknown secret key.
  - Typically, it is not possible to well approximate the set of possible cryptographic functions.
  - The theoretical cryptography: definitions of security excluding even approximate inference by the cryptanalyst. Such theoretical definitions and corresponding results are thus applicable to derive results on the difficulty of (even approximately) learning, as we will see.
- In the machine learning field: both exact inference and approximate inference.
  - Because exact inference is often too difficult to perform efficiently, much of the more recent research in this area deals with approximate inference.

Machine learning (ML) attacks against PUFs

Silicon Physical Random Functions*

Blaise Gassend, Dwaine Clarke, Marten van Dijk† and Srinivas Devadas
Massachusetts Institute of Technology
Laboratory for Computer Science
Cambridge, MA 02139, USA
{gassend,declarke,marten,devadas}@mit.edu

ABSTRACT

We introduce the notion of a Physical Random Function (PUF). We argue that a complex integrated circuit can be viewed as a silicon PUF and describe a technique to identify and authenticate individual integrated circuits (ICs).

We describe several possible circuit realizations of different PUFs. These circuits have been implemented in commodity Field Programmable Gate Arrays (FPGAs). We present experiments which indicate that reliable authentication of individual FPGAs can be performed even in the presence of significant environmental variations.

We describe how secure smart cards can be built, and also briefly describe how PUFs can be applied to licensing and certification applications.

Categories and Subject Descriptors

C.3 [**Special-Purpose and Application-Based Systems**]: Smartcards

General Terms

Measurement, Experimentation, Security

Keywords

Identification, physical random function, physical security, smartcard, tamper resistance, unclonability

1. INTRODUCTION

We describe the notion of Physical Random Functions (PUFs) and argue that PUFs can be implemented using conventional integrated circuit (IC) design techniques. This leads us to a method of identifying and authenticating individual ICs and a means of building secure smartcards. A host of other applications are also possible.

Many methods are already available to identify and authenticate ICs. One can embed a unique identifier in an IC to give it a unique identity. This approach can identify the IC, but cannot authenticate it. To enable authentication, one needs to embed a secret key onto the IC. Of course, for the system to work, this key needs to remain secret, which means that the packaged IC has to be made resistant to attacks that attempt to discover the key. Numerous attacks are described in the literature. These attacks may be invasive, e.g., removal of the package and layers of the IC, or non-invasive, e.g., differential power analysis that attempts to determine the key by stimulating the IC and observing the power and ground rails. Making an IC tamper-resistant to all forms of attacks is a challenging problem and is receiving some attention [1]. IBM's PCI Cryptographic Coprocessor encapsulates a 486-class processing subsystem within a tamper-sensing and tamper-responding environment where one can run security-sensitive processes [13]. However, providing high-grade tamper resistance, which makes it impossible for an attacker to access or modify the secrets held inside a device, is expensive and difficult [2, 3].

We propose a completely different approach to IC authentication in this paper. Our thesis is that there is enough manufacturing process variations across ICs with identical masks to uniquely characterize each IC, and this characterization can be performed with a large signal-to-noise ratio (SNR). The characterization of an IC involves the generation of a set of challenge-response pairs. To authenticate ICs we require the set of challenge-response pairs to be characteristic of each IC. For reliable authentication, we require that environmental variations and measurement errors do not produce so much noise that they hide inter-IC variations. We will show in this paper, using experiments and analysis, that we can perform reliable authentication using the techniques that we now introduce.

How can we produce a unique set of challenge-response pairs for each IC, even if the digital IC functionality or masks of the ICs are exactly the same? We rely on there being enough statistical delay variation for equivalent wires and devices across different ICs. Sources of statistical variation in manufacturing are well documented in the literature (e.g., [5] and [6]) and statistical variation has been exploited to create IC identification circuits that generate a single unique response for each manufactured IC [11]. The *transient* response of the IC to a challenge, i.e., input stimulus,

*This work was funded by Acer Inc., Delta Electronics Inc., HP Corp., NTT Inc., Nokia Research Center, and Philips Research under the MIT Project Oxygen partnership.
†Visiting researcher from Philips Research, Prof Holstlaan 4, Eindhoven, The Netherlands.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
CCS'02 November 18–22, 2002, Washington, DC, USA.
Copyright 2002 ACM 1-58113-612-9/02/0011 ...$5.00.

[1] Gassend, B., Clarke, D., Van Dijk, M. and Devadas, S., 2002, November. Silicon physical random functions. In Proceedings of the 9th ACM conference on Computer and communications security (pp. 148-160). ACM.
[2] Gassend, B., Lim, D., Clarke, D., Van Dijk, M. and Devadas, S., 2004. Identification and authentication of integrated circuits. Concurrency and Computation: Practice and Experience, 16(11), pp.1077-1098.
[3] Herder, C., Ren, L., van Dijk, M., Yu, M.D.M. and Devadas, S., 2017. Trapdoor computational fuzzy extractors and stateless cryptographically-secure physical unclonable functions. IEEE Transactions on Dependable and Secure Computing, 14(1), pp.65-82.

Machine learning (ML) attacks against PUFs

Silicon Physical Random Functions·

Blaise Gassend, Dwaine Clarke, Marten van Dijk† and Srinivas Devadas
Massachusetts Institute of Technology
Laboratory for Computer Science
Cambridge, MA 02139, USA
{gassend,declarke,marten,devadas}@mit.edu

ABSTRACT

We introduce the notion of a Physical Random Function (PUF). We argue that a complex integrated circuit can be viewed as a silicon PUF and describe a technique to identify and authenticate individual integrated circuits (ICs).

We describe several possible circuit realizations of different PUFs. These circuits have been implemented in commodity Field Programmable Gate Arrays (FPGAs). We present experiments which indicate that reliable authentication of individual FPGAs can be performed even in the presence of significant environmental variations.

We describe how secure smart cards can be built, and also briefly describe how PUFs can be applied to licensing and certification applications.

Categories and Subject Descriptors

C.3 [**Special-Purpose and Application-Based Systems**]: Smartcards

General Terms

Measurement, Experimentation, Security

Keywords

Identification, physical random function, physical security, smartcard, tamper resistance, unclonability

1. INTRODUCTION

We describe the notion of Physical Random Functions (PUFs) and argue that PUFs can be implemented using conventional integrated circuit (IC) design techniques. This leads us to a method of identifying and authenticating individual ICs and a means of building secure smartcards. A host of other applications are also possible.

Many methods are already available to identify and authenticate ICs. One can embed a unique identifier in an IC to give it a unique identity. This approach can identify the IC, but cannot authenticate it. To enable authentication, one needs to embed a secret key onto the IC. Of course, for the system to work, this key needs to remain secret, which means that the packaged IC has to be made resistant to attacks that attempt to discover the key. Numerous attacks are described in the literature. These attacks may be invasive, e.g., removal of the package and layers of the IC, or non-invasive, e.g., differential power analysis that attempts to determine the key by stimulating the IC and observing the power and ground rails. Making an IC tamper-resistant to all forms of attacks is a challenging problem and is receiving some attention [1]. IBM's PCI Cryptographic Coprocessor encapsulates a 486-class processing subsystem within a tamper-sensing and tamper-responding environment where one can run security-sensitive processes [13]. However, providing high-grade tamper resistance, which makes it impossible for an attacker to access or modify the secrets held inside a device, is expensive and difficult [2, 3].

We propose a completely different approach to IC authentication in this paper. Our thesis is that there is enough manufacturing process variations across ICs with identical masks to uniquely characterize each IC, and this characterization can be performed with a large signal-to-noise ratio (SNR). The characterization of an IC involves the generation of a set of challenge-response pairs. To authenticate ICs we require the set of challenge-response pairs to be characteristic of each IC. For reliable authentication, we require that environmental variations and measurement errors do not produce so much noise that they hide inter-IC variations. We will show in this paper, using experiments and analysis, that we can perform reliable authentication using the techniques that we now introduce.

How can we produce a unique set of challenge-response pairs for each IC, even if the digital IC functionality or masks of the ICs are exactly the same? We rely on there being enough statistical delay variation for equivalent wires and devices across different ICs. Sources of statistical variation in manufacturing are well documented in the literature (e.g., [5] and [6]) and statistical variation has been exploited to create IC identification circuits that generate a single unique response for each manufactured IC [11]. The *transient* response of the IC to a challenge, i.e., input stimulus,

*This work was funded by Acer Inc., Delta Electronics Inc., HP Corp., NTT Inc., Nokia Research Center, and Philips Research under the MIT Project Oxygen partnership.
†Visiting researcher from Philips Research, Prof Holstlaan 4, Eindhoven, The Netherlands.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
CCS'02 November 18–22, 2002, Washington, DC, USA.
Copyright 2002 ACM 1-58113-612-9/02/0011 ...$5.00.

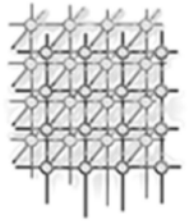
- **"If the adversary can learn the entire set of challenge-response pairs, he can create a model of a counterfeit IC" [1].**

[1] Gassend, B., Clarke, D., Van Dijk, M. and Devadas, S., 2002, November. Silicon physical random functions. In Proceedings of the 9th ACM conference on Computer and communications security (pp. 148-160). ACM.
[2] Gassend, B., Lim, D., Clarke, D., Van Dijk, M. and Devadas, S., 2004. Identification and authentication of integrated circuits. Concurrency and Computation: Practice and Experience, 16(11), pp.1077-1098.
[3] Herder, C., Ren, L., van Dijk, M., Yu, M.D.M. and Devadas, S., 2017. Trapdoor computational fuzzy extractors and stateless cryptographically-secure physical unclonable functions. IEEE Transactions on Dependable and Secure Computing, 14(1), pp.65-82.

Machine learning (ML) attacks against PUFs

CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE
Concurrency Computat.: Pract. Exper. 2003; **3**:1–20 *Prepared using* cpeauth.cls *[Version: 2002/09/19 v2.02]*

Identification and Authentication of Integrated Circuits



Blaise Gassend[1], Dwaine Clarke[1], Daihyun Lim[1]
Marten van Dijk[2], Srinivas Devadas[1*]

[1] *Massachusetts Institute of Technology, Laboratory for Computer Science, Cambridge, MA 02139, USA*
[2] *Prof Holstlaan 4, Eindhoven, The Netherlands*

SUMMARY

This paper describes a technique to reliably and securely identify individual integrated circuits (ICs) based on the precise measurement of circuit delays and a simple challenge-response protocol. This technique could be used to produce key-cards that are more difficult to clone than ones involving digital keys on the IC. We consider potential venues of attack against our system, and present candidate implementations. Experiments on Field Programmable Gate Arrays show that the technique is viable. Finally, we analyze the difficulty of breaking the system in an idealized additive delay model.

KEY WORDS: Physical random function, physical security, smartcard, tamper resistance, unclonability

1. Introduction

We describe a technique to identify and authenticate arbitrary integrated circuits (IC's) based on a prior *delay* characterization of the IC. While IC's can be reliably mass-manufactured to have identical digital logic functionality, the premise of our approach is that each IC is unique in its delay characteristics due to inherent variations in manufacturing across different dies, wafers, and processes. While digital logic functionality relies on timing constraints being met, different ICs with the exact same digital functionality will have unique behaviors when these constraints are not met, because their delay characteristics are different.

Researchers have proposed the addition of specific circuits that produce unique responses due to manufacturing variations in IC's such that these IC's can be identified (e.g., [12]).

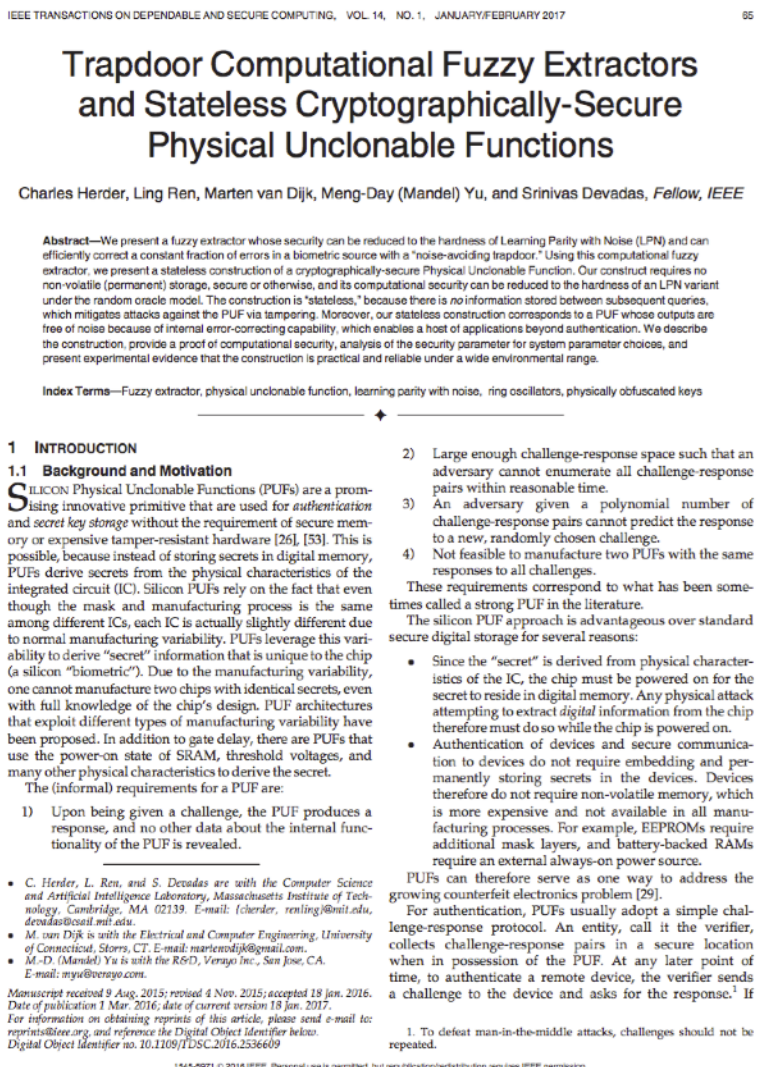
*Correspondence to: Massachusetts Institute of Technology, Laboratory for Computer Science, Cambridge, MA 02139, USA

Copyright © 2003 John Wiley & Sons, Ltd.

- **"If the adversary can learn the entire set of challenge-response pairs, he can create a model of a counterfeit IC" [1].**
- **"[some PUF] circuits are not difficult enough to model, contrarily to what we had conjectured [before]" [2].**

[1] Gassend, B., Clarke, D., Van Dijk, M. and Devadas, S., 2002, November. Silicon physical random functions. In Proceedings of the 9th ACM conference on Computer and communications security (pp. 148-160). ACM.
[2] Gassend, B., Lim, D., Clarke, D., Van Dijk, M. and Devadas, S., 2004. Identification and authentication of integrated circuits. Concurrency and Computation: Practice and Experience, 16(11), pp.1077-1098.
[3] Herder, C., Ren, L., van Dijk, M., Yu, M.D.M. and Devadas, S., 2017. Trapdoor computational fuzzy extractors and stateless cryptographically-secure physical unclonable functions. IEEE Transactions on Dependable and Secure Computing, 14(1), pp.65-82.

Machine learning (ML) attacks against PUFs



IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, VOL. 14, NO. 1, JANUARY/FEBRUARY 2017 65

Trapdoor Computational Fuzzy Extractors and Stateless Cryptographically-Secure Physical Unclonable Functions

Charles Herder, Ling Ren, Marten van Dijk, Meng-Day (Mandel) Yu, and Srinivas Devadas, *Fellow, IEEE*

Abstract—We present a fuzzy extractor whose security can be reduced to the hardness of Learning Parity with Noise (LPN) and can efficiently correct a constant fraction of errors in a biometric source with a "noise-avoiding trapdoor." Using this computational fuzzy extractor, we present a stateless construction of a cryptographically-secure Physical Unclonable Function. Our construct requires no non-volatile (permanent) storage, secure or otherwise, and its computational security can be reduced to the hardness of an LPN variant under the random oracle model. The construction is "stateless," because there is *no* information stored between subsequent queries, which mitigates attacks against the PUF via tampering. Moreover, our stateless construction corresponds to a PUF whose outputs are free of noise because of internal error-correcting capability, which enables a host of applications beyond authentication. We describe the construction, provide a proof of computational security, analysis of the security parameter for system parameter choices, and present experimental evidence that the construction is practical and reliable under a wide environmental range.

Index Terms—Fuzzy extractor, physical unclonable function, learning parity with noise, ring oscillators, physically obfuscated keys

1 Introduction

1.1 Background and Motivation

Silicon Physical Unclonable Functions (PUFs) are a promising innovative primitive that are used for *authentication* and *secret key storage* without the requirement of secure memory or expensive tamper-resistant hardware [26], [53]. This is possible, because instead of storing secrets in digital memory, PUFs derive secrets from the physical characteristics of the integrated circuit (IC). Silicon PUFs rely on the fact that even though the mask and manufacturing process is the same among different ICs, each IC is actually slightly different due to normal manufacturing variability. PUFs leverage this variability to derive "secret" information that is unique to the chip (a silicon "biometric"). Due to the manufacturing variability, one cannot manufacture two chips with identical secrets, even with full knowledge of the chip's design. PUF architectures that exploit different types of manufacturing variability have been proposed. In addition to gate delay, there are PUFs that use the power-on state of SRAM, threshold voltages, and many other physical characteristics to derive the secret.

The (informal) requirements for a PUF are:

1) Upon being given a challenge, the PUF produces a response, and no other data about the internal functionality of the PUF is revealed.
2) Large enough challenge-response space such that an adversary cannot enumerate all challenge-response pairs within reasonable time.
3) An adversary given a polynomial number of challenge-response pairs cannot predict the response to a new, randomly chosen challenge.
4) Not feasible to manufacture two PUFs with the same responses to all challenges.

These requirements correspond to what has been sometimes called a strong PUF in the literature.

The silicon PUF approach is advantageous over standard secure digital storage for several reasons:

- Since the "secret" is derived from physical characteristics of the IC, the chip must be powered on for the secret to reside in digital memory. Any physical attack attempting to extract *digital* information from the chip therefore must do so while the chip is powered on.
- Authentication of devices and secure communication to devices do not require embedding and permanently storing secrets in the devices. Devices therefore do not require non-volatile memory, which is more expensive and not available in all manufacturing processes. For example, EEPROMs require additional mask layers, and battery-backed RAMs require an external always-on power source.

PUFs can therefore serve as one way to address the growing counterfeit electronics problem [29].

For authentication, PUFs usually adopt a simple challenge-response protocol. An entity, call it the verifier, collects challenge-response pairs in a secure location when in possession of the PUF. At any later point of time, to authenticate a remote device, the verifier sends a challenge to the device and asks for the response.[1] If

1. To defeat man-in-the-middle attacks, challenges should not be repeated.

- C. Herder, L. Ren, and S. Devadas are with the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139. E-mail: {cherder, renling}@mit.edu, devadas@csail.mit.edu.
- M. van Dijk is with the Electrical and Computer Engineering, University of Connecticut, Storrs, CT. E-mail: marlenvdijk@gmail.com.
- M.-D. (Mandel) Yu is with the R&D, Verayo Inc., San Jose, CA. E-mail: myu@verayo.com.

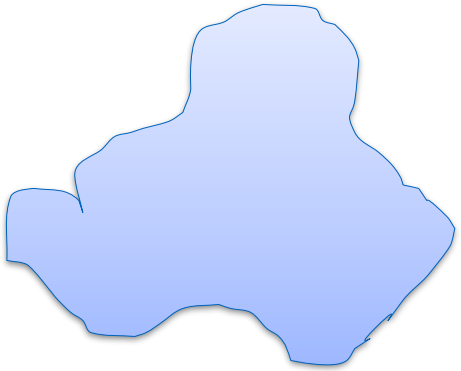
Manuscript received 9 Aug. 2015; revised 4 Nov. 2015; accepted 18 Jan. 2016. Date of publication 1 Mar. 2016; date of current version 18 Jan. 2017. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TDSC.2016.2536609

1545-5971 © 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

- **"If the adversary can learn the entire set of challenge-response pairs, he can create a model of a counterfeit IC" [1].**
- **"[some PUF] circuits are not difficult enough to model, contrarily to what we had conjectured [before]" [2].**
- **"Unfortunately, none of the candidate [PUF] constructions have a proof of computational security, and further, most, if not all, of them have been shown to be susceptible to ML attacks" [3].**

[1] Gassend, B., Clarke, D., Van Dijk, M. and Devadas, S., 2002, November. Silicon physical random functions. In Proceedings of the 9th ACM conference on Computer and communications security (pp. 148-160). ACM.
[2] Gassend, B., Lim, D., Clarke, D., Van Dijk, M. and Devadas, S., 2004. Identification and authentication of integrated circuits. Concurrency and Computation: Practice and Experience, 16(11), pp.1077-1098.
[3] Herder, C., Ren, L., van Dijk, M., Yu, M.D.M. and Devadas, S., 2017. Trapdoor computational fuzzy extractors and stateless cryptographically-secure physical unclonable functions. IEEE Transactions on Dependable and Secure Computing, 14(1), pp.65-82.

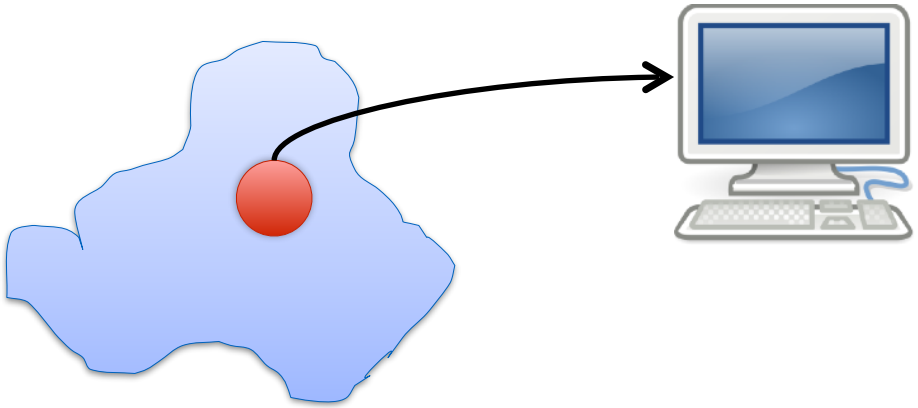
ML algorithms



- **Empirical learning approaches**
  - **No predefined levels of accuracy and confidence**

[1] Kearns, M.J. and Vazirani, U.V., 1994. An introduction to computational learning theory. MIT press.

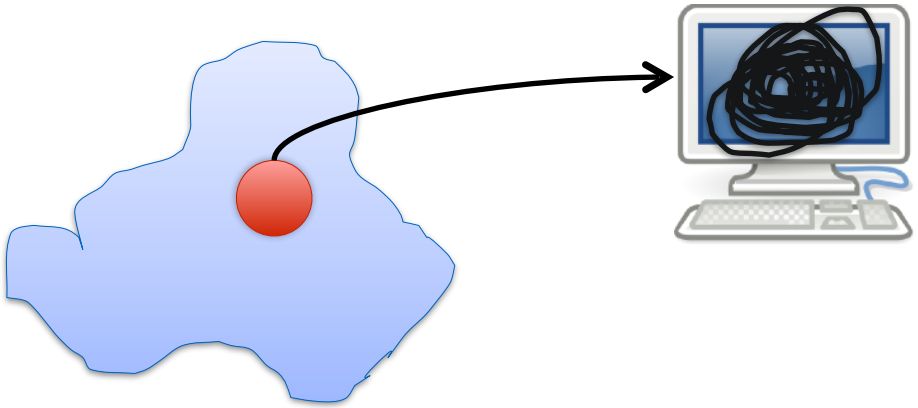
ML algorithms



- **Empirical learning approaches**
  - **No predefined levels of accuracy and confidence**

[1] Kearns, M.J. and Vazirani, U.V., 1994. An introduction to computational learning theory. MIT press.

ML algorithms



- **Empirical learning approaches**
  - **No predefined levels of accuracy and confidence**

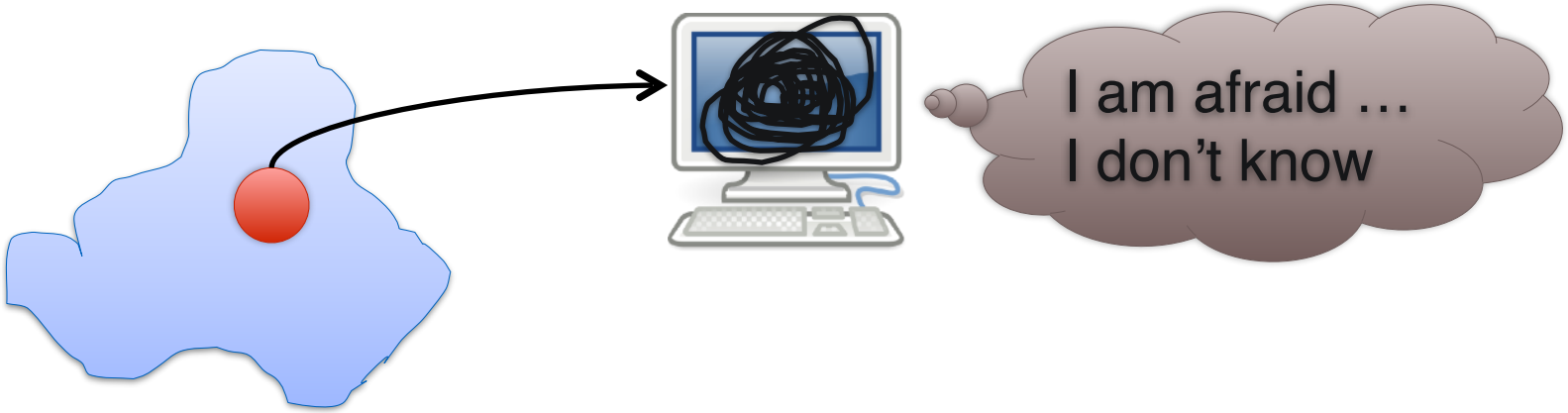
[1] Kearns, M.J. and Vazirani, U.V., 1994. An introduction to computational learning theory. MIT press.

ML algorithms

- **Empirical learning approaches**
  - **No predefined levels of accuracy and confidence**

[1] Kearns, M.J. and Vazirani, U.V., 1994. An introduction to computational learning theory. MIT press.

ML algorithms



- **Empirical learning approaches**
  - **No predefined levels of accuracy and confidence**

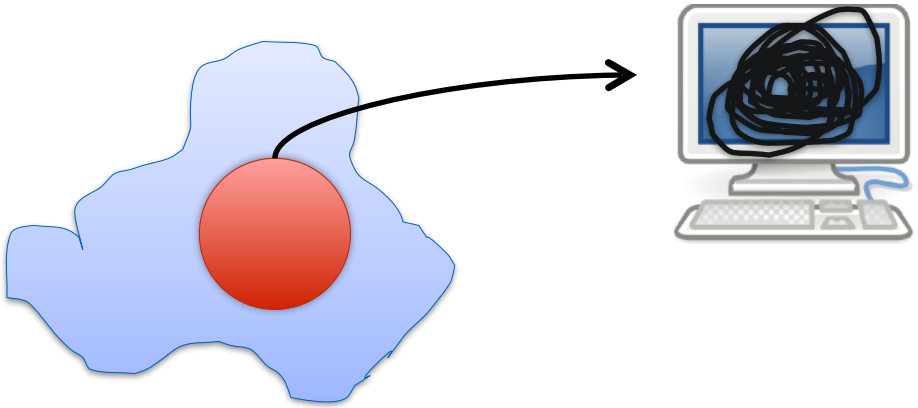
[1] Kearns, M.J. and Vazirani, U.V., 1994. An introduction to computational learning theory. MIT press.

ML algorithms

- **Empirical learning approaches**
  - **No predefined levels of accuracy and confidence**

[1] Kearns, M.J. and Vazirani, U.V., 1994. An introduction to computational learning theory. MIT press.

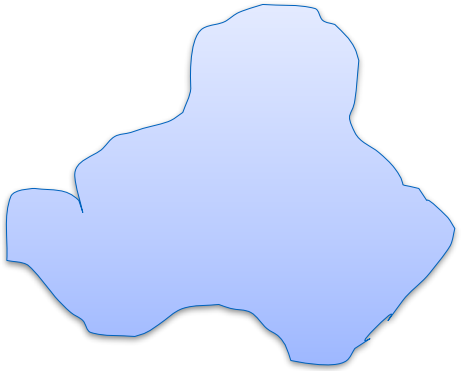
ML algorithms



- Empirical learning approaches
  - No predefined levels of accuracy and confidence
- Probably Approximately Correct (PAC) learning approaches [1]
  - For given levels of accuracy and confidence

[1] Kearns, M.J. and Vazirani, U.V., 1994. An introduction to computational learning theory. MIT press.

ML algorithms



- Empirical learning approaches
  - No predefined levels of accuracy and confidence
- Probably Approximately Correct (PAC) learning approaches [1]
  - For given levels of accuracy and confidence

[1] Kearns, M.J. and Vazirani, U.V., 1994. An introduction to computational learning theory. MIT press.

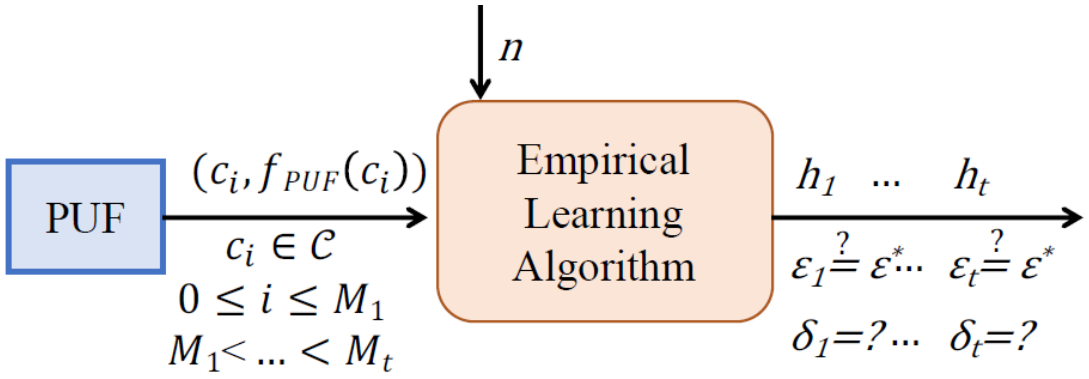
ML algorithms

- Empirical learning approaches
  - No predefined levels of accuracy and confidence
- Probably Approximately Correct (PAC) learning approaches [1]
  - For given levels of accuracy and confidence

[1] Kearns, M.J. and Vazirani, U.V., 1994. An introduction to computational learning theory. MIT press.

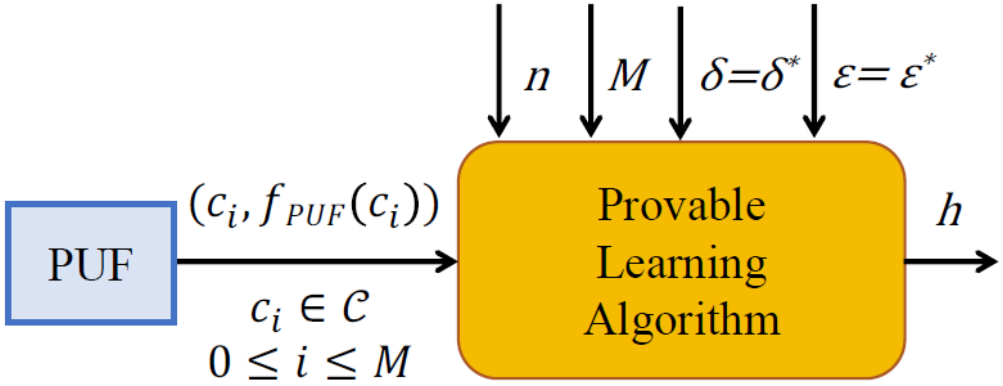
Empirical vs. Provable ML (1)

- Collect varying number of challenge-response pairs (CRPs) at random
  - Employ ML algorithms for each set of CRPs in a plug-and-play fashion
- Drawbacks
  - Assessment is algorithm, parameter, and instance dependent with no convergence guarantees
  - Standardization and comparison infeasible

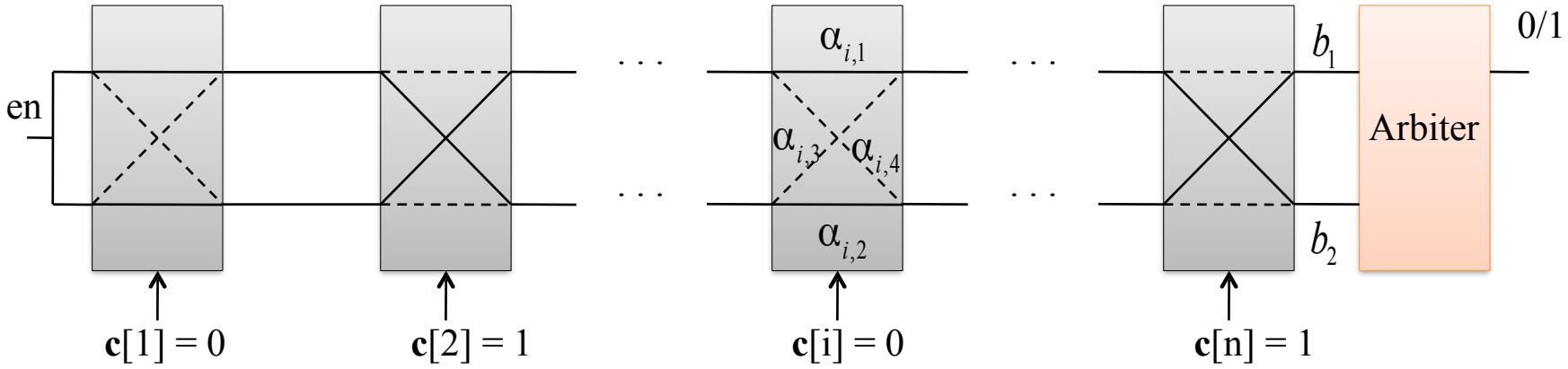


Empirical vs. Provable ML (2)

- Takes desired number of CRPs (M), accuracy (ε), and confidence (δ) for ML as input parameters
- (May) Adaptively requests specific CRPs
- Main Features
  - Provably determines if PUF is learnable by any polynomial ML algorithm for specified input parameters



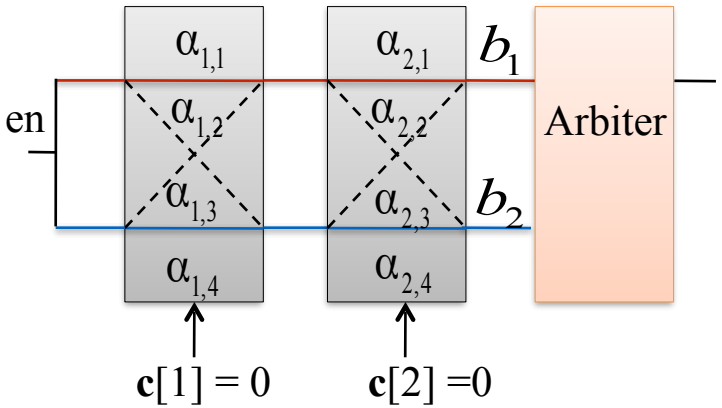
Arbiter PUFs



- **The security of Arbiter PUFs [1] is relying on an assumption:**
  - **The attacker cannot measure the delays in each stage**

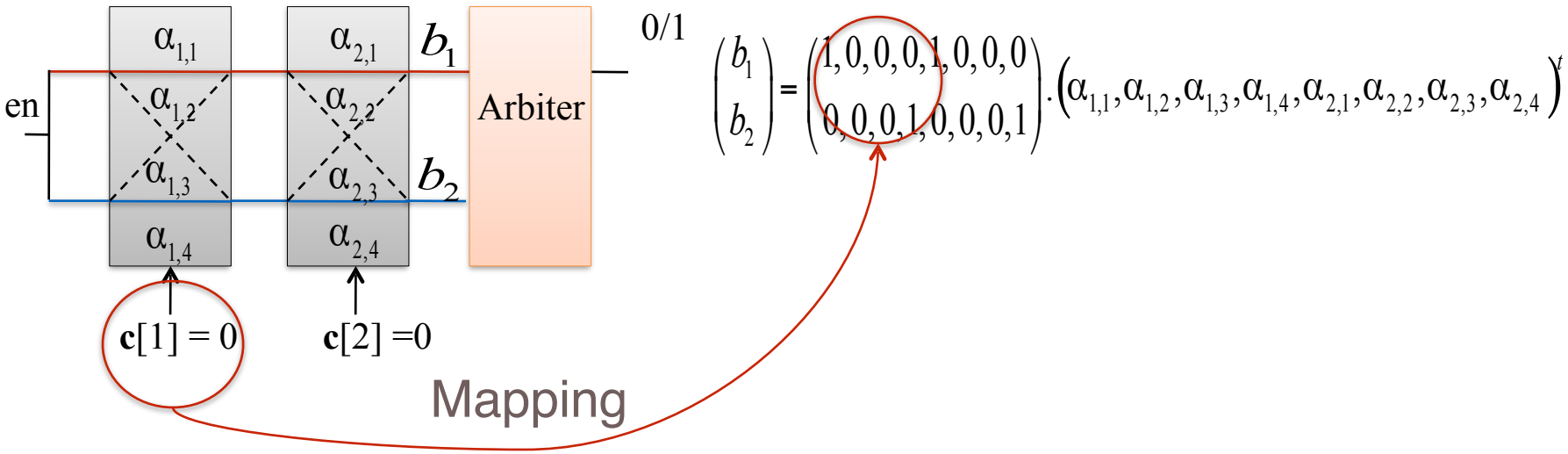
[1] Lee, J.W., Lim, D., Gassend, B., Suh, G.E., Van Dijk, M. and Devadas, S., 2004, June. A technique to build a secret key in integrated circuits for identification and authentication applications. In VLSI Circuits, 2004. Digest of Technical Papers. 2004 Symposium on (pp. 176-179). IEEE.

Arbiter PUF and its linear behavior

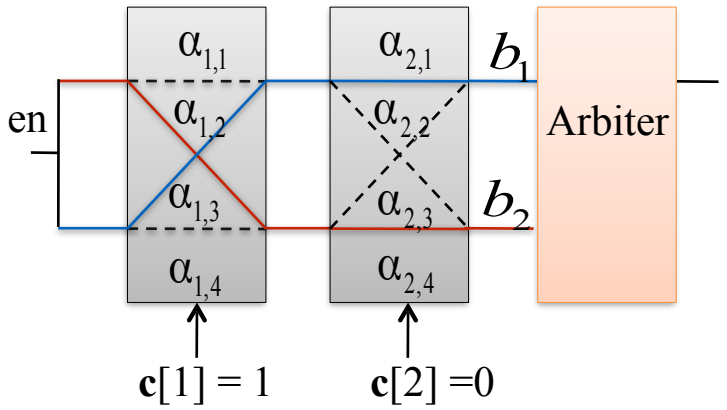


$$\begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} 1,0,0,0,1,0,0,0 \\ 0,0,0,1,0,0,0,1 \end{pmatrix} \cdot \left(\alpha_{1,1},\alpha_{1,2},\alpha_{1,3},\alpha_{1,4},\alpha_{2,1},\alpha_{2,2},\alpha_{2,3},\alpha_{2,4}\right)^t$$

Arbiter PUF and its linear behavior

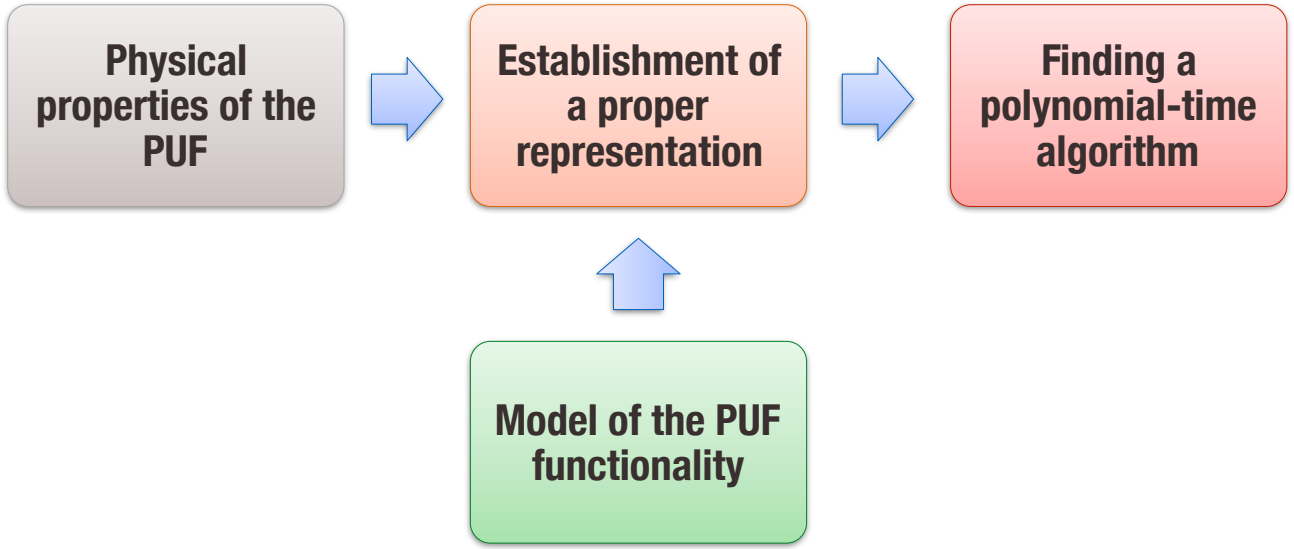


Arbiter PUF and its linear behavior



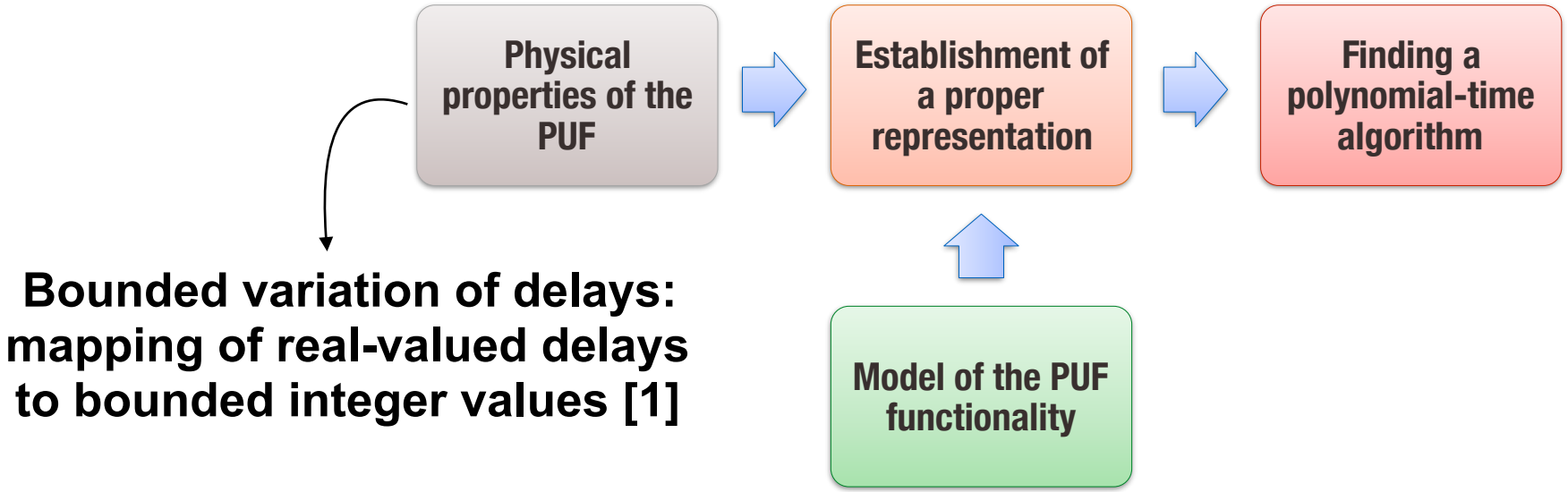
$$\begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} 0,0,1,0,1,0,0,0 \\ 0,1,0,0,0,0,0,1 \end{pmatrix} \cdot \left(\alpha_{1,1},\alpha_{1,2},\alpha_{1,3},\alpha_{1,4},\alpha_{2,1},\alpha_{2,2},\alpha_{2,3},\alpha_{2,4}\right)^t$$

An Example: Provable learning of Arbiter PUFs



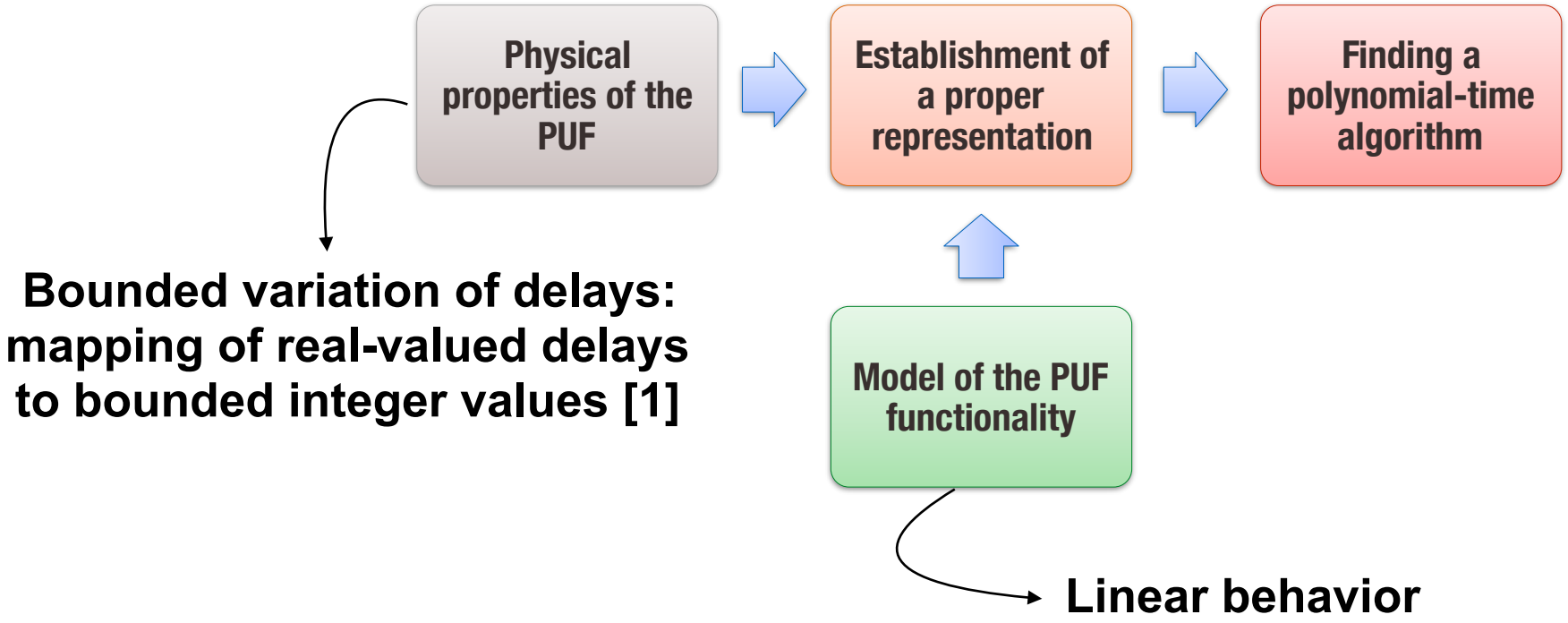
[1] Ganji, F., Tajik, S. and Seifert, J.P., 2016. PAC learning of arbiter PUFs. Journal of Cryptographic Engineering, 6(3), pp.249-258.
[2] Angluin, D., 1987. Learning regular sets from queries and counterexamples. Information and computation, 75(2), pp.87-106.

An Example: Provable learning of Arbiter PUFs



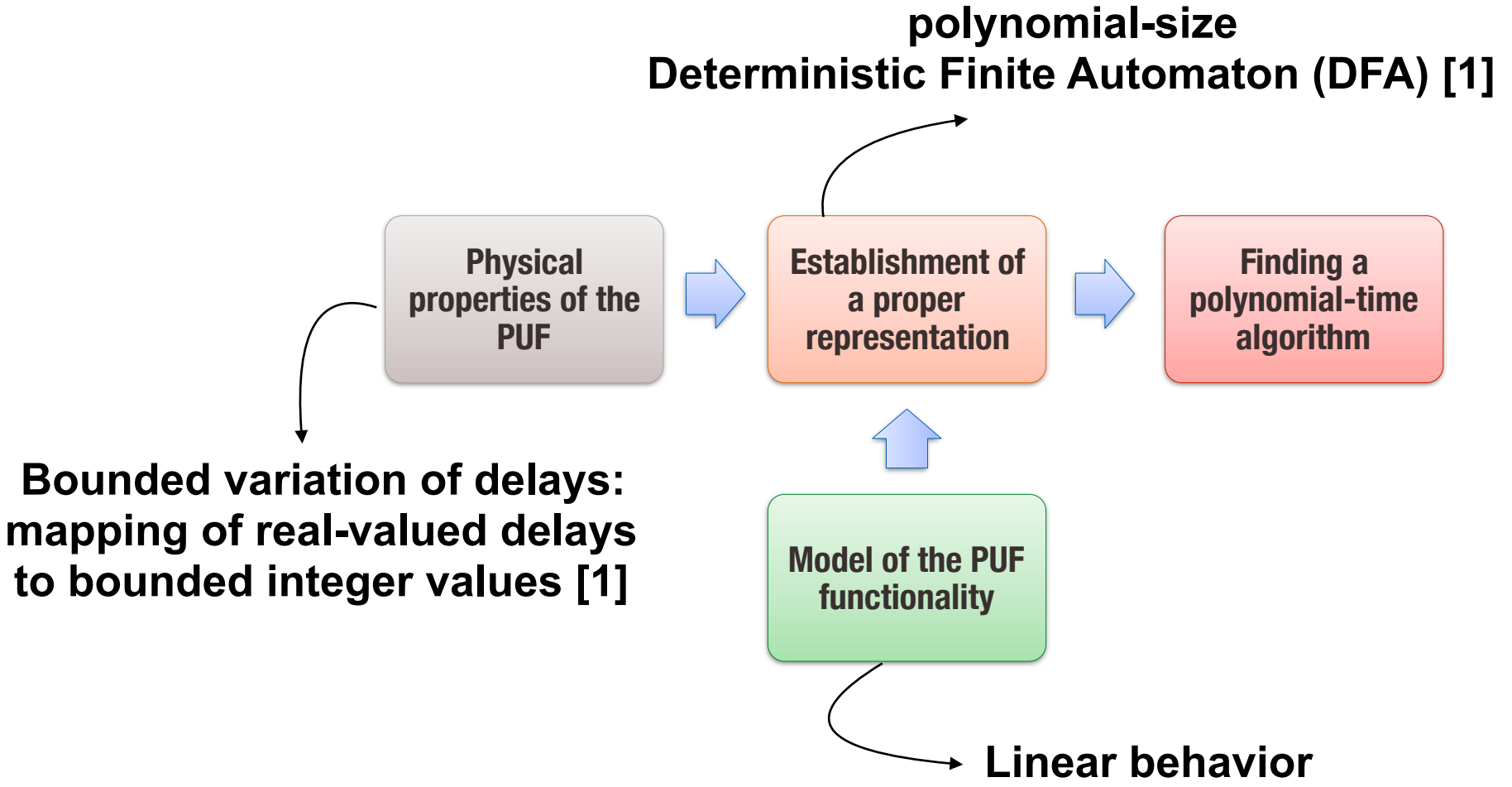
[1] Ganji, F., Tajik, S. and Seifert, J.P., 2016. PAC learning of arbiter PUFs. Journal of Cryptographic Engineering, 6(3), pp.249-258.
[2] Angluin, D., 1987. Learning regular sets from queries and counterexamples. Information and computation, 75(2), pp.87-106.

An Example: Provable learning of Arbiter PUFs



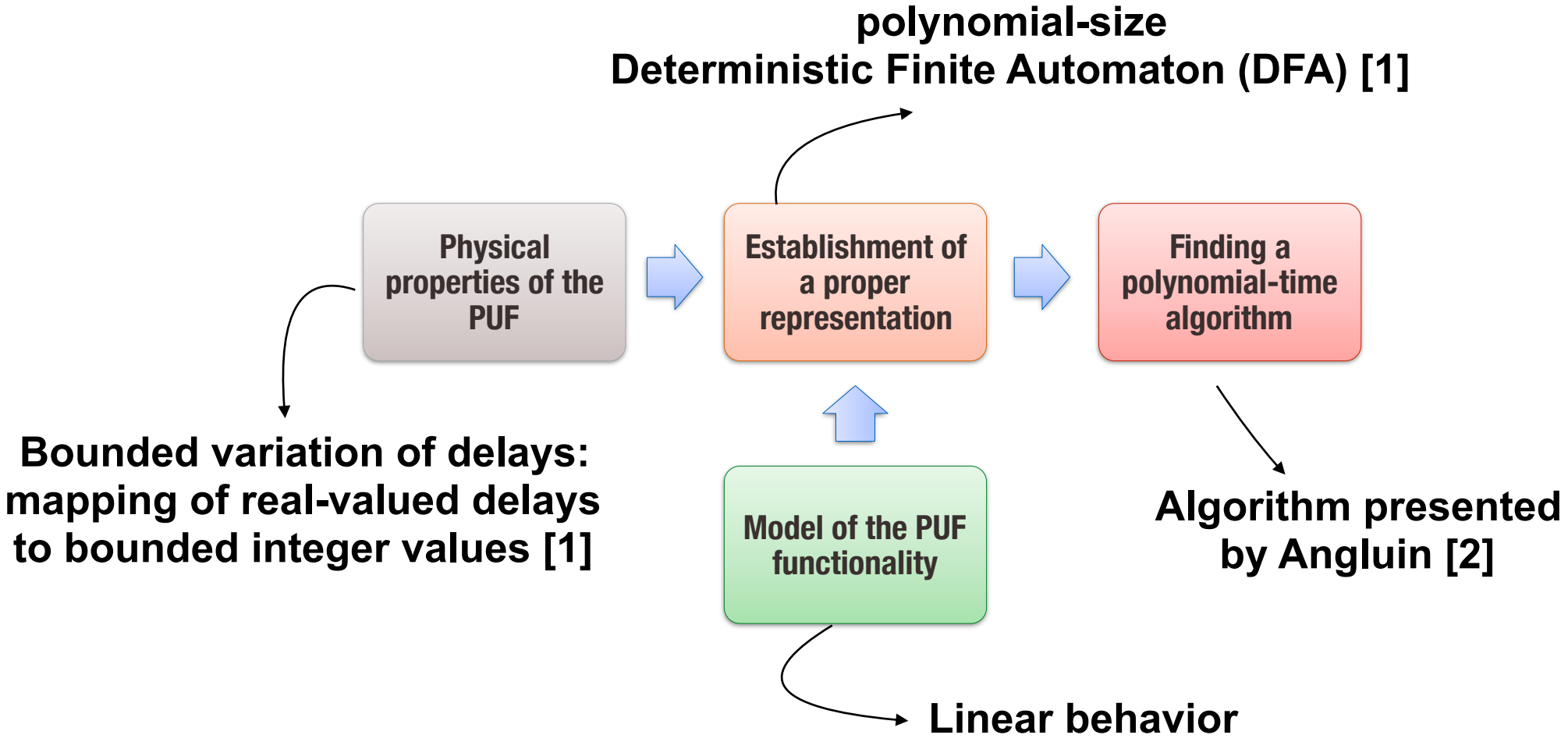
[1] Ganji, F., Tajik, S. and Seifert, J.P., 2016. PAC learning of arbiter PUFs. Journal of Cryptographic Engineering, 6(3), pp.249-258.
[2] Angluin, D., 1987. Learning regular sets from queries and counterexamples. Information and computation, 75(2), pp.87-106.

An Example: Provable learning of Arbiter PUFs



[1] Ganji, F., Tajik, S. and Seifert, J.P., 2016. PAC learning of arbiter PUFs. Journal of Cryptographic Engineering, 6(3), pp.249-258.
[2] Angluin, D., 1987. Learning regular sets from queries and counterexamples. Information and computation, 75(2), pp.87-106.

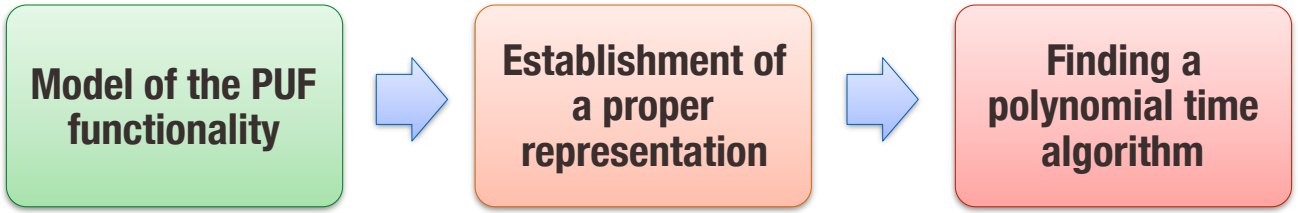
An Example: Provable learning of Arbiter PUFs



[1] Ganji, F., Tajik, S. and Seifert, J.P., 2016. PAC learning of arbiter PUFs. Journal of Cryptographic Engineering, 6(3), pp.249-258.
[2] Angluin, D., 1987. Learning regular sets from queries and counterexamples. Information and computation, 75(2), pp.87-106.

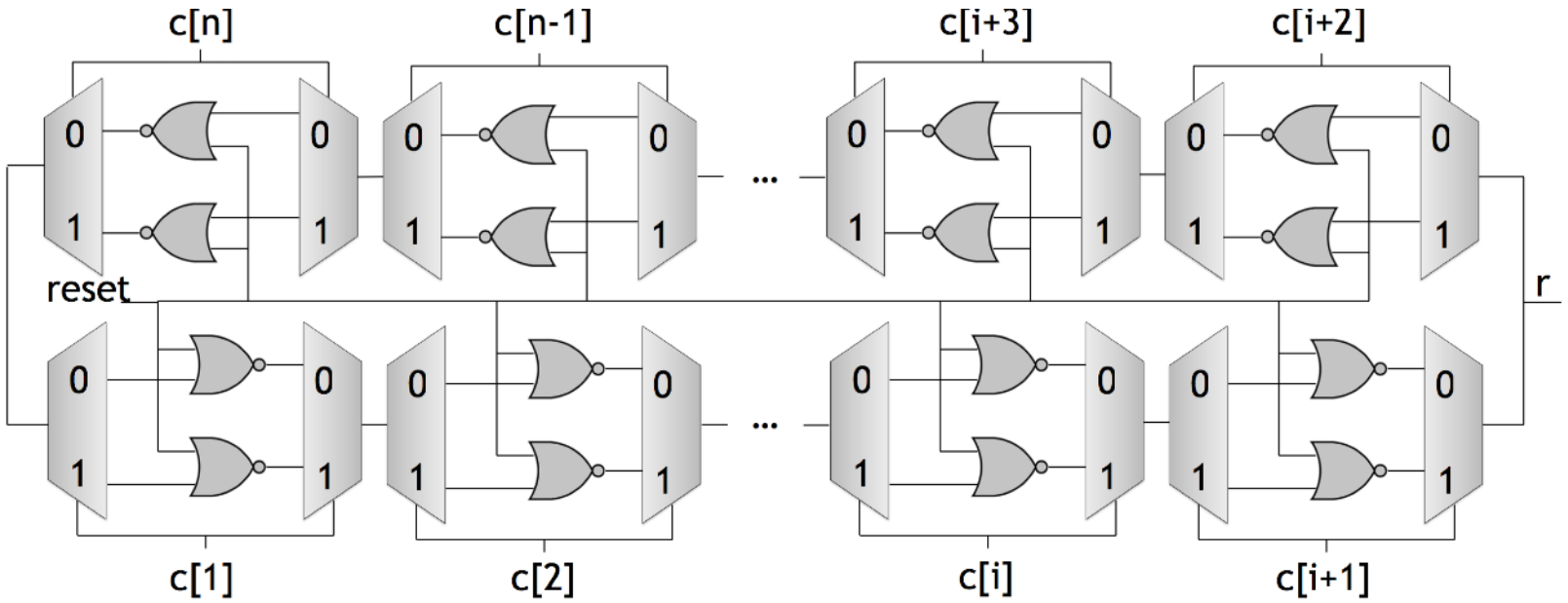
What helps an adversary…

- **Linear behavior of Arbiter PUFs: an example of the model representing the internal functionality of the respective PUF**



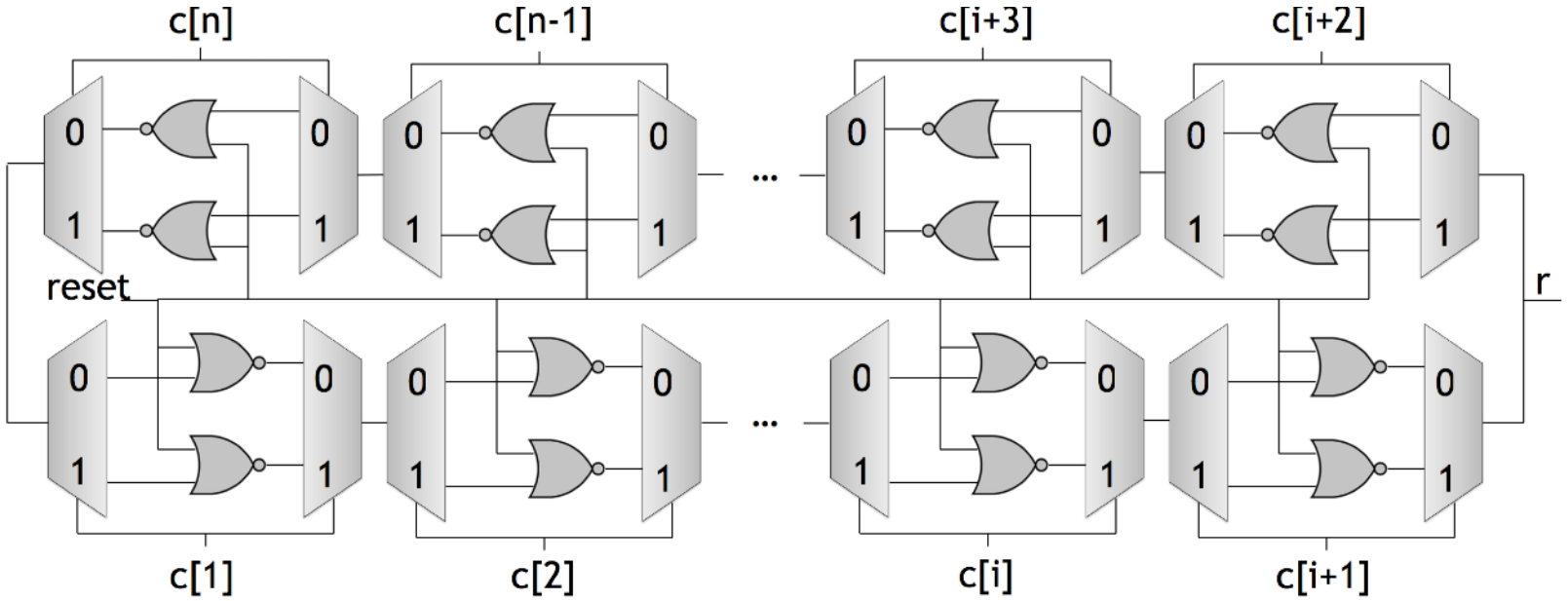
- **What happens if this model is unknown?**
  - **Prime example: Bistable Ring (BR) PUFs**

BR PUFs

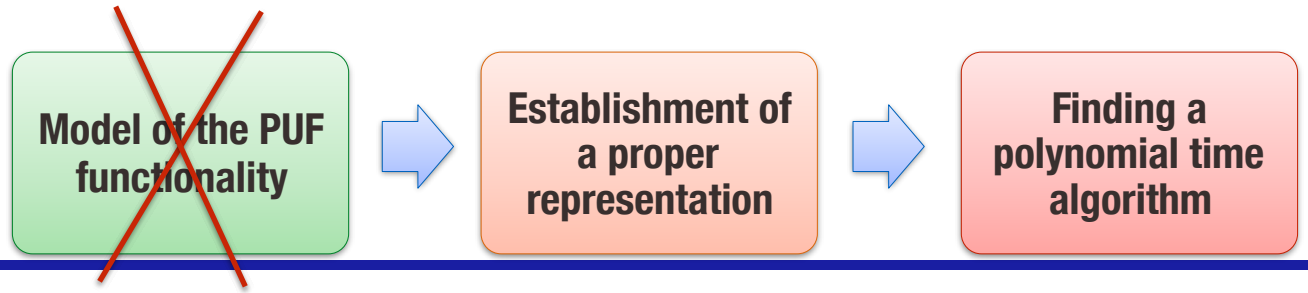


- **No precise mathematical model of the BR PUF functionality**

BR PUFs



- **No precise mathematical model of the BR PUF functionality**



Existence of Influential Bits

- **Unequal influence of challenge bit positions on the respective responses**

Existence of Influential Bits

- **Unequal influence of challenge bit positions on the respective responses**

How many influential bits?

Existence of Influential Bits

- **Unequal influence of challenge bit positions on the respective responses**

Existence of Influential Bits

- Unequal influence of challenge bit positions on the respective responses

- Determined by the notion of the average sensitivity $I(f_{PUF})$

Existence of Influential Bits

- Unequal influence of challenge bit positions on the respective responses

- Determined by the notion of the average sensitivity $I(f_{PUF})$
  - c_1:randomly, uniformly chosen challenge

Existence of Influential Bits

- Unequal influence of challenge bit positions on the respective responses

- Determined by the notion of the average sensitivity $I(f_{PUF})$
  - c_1:randomly, uniformly chosen challenge

Existence of Influential Bits

- Unequal influence of challenge bit positions on the respective responses

- Determined by the notion of the average sensitivity $I(f_{PUF})$
  - c_1:randomly, uniformly chosen challenge

Existence of Influential Bits

- Unequal influence of challenge bit positions on the respective responses

- Determined by the notion of the average sensitivity $I(f_{PUF})$
  - c_1:randomly, uniformly chosen challenge

Existence of Influential Bits

- Unequal influence of challenge bit positions on the respective responses

- Determined by the notion of the average sensitivity $I(f_{PUF})$
  - c_1:randomly, uniformly chosen challenge

$$I(f_{PUF}) := \sum_{i=1}^{n} \Pr[r_2 \neq r_1]$$

c	r
$c=1 \ldots 00$	1
$c'=1 \ldots 11$	1
$c''=1 \ldots 01$	

- Example of a *1*-junta
- K-junta learning: finding the relevant coordinates
  - Algorithm presented by, e.g., Angluin

c	r
$c=1 \ldots 00$	1
$c'=1 \ldots 11$	1
$c''=1 \ldots 01$	1

- Example of a *1*-junta
- K-junta learning: finding the relevant coordinates
  - Algorithm presented by, e.g., Angluin

Learning juntas

c	r
$c=1 \ldots 00$	1
$c'=1 \ldots 11$	1
$c''=1 \ldots 01$	1

- Example of a *1*-junta
- K-junta learning: finding the relevant coordinates
  - Algorithm presented by, e.g., Angluin

Learning juntas

c	r
$c=1 \ldots 00$	1
$c'=1 \ldots 11$	1
$c''=1 \ldots 01$	1

- Example of a *1*-junta
- K-junta learning: finding the relevant coordinates
  - Algorithm presented by, e.g., Angluin

Is ‘K’ a constant value?

What we know about BR PUFs

What we know about BR PUFs

- **Practical observations**

What we know about BR PUFs

- **Practical observations**
  - **Statistical analysis of 64-bit BR-PUFs: 5 influential bits [17]**

What we know about BR PUFs

- Practical observations
  - Statistical analysis of 64-bit BR-PUFs: 5 influential bits [17]
    - Our experiments on 64-bit BR PUFs implemented on Altera Cyclone IV FPGAs: 7 influential bits

What we know about BR PUFs

- Practical observations
  - Statistical analysis of 64-bit BR-PUFs: 5 influential bits [17]
    - Our experiments on 64-bit BR PUFs implemented on Altera Cyclone IV FPGAs: 7 influential bits
- Mathematical, more precise observation

What we know about BR PUFs

- Practical observations
  - Statistical analysis of 64-bit BR-PUFs: 5 influential bits [17]
    - Our experiments on 64-bit BR PUFs implemented on Altera Cyclone IV FPGAs: 7 influential bits
- Mathematical, more precise observation
  - K-junta testing: determining whether the function f_{PUF} is involved in the class of K-junta functions, e.g.,

What we know about BR PUFs

- Practical observations
  - Statistical analysis of 64-bit BR-PUFs: 5 influential bits [17]
    - Our experiments on 64-bit BR PUFs implemented on Altera Cyclone IV FPGAs: 7 influential bits
- Mathematical, more precise observation
  - K-junta testing: determining whether the function f_{PUF} is involved in the class of K-junta functions, e.g.,
    - for 64-bit BR PUFs, K=7

Fourier analysis: An example

$f_{Max2}: \{-1,1\}^2 \rightarrow \mathbb{R}$

c_1	c_2	$Max_2(c_1,c_2)$
1	1	1
1	-1	1
-1	1	1
-1	-1	-1

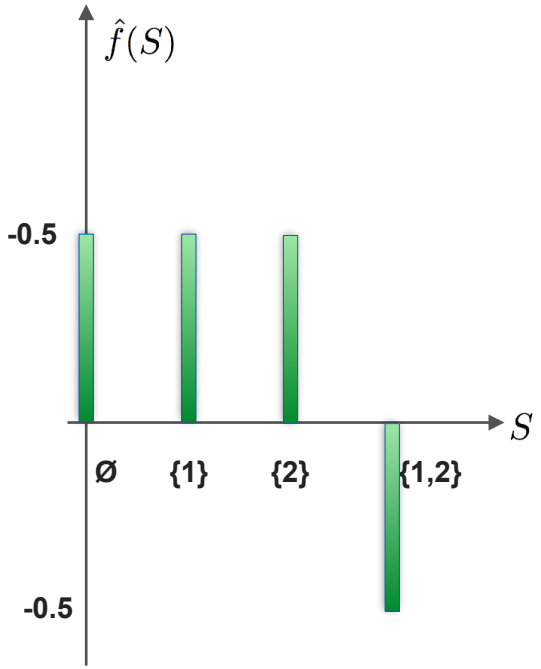
Fourier analysis: An example

f_{Max2}: $\{-1,1\}^2 \rightarrow \mathbb{R}$

c_1	c_2	$Max_2(c_1,c_2)$
1	1	1
1	-1	1
-1	1	1
-1	-1	-1

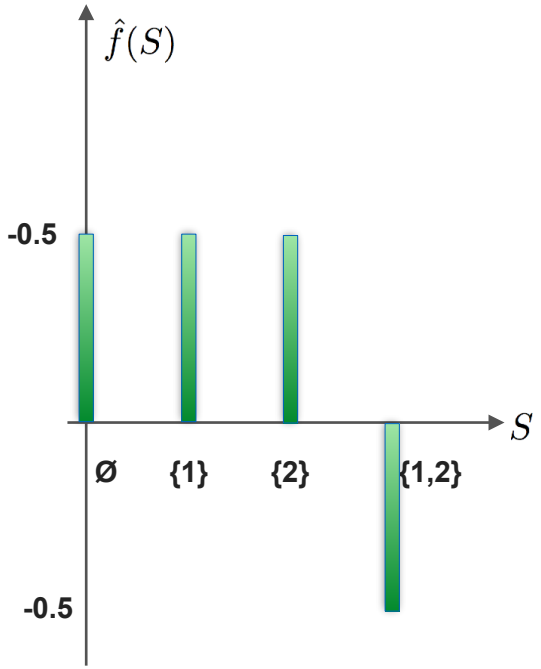
$$\hat{f}(S) := \mathbf{E}_{c \in \mathcal{U}}[f(c)\chi_S(c)]$$

$$\chi_S(c) := \prod_{i \in S} c_i$$



Fourier analysis: An example

f_{Max2}: $\{-1,1\}^2 \rightarrow \mathbb{R}$

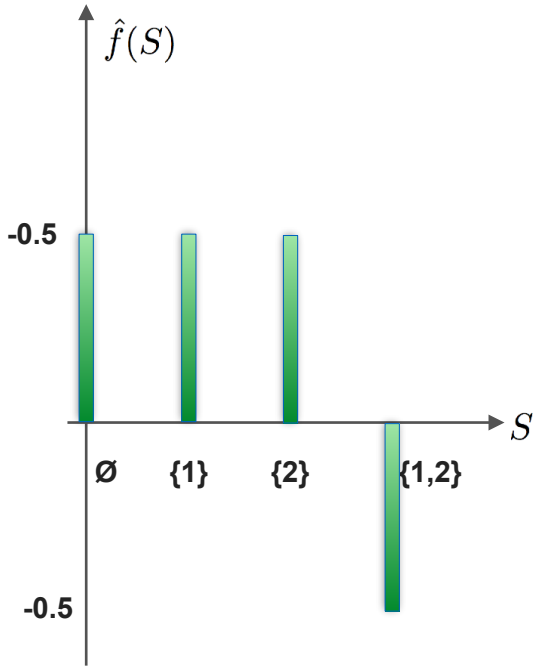


Fourier analysis: An example

f_{Max2}: $\{-1,1\}^2 \rightarrow \mathbb{R}$

c_1	c_2	$Max_2(c_1,c_2)$
1	1	1
1	-1	1
-1	1	1
-1	-1	-1

$$f(c) = \sum_{S \subseteq [n]} \hat{f}(S)\chi_S(c)$$

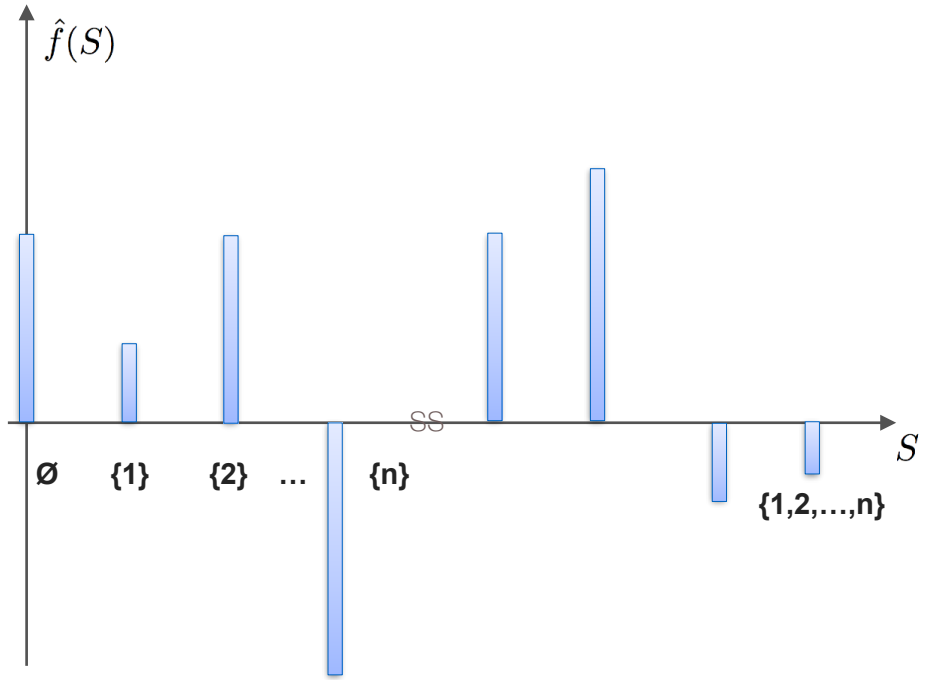


Approximation (or cutting the tail!)

$f: \{-1,1\}^n \rightarrow \mathbb{R}$

M

c_1	c_2	c_3	…	c_n	$f(c_1,c_2,\ldots,c_n)$
1	1	1	…	1	1
1	-1	1			
-1	1	1			
.					
.					
.					
-1	-1	-1			

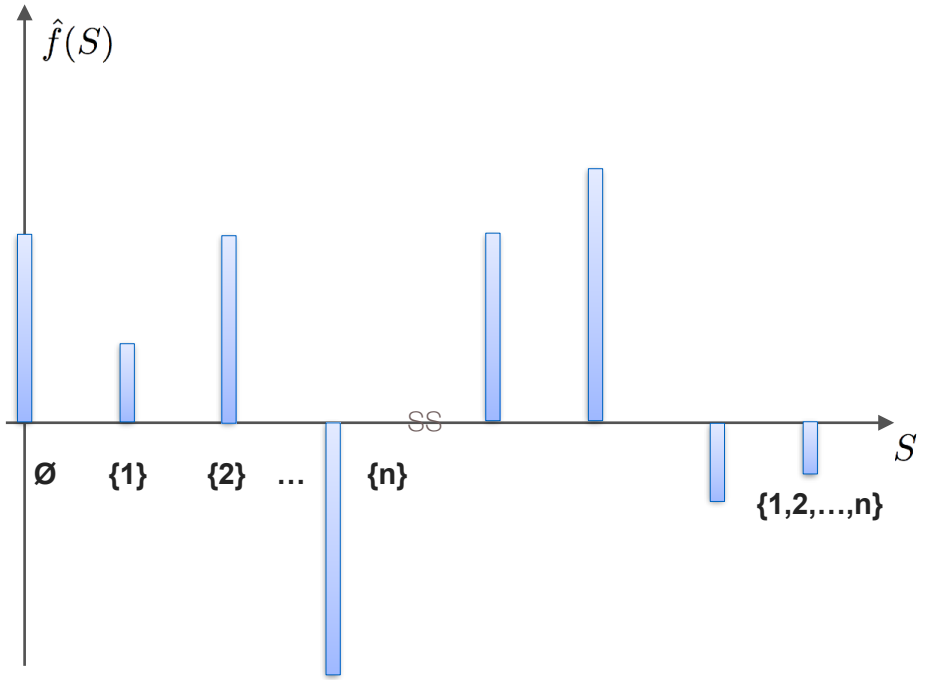


Approximation (or cutting the tail!)

$f: \{-1,1\}^n \rightarrow \mathbb{R}$

M

c_1	c_2	c_3	…	c_n	$f(c_1,c_2,\ldots,c_n)$
1	1	1	…	1	1
1	-1	1			
-1	1	1			
.					
.					
.					
-1	-1	-1			



- **Low-degree algorithm: for *some functions,* a polynomial number of examples required to approximate the "low" Fourier coefficients [1]**

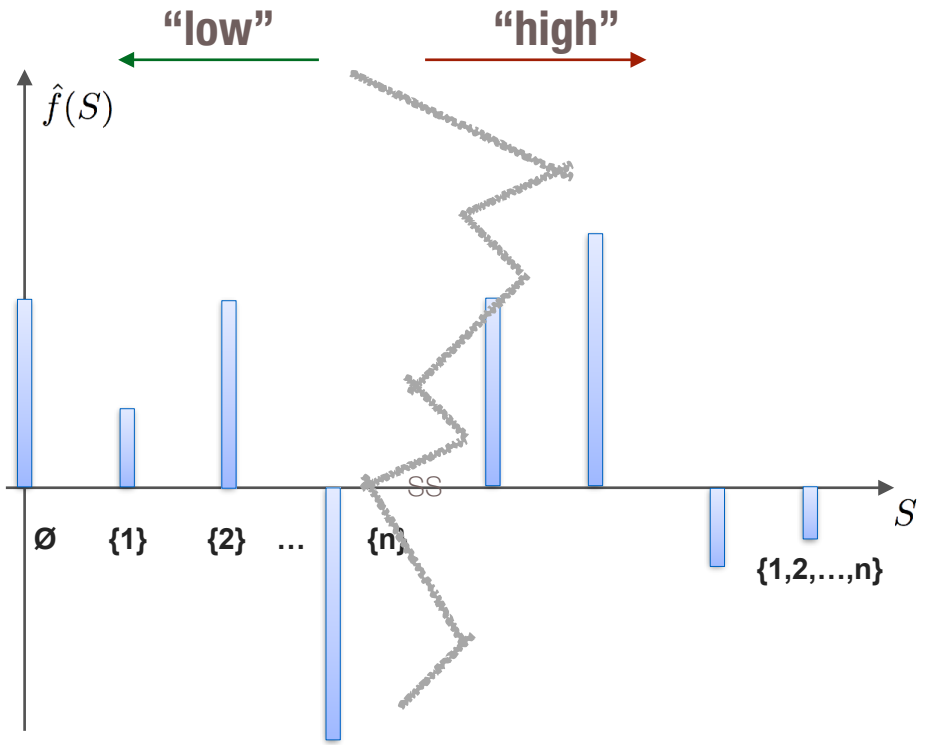
[1] Linial, N., Mansour, Y. and Nisan, N., 1993. Constant depth circuits, Fourier transform, and learnability. Journal of the ACM (JACM), 40(3), pp.607-620.

Approximation (or cutting the tail!)

$f: \{-1,1\}^n \rightarrow \mathbb{R}$

M

c_1	c_2	c_3	…	c_n	$f(c_1,c_2,\ldots,c_n)$
1	1	1	…	1	1
1	-1	1			
-1	1	1			
.					
.					
.					
-1	-1	-1			



- **Low-degree algorithm: for *some functions*, a polynomial number of examples required to approximate the "low" Fourier coefficients [1]**

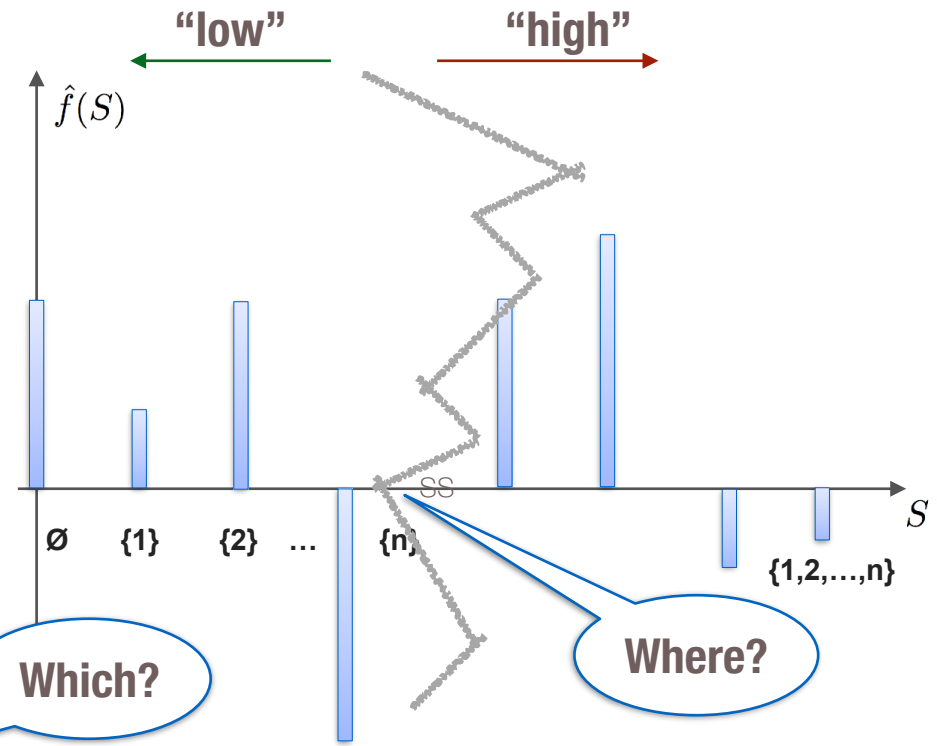
[1] Linial, N., Mansour, Y. and Nisan, N., 1993. Constant depth circuits, Fourier transform, and learnability. Journal of the ACM (JACM), 40(3), pp.607-620.

Approximation (or cutting the tail!)

$f: \{-1,1\}^n \rightarrow \mathbb{R}$

M

c_1	c_2	c_3	…	c_n	$f(c_1,c_2,\ldots,c_n)$
1	1	1	…	1	1
1	-1	1			
-1	1	1			
.					
.					
.					
-1	-1	-1			

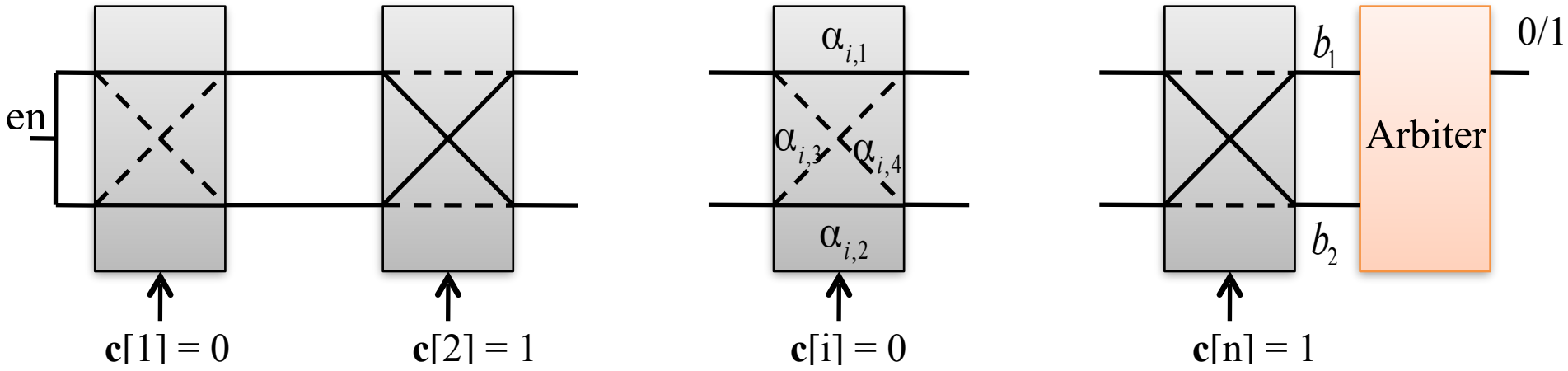


- **Low-degree algorithm: for *some functions*, a polynomial number of examples required to approximate the "low" Fourier coefficients [1]**

[1] Linial, N., Mansour, Y. and Nisan, N., 1993. Constant depth circuits, Fourier transform, and learnability. Journal of the ACM (JACM), 40(3), pp.607-620.

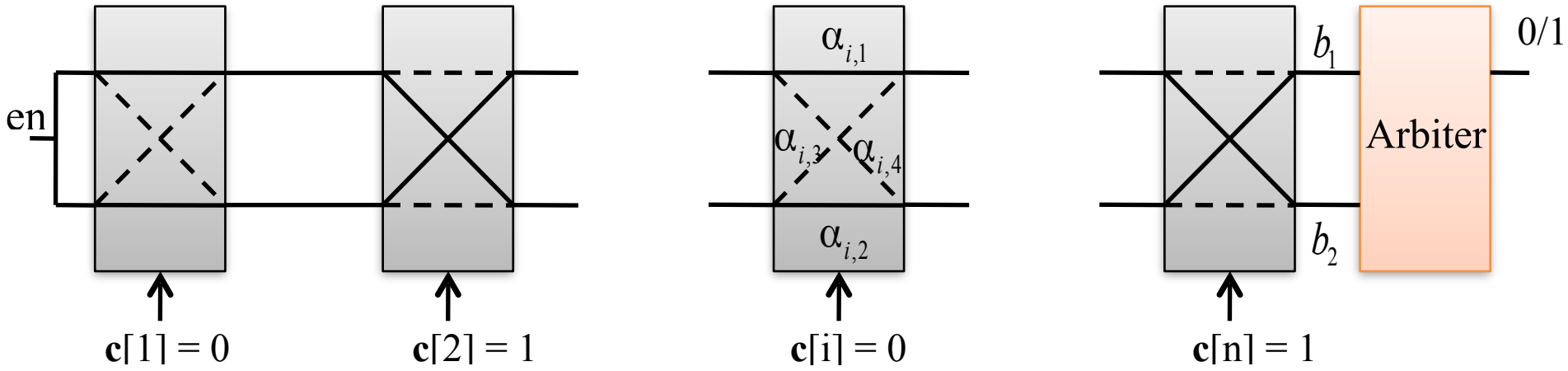
A PUF can be represented by a Boolean function.
But, do we reflect some important characteristics of that, e.g., being noisy, biased, etc., in this model?

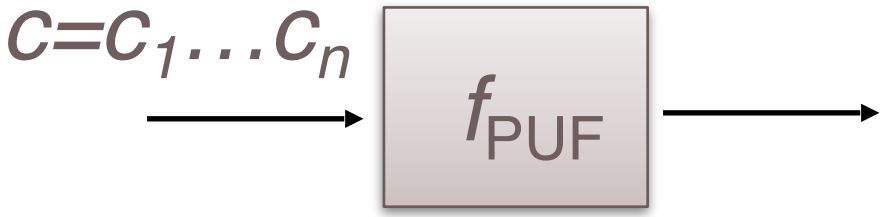
Classification Noise



[1] Ganji, F., Tajik, S. and Seifert, J.P., 2015, August. Why attackers win: on the learnability of XOR arbiter PUFs. In *International Conference on Trust and Trustworthy Computing* (pp. 22-39). Springer.

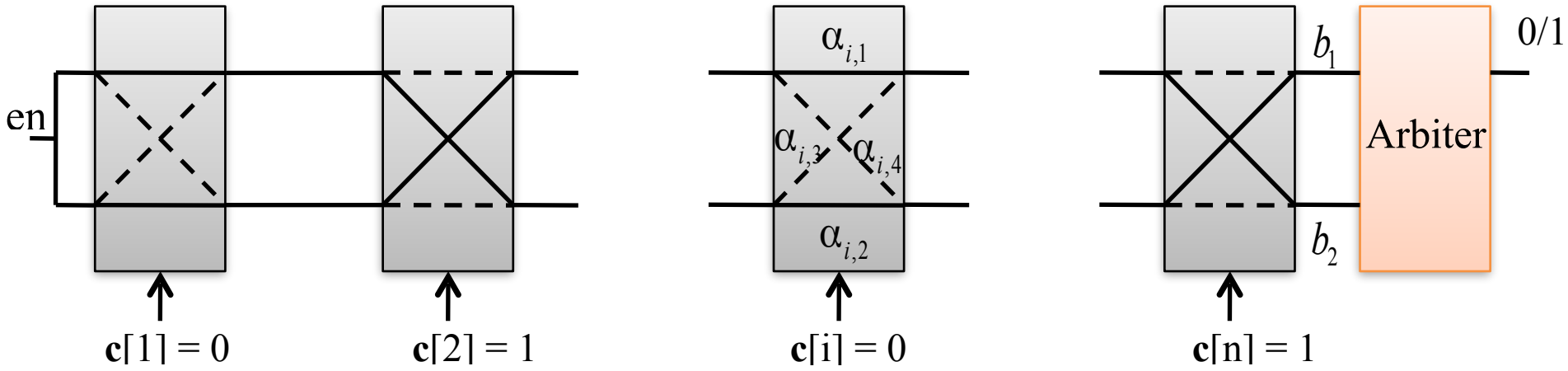
Classification Noise

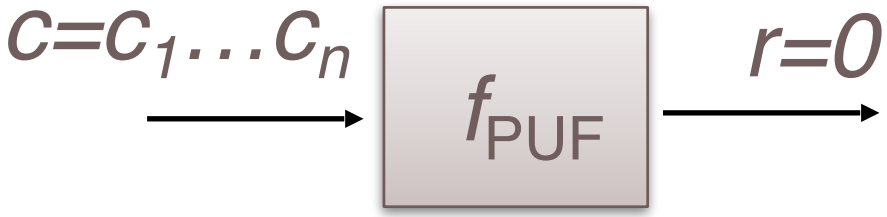




[1] Ganji, F., Tajik, S. and Seifert, J.P., 2015, August. Why attackers win: on the learnability of XOR arbiter PUFs. In *International Conference on Trust and Trustworthy Computing* (pp. 22-39). Springer.

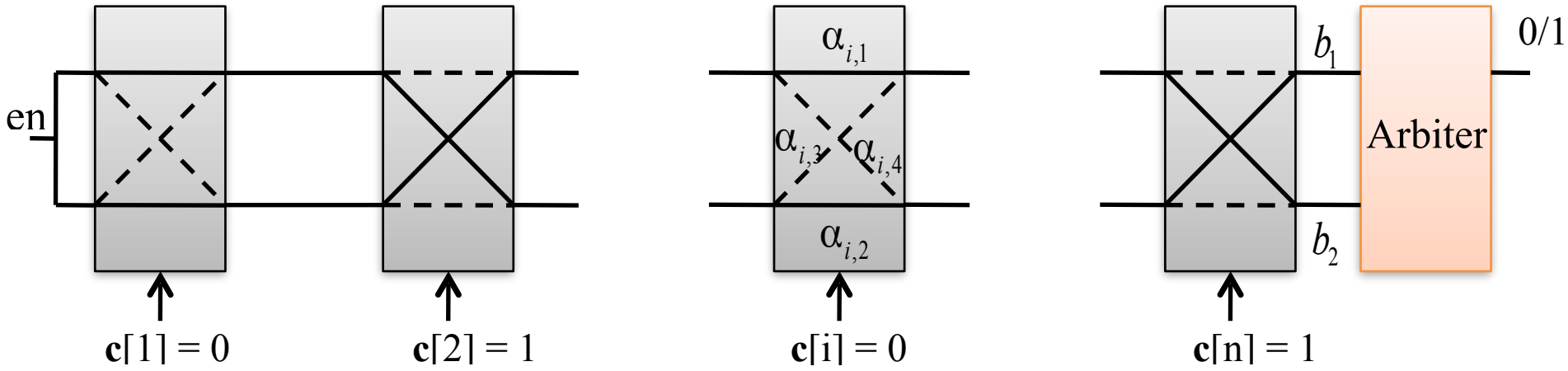
Classification Noise

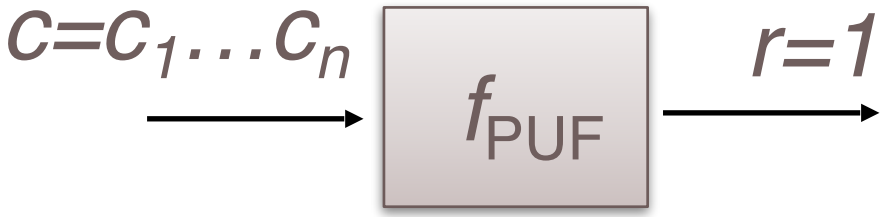




[1] Ganji, F., Tajik, S. and Seifert, J.P., 2015, August. Why attackers win: on the learnability of XOR arbiter PUFs. In *International Conference on Trust and Trustworthy Computing* (pp. 22-39). Springer.

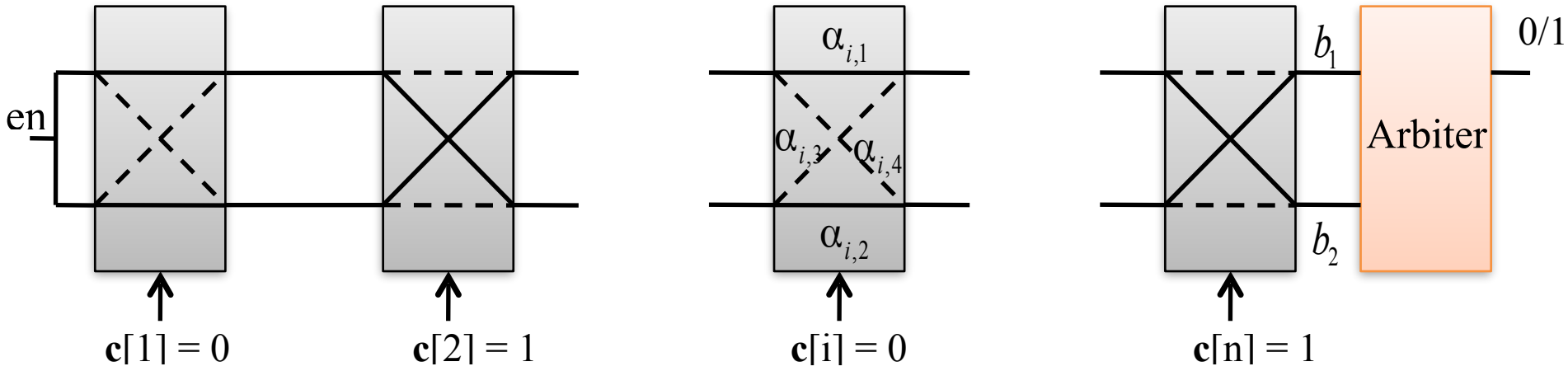
Classification Noise

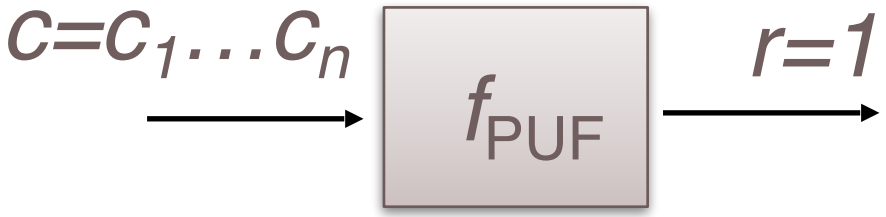




[1] Ganji, F., Tajik, S. and Seifert, J.P., 2015, August. Why attackers win: on the learnability of XOR arbiter PUFs. In *International Conference on Trust and Trustworthy Computing* (pp. 22-39). Springer.

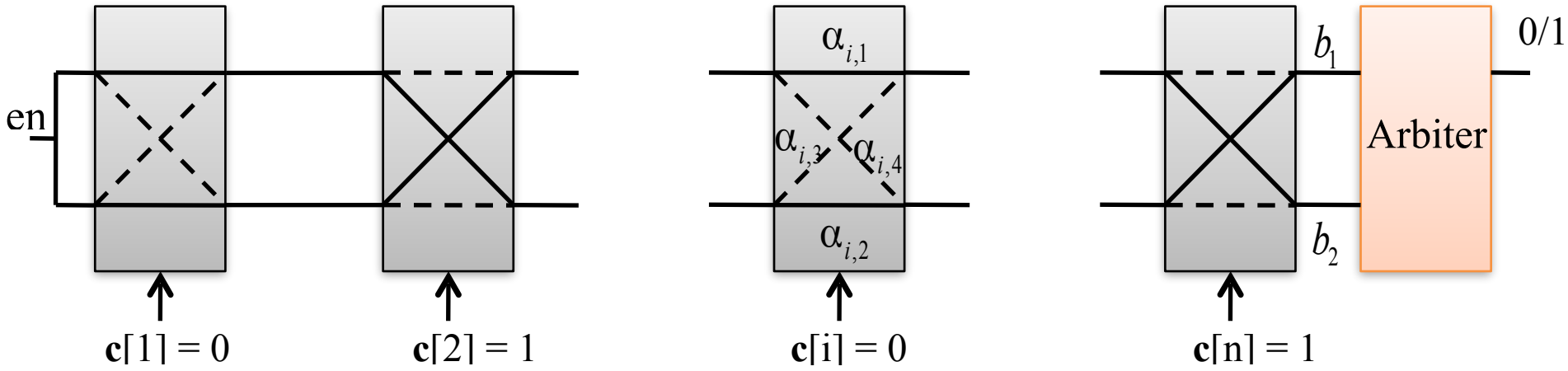
Classification Noise

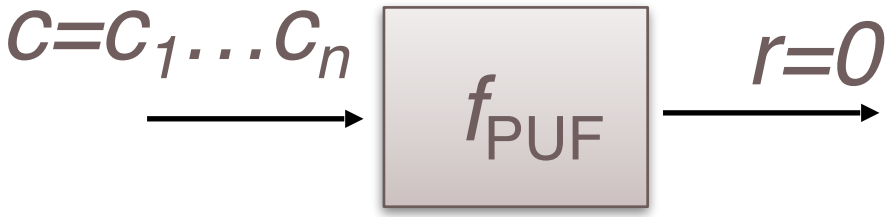




[1] Ganji, F., Tajik, S. and Seifert, J.P., 2015, August. Why attackers win: on the learnability of XOR arbiter PUFs. In *International Conference on Trust and Trustworthy Computing* (pp. 22-39). Springer.

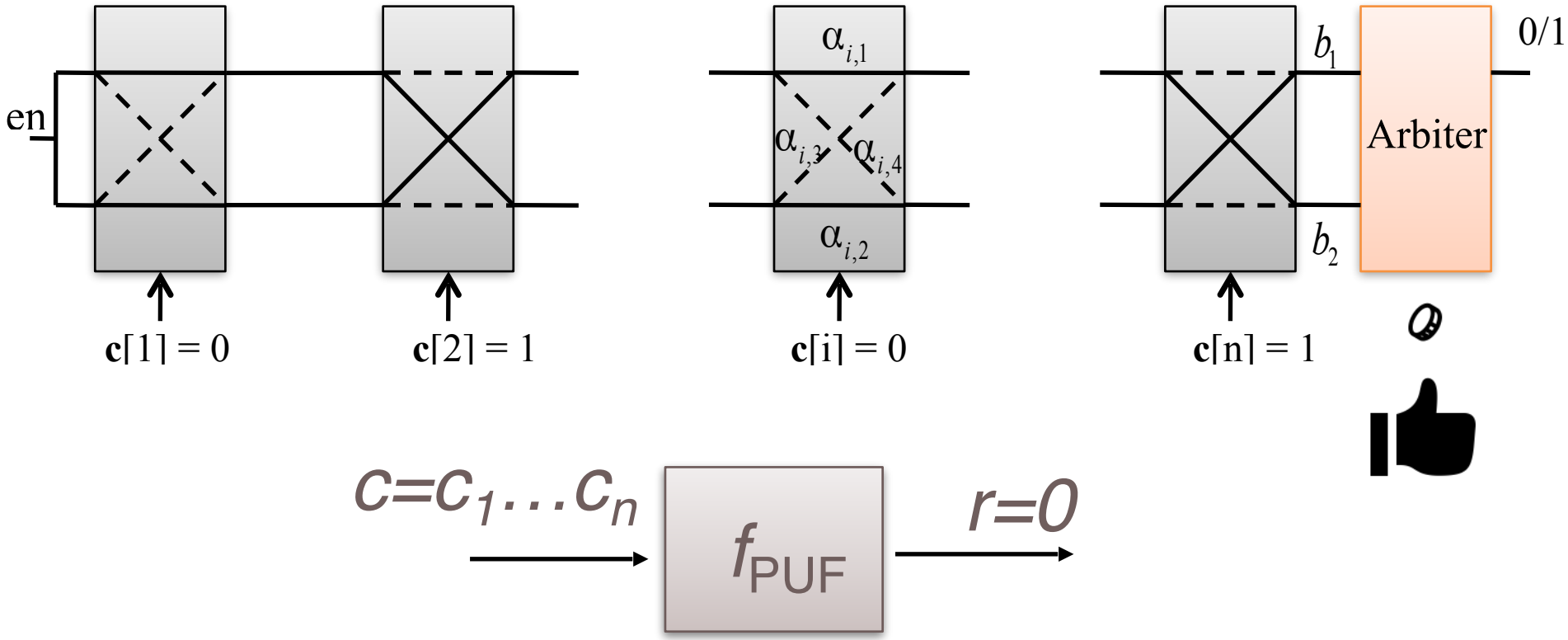
Classification Noise





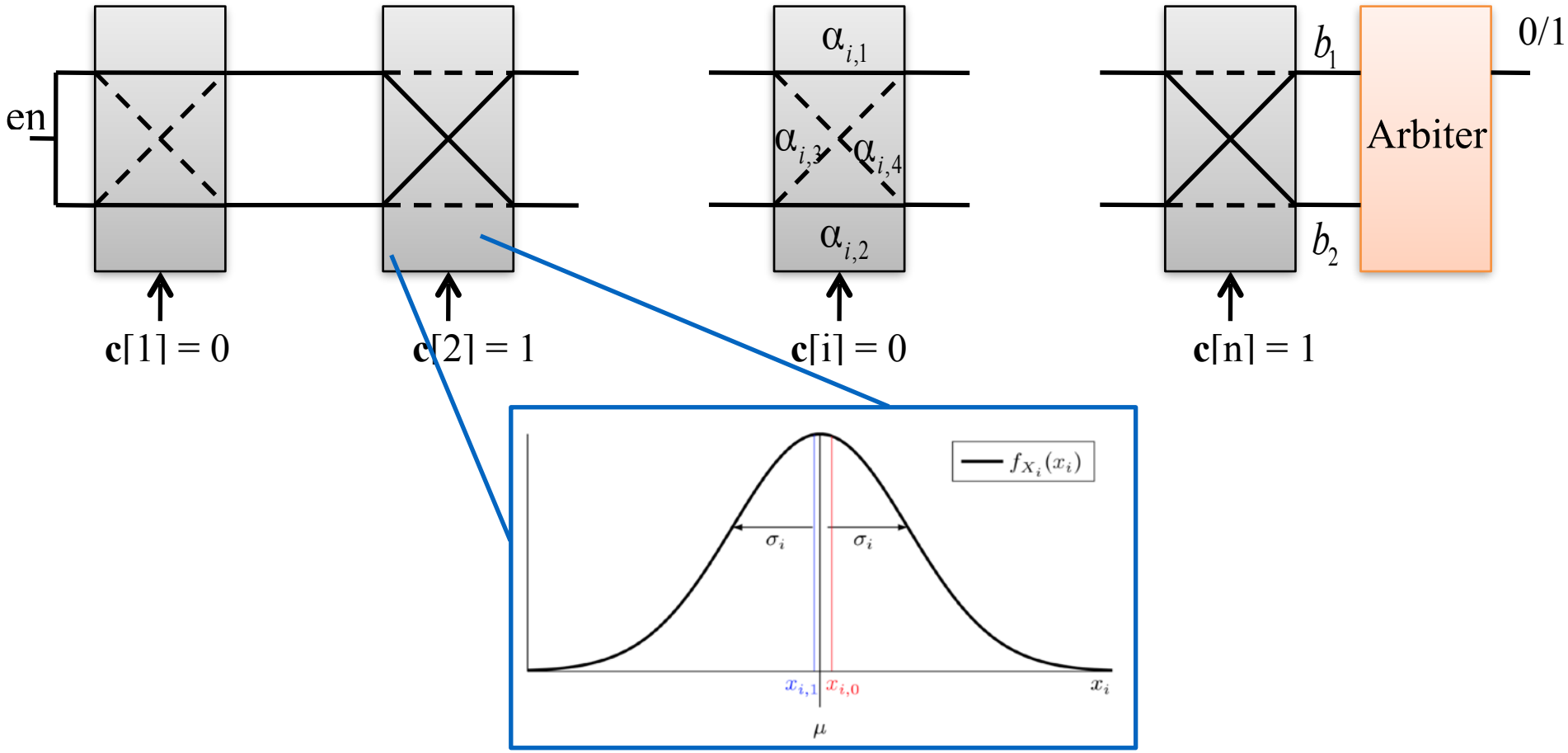
[1] Ganji, F., Tajik, S. and Seifert, J.P., 2015, August. Why attackers win: on the learnability of XOR arbiter PUFs. In *International Conference on Trust and Trustworthy Computing* (pp. 22-39). Springer.

Classification Noise



[1] Ganji, F., Tajik, S. and Seifert, J.P., 2015, August. Why attackers win: on the learnability of XOR arbiter PUFs. In *International Conference on Trust and Trustworthy Computing* (pp. 22-39). Springer.

Attribute Noise

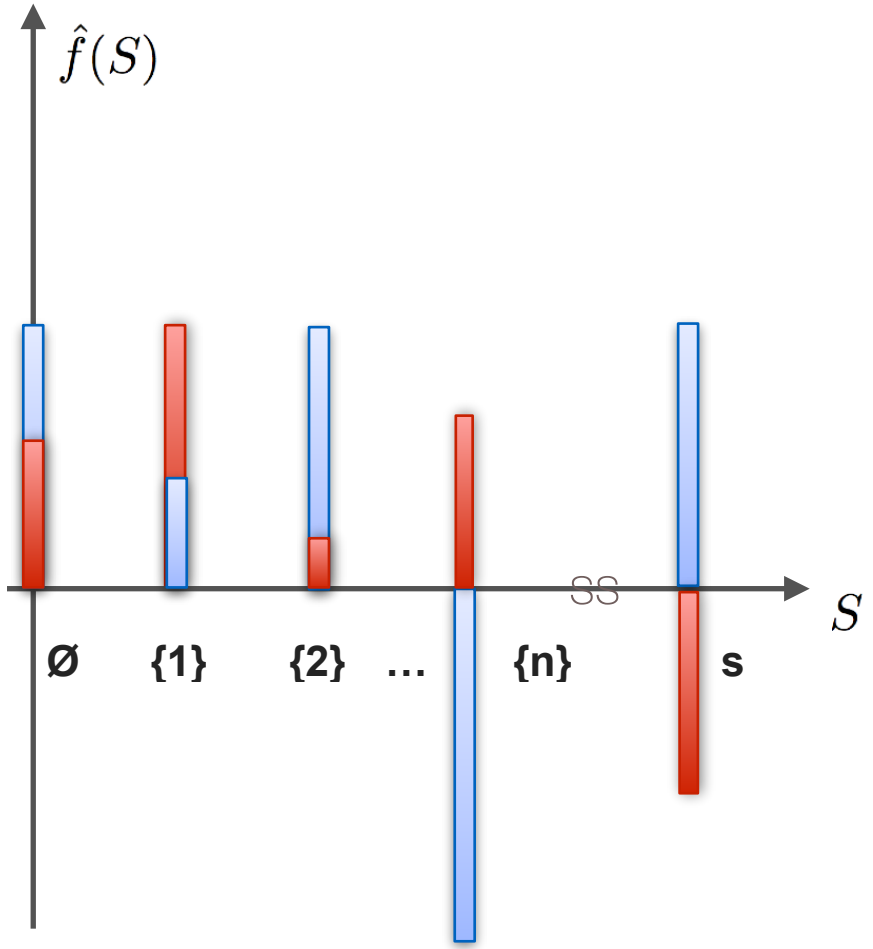


[1] Ganji, F., S.Tajik, and J.-P.Seifert. 2018, A Fourier analysis based attack against physically unclonable functions. In Intl. Conf. on Financial Crypto. and Data Security. Springer.

[2] Maes, R., 2013, August. An accurate probabilistic reliability model for silicon PUFs. In International Workshop on Cryptographic Hardware and Embedded Systems (pp. 73-89). Springer.

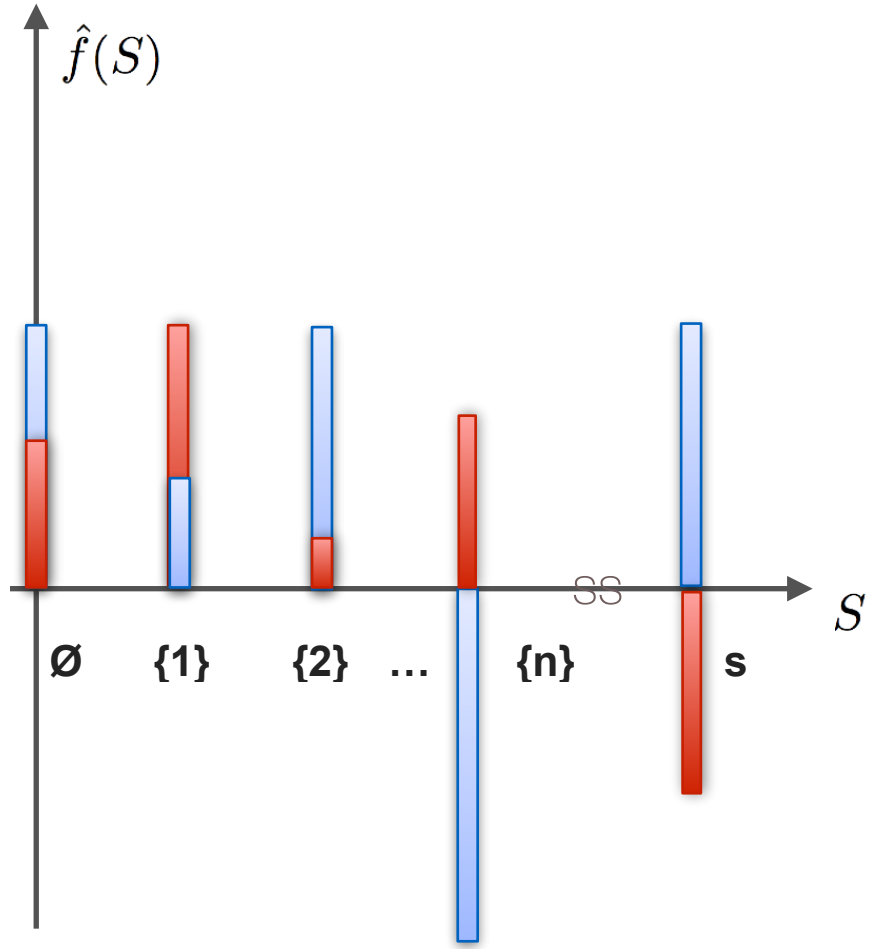
Attribute Noise in Real-world Implementations

- Some examples
  - The effect of the routing
  - Having not sufficient deviation in the manufacturing process variations from one instance to another
  - Aging
- What is the impact of these? More biased responses.
- Under the noisy conditions: the approximated Fourier coefficients are attenuated, and a polynomial increase in the number of example [1]



Attribute Noise in Real-world Implementations

- Some examples
  - The effect of the routing
  - Having not sufficient deviation in the manufacturing process variations from one instance to another
  - Aging
- What is the impact of these? More biased responses.
- Under the noisy conditions: the approximated Fourier coefficients are attenuated, and a polynomial increase in the number of example [1]



[1] Bshouty, N.H., Jackson, J.C. and Tamon, C., 2003. Uniform-distribution attribute noise learnability. Information and Computation, 187(2), pp.277-290.

Fourier-based attacks are still feasible!
BUT
To prove the security against ML attacks, should we rely on the infeasibility of some ML attacks?!

What to do to stop the attacker?

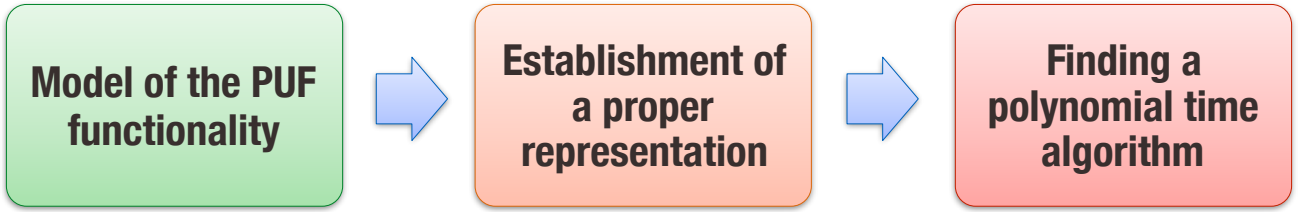
- **Ingredient:**
  - **Controlling mechanisms,**
  - **Non-linearity,**
  - **Adding the noise**

What to do to stop the attacker?

- **Ingredient:**
  - **Controlling mechanisms,**
  - **Non-linearity,**
  - **Adding the noise**

What to do to stop the attacker?

- **Ingredient:**
  - **Controlling mechanisms,**
  - **Non-linearity,**
  - **Adding the noise**



What to do to stop the attacker?

- **Ingredient:**
  - **Controlling mechanisms,**
  - **Non-linearity,**
  - **Adding the noise**

What to do to stop the attacker?

- **Ingredient:**
  - **Controlling mechanisms,**
  - **Non-linearity,**
  - **Adding the noise**

At least, known attacks are ineffective

What to do to stop the attacker?

- **Ingredient:**
  - **Controlling mechanisms,**
  - **Non-linearity,**
  - **Adding the noise**

Model of the PUF functionality → **Establishment of a proper representation** → **Finding a polynomial time algorithm**

Are you sure that it is secure now?

At least, known attacks are ineffective

What to do to stop the attacker?

- **Ingredient:**
  - **Controlling mechanisms,**
  - **Non-linearity,**
  - **Adding the noise**

Model of the PUF functionality → **Establishment of a proper representation** → **Finding a polynomial time algorithm**

Are you sure that it is secure now?

But, it can be attacked one way or another.

At least, known attacks are ineffective

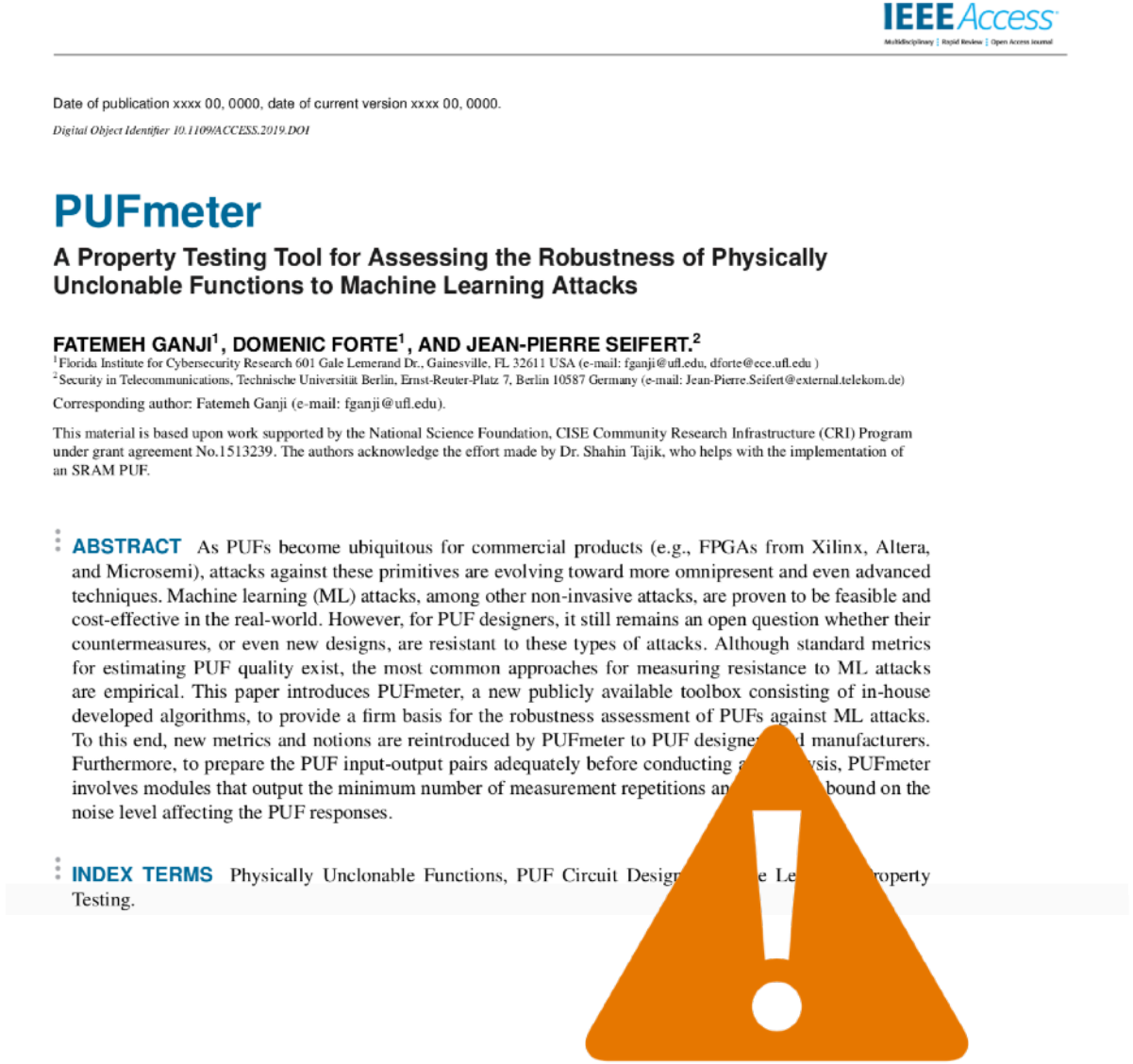
Take-home Message

- If we are given a set of CRPs, how to proceed?
  - Find the most appropriate representation
  - Find the most efficient algorithm to learn that representation
  - Be careful about what you conclude ;-)

Source: https://blogs.mathworks.com/loren/2016/08/08/text-mining-machine-learning-research-papers-with-matlab/

Future Directions

- Development of a new benchmarking system and Security assessment tools and metrics
  - Boolean and Fourier analyses
  - Property testing
  - Metrics, e.g., the average sensitivity
- Infeasibility of PAC learning and its consequences
  - No polynomial-sized representation of a function, and/or existence of no polynomial time algorithm to learn a target concept or its associated representation
  - Empirical algorithms may still be useful!
  - Generalization of the results of this study to other PUFs



IEEE Access

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2019.DOI

PUFmeter

A Property Testing Tool for Assessing the Robustness of Physically Unclonable Functions to Machine Learning Attacks

FATEMEH GANJI[1], DOMENIC FORTE[1], AND JEAN-PIERRE SEIFERT.[2]
[1]Florida Institute for Cybersecurity Research 601 Gale Lemerand Dr., Gainesville, FL 32611 USA (e-mail: fganji@ufl.edu, dforte@ece.ufl.edu)
[2]Security in Telecommunications, Technische Universität Berlin, Ernst-Reuter-Platz 7, Berlin 10587 Germany (e-mail: Jean-Pierre.Seifert@external.telekom.de)
Corresponding author: Fatemeh Ganji (e-mail: fganji@ufl.edu).

This material is based upon work supported by the National Science Foundation, CISE Community Research Infrastructure (CRI) Program under grant agreement No.1513239. The authors acknowledge the effort made by Dr. Shahin Tajik, who helps with the implementation of an SRAM PUF.

ABSTRACT As PUFs become ubiquitous for commercial products (e.g., FPGAs from Xilinx, Altera, and Microsemi), attacks against these primitives are evolving toward more omnipresent and even advanced techniques. Machine learning (ML) attacks, among other non-invasive attacks, are proven to be feasible and cost-effective in the real-world. However, for PUF designers, it still remains an open question whether their countermeasures, or even new designs, are resistant to these types of attacks. Although standard metrics for estimating PUF quality exist, the most common approaches for measuring resistance to ML attacks are empirical. This paper introduces PUFmeter, a new publicly available toolbox consisting of in-house developed algorithms, to provide a firm basis for the robustness assessment of PUFs against ML attacks. To this end, new metrics and notions are reintroduced by PUFmeter to PUF designe[illegible]d manufacturers. Furthermore, to prepare the PUF input-output pairs adequately before conducting a[illegible]ysis, PUFmeter involves modules that output the minimum number of measurement repetitions an[illegible]bound on the noise level affecting the PUF responses.

INDEX TERMS Physically Unclonable Functions, PUF Circuit Design[illegible]e Le[illegible]roperty Testing.

A *short* list of related work

- **Pure, empirical machine learning attacks**

1. Rührmair, U., Sehnke, F., Sölter, J., Dror, G., Devadas, S. and Schmidhuber, J., 2010, October. Modeling attacks on physical unclonable functions. In Proceedings of the 17th ACM conference on Computer and communications security (pp. 237-249). ACM.
2. Rührmair, U., Sölter, J., Sehnke, F., Xu, X., Mahmoud, A., Stoyanova, V., Dror, G., Schmidhuber, J., Burleson, W. and Devadas, S., 2013. PUF modeling attacks on simulated and silicon data. IEEE Transactions on Information Forensics and Security, 8(11), pp.1876-1891.
3. Vijayakumar, A., Patil, V.C., Prado, C.B. and Kundu, S., 2016, May. Machine learning resistant strong PUF: Possible or a pipe dream?. In 2016 IEEE international symposium on hardware oriented security and trust (HOST) (pp. 19-24). IEEE.

- **Hybrid attacks**

4. Becker, G.T., 2015, September. The gap between promise and reality: On the insecurity of XOR arbiter PUFs. In International Workshop on Cryptographic Hardware and Embedded Systems (pp. 535-555). Springer, Berlin, Heidelberg.
5. Rührmair, U., Xu, X., Sölter, J., Mahmoud, A., Majzoobi, M., Koushanfar, F. and Burleson, W., 2014, September. Efficient power and timing side channels for physical unclonable functions. In International Workshop on Cryptographic Hardware and Embedded Systems (pp. 476-492). Springer, Berlin, Heidelberg.

- **Mathematical approaches and cryptanalysis**

6. Delvaux, J. and Verbauwhede, I., 2013, June. Side channel modeling attacks on 65nm arbiter PUFs exploiting CMOS device noise. In 2013 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST) (pp. 137-142). IEEE.
7. Sahoo, D.P., Nguyen, P.H., Mukhopadhyay, D. and Chakraborty, R.S., 2015. A case of lightweight PUF constructions: Cryptanalysis and machine learning attacks. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 34(8), pp.1334-1343.
8. Yamamoto, D., Takenaka, M., Sakiyama, K. and Torii, N., 2014, September. Security evaluation of bistable ring PUFs on FPGAs using differential and linear analysis. In 2014 Federated Conference on Computer Science and Information Systems (pp. 911-918). IEEE.

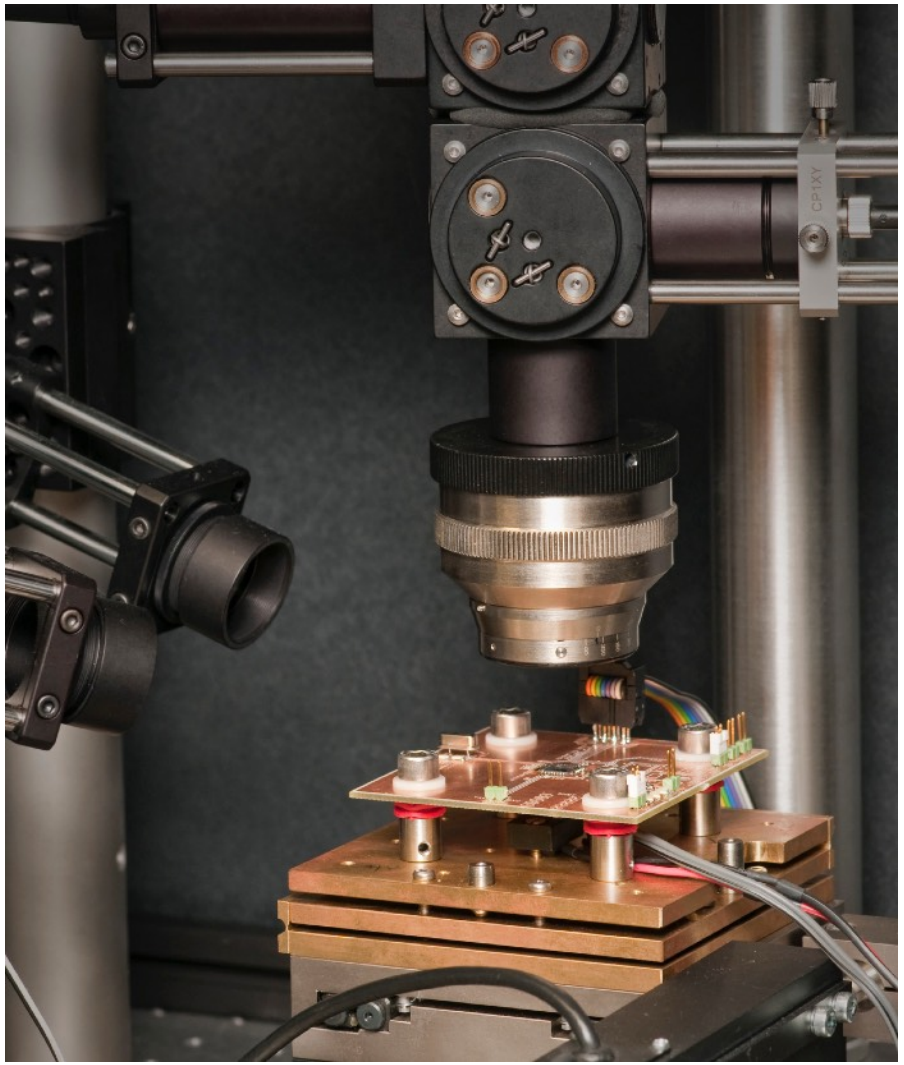
Physical Attacks

Are PUFs Tamper-evident?

- PUFs are believed to be tamper-evident against invasive attacks!
- Being tamper-evident against fully-invasive attacks have been experimentally verified for optical and coating PUFs.
- Unfortunately, for Intrinsic PUFs, limited information on tamper-evidence is available in the literature.
- mechanical stress from depackaging and substrate thinning have negligible effects on the absolute and relative frequencies of ring-oscillators [14]
- PUF developers do their best to mitigate the noisy responses of the PUF by different error correction techniques.
-

Optical Attacks

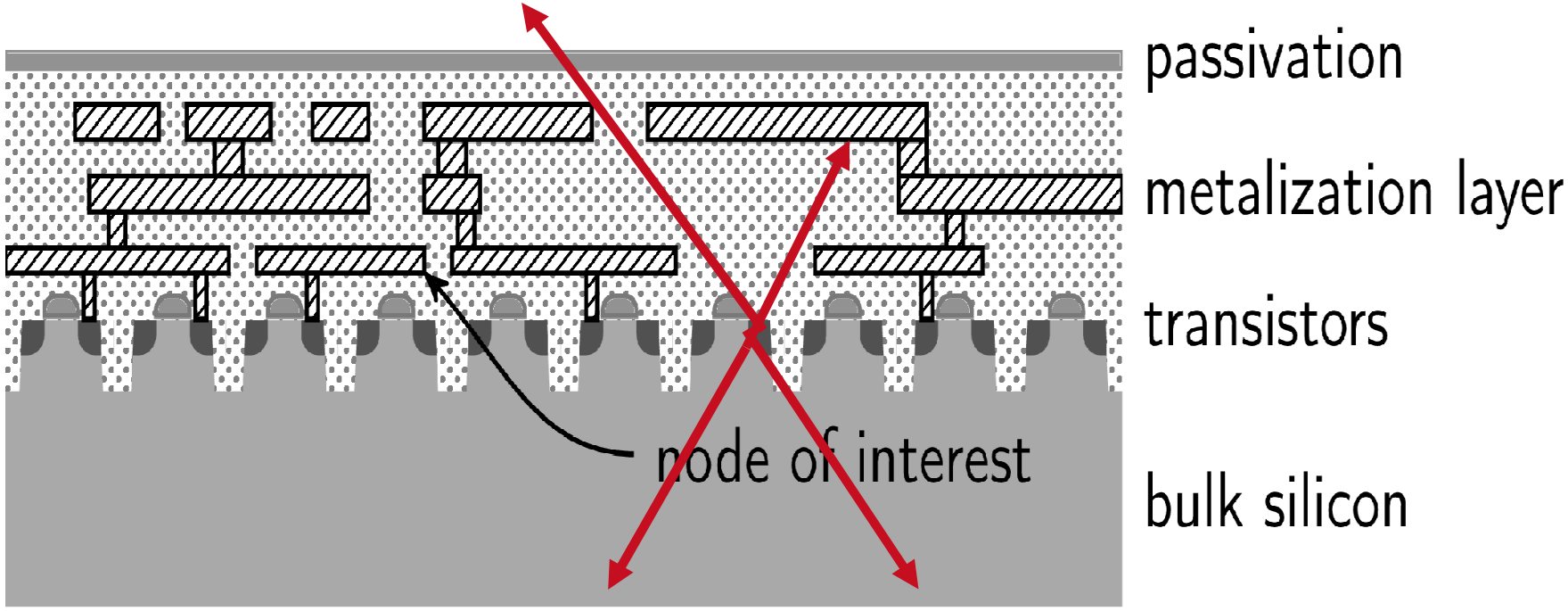
- Access to the surface of the chip without creating contacts with internal wires
- Optical interactions with transistors using known Failure Analysis (FA) tools
- Normally does not damage the system
- May or may not leave tamper evidence



[1] Schlösser, A., Nedospasov, D., Krämer, J., Orlic, S., & Seifert, J. P. (2012, September). Simple photonic emission analysis of AES. In International Workshop on Cryptographic Hardware and Embedded Systems (pp. 41-57). Springer, Berlin, Heidelberg.

IC Backside vs. IC Frontside

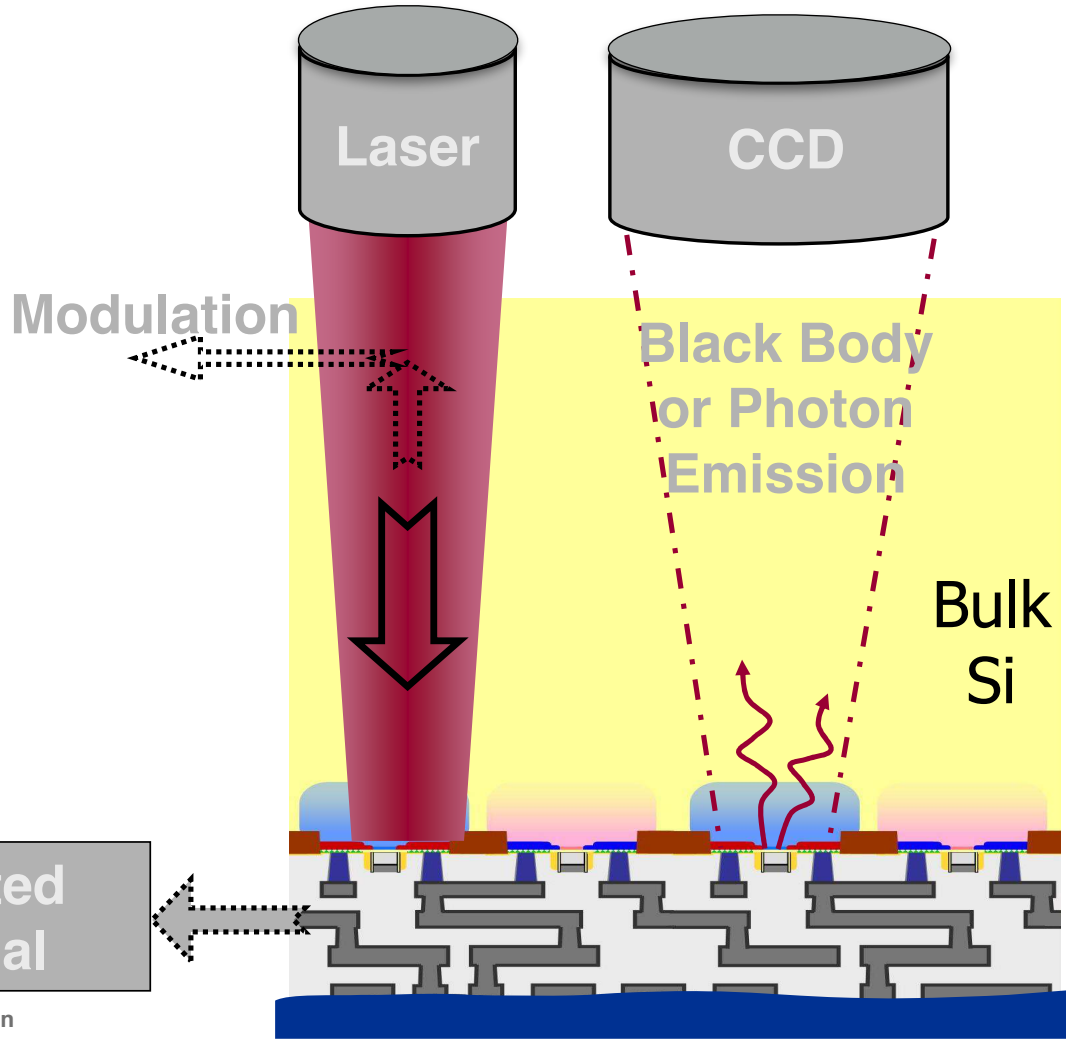
- **Frontside:** Multiple interconnect layers obstruct the optical path to transistor devices
- **Backside:** Active devices are directlv accessible



[1] Boit, Christian, and Philipp Scholz. "IC Debug and Fault Isolation for an Age of IoE and High Data Rates—A Vision." *IEEE Transactions on Components, Packaging and Manufacturing Technology* 8.5 (2018): 719-724.

Optical Techniques

- **Photon Emission**
- **Laser Stimulation**
- **Laser Fault Injection**
- **Optical Probing**



[1] Boit, Christian, and Philipp Scholz. "IC Debug and Fault Isolation for an Age of IoE and High Data Rates—A Vision." *IEEE Transactions on Components, Packaging and Manufacturing Technology* 8.5 (2018): 719-724.

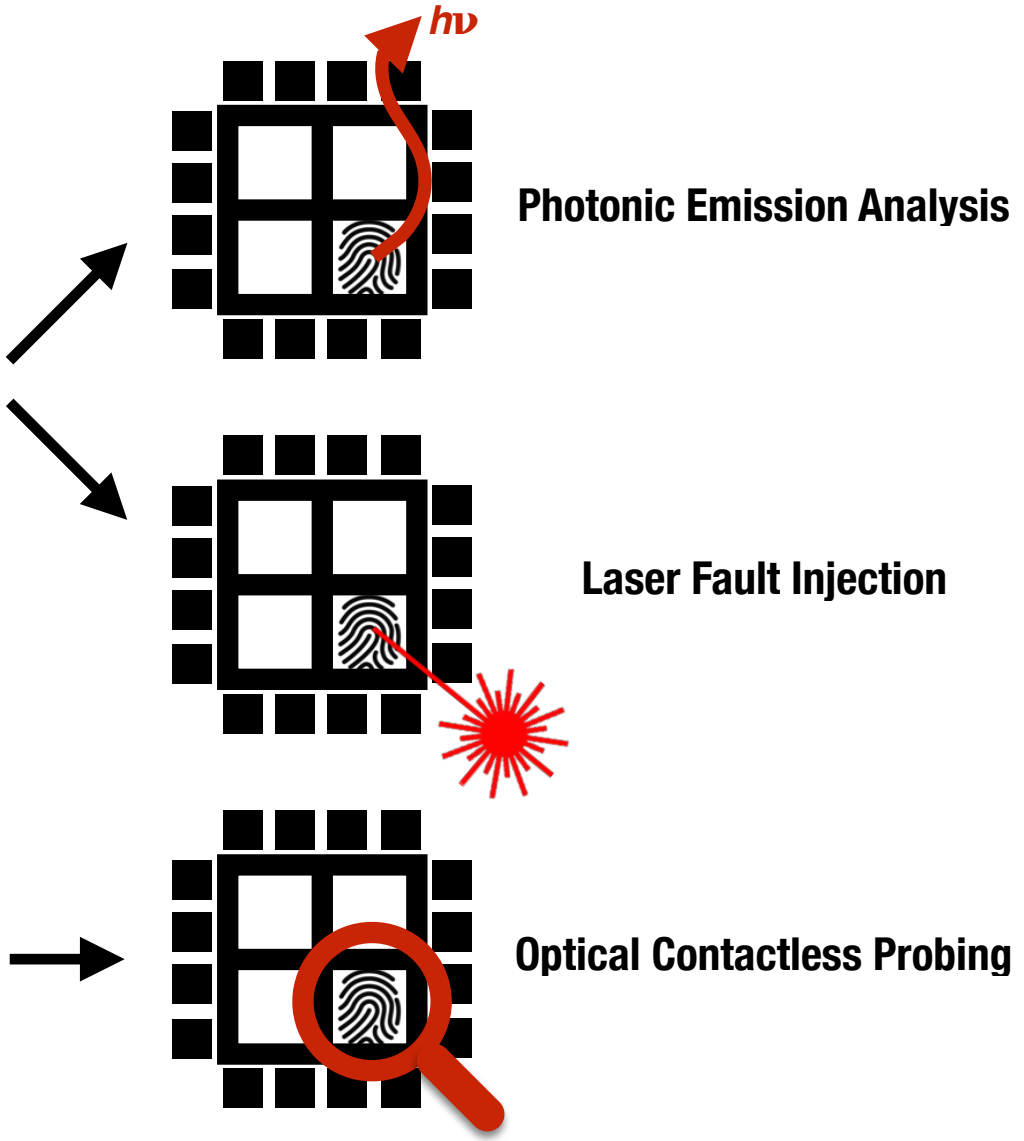
PUF Targets and Attack Techniques

1st scenario:

- PUF implementations are part of the user design and used during **runtime**
- Assumption: Access to the challenges and the responses are available

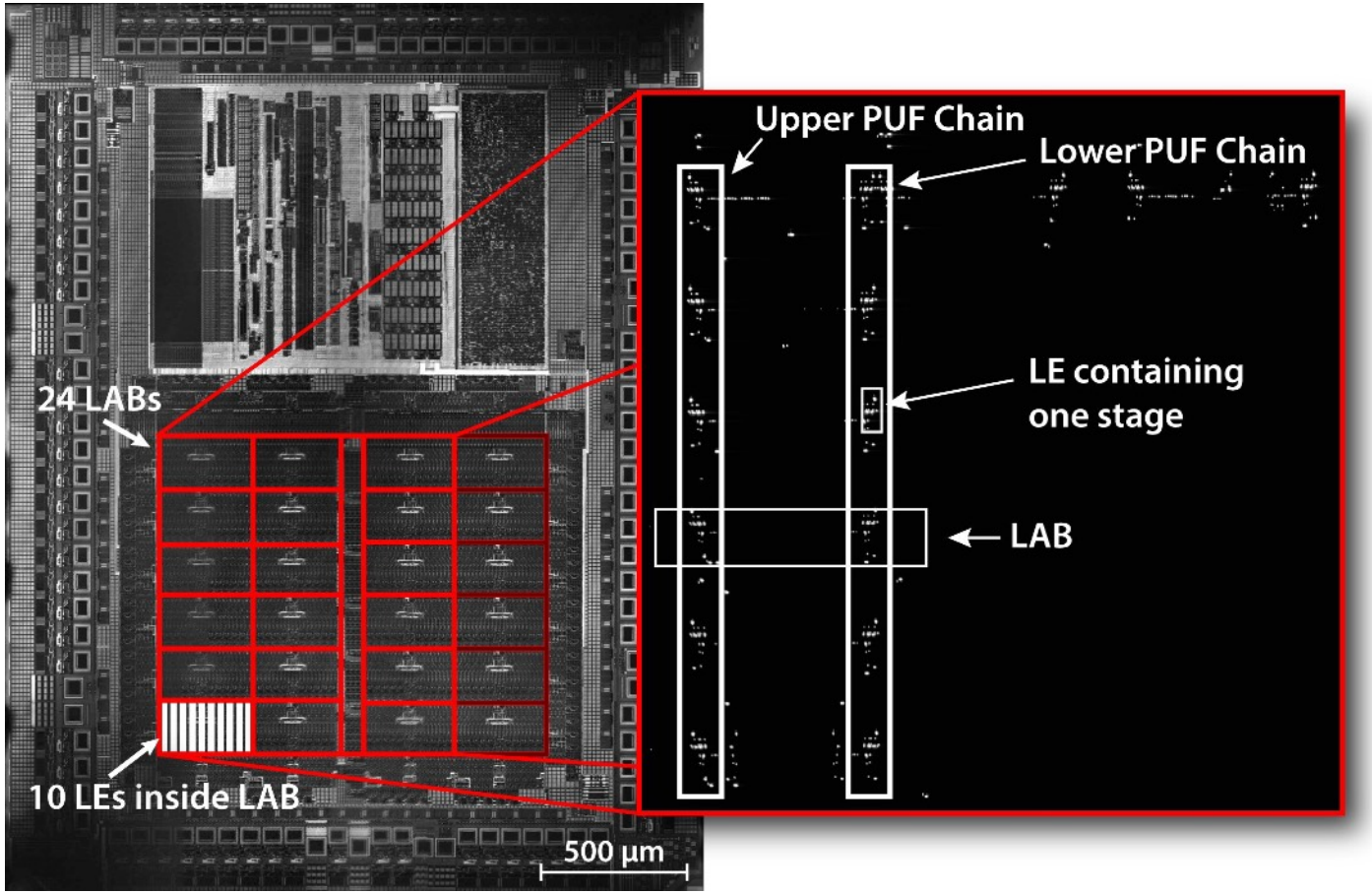
2nd scenario:

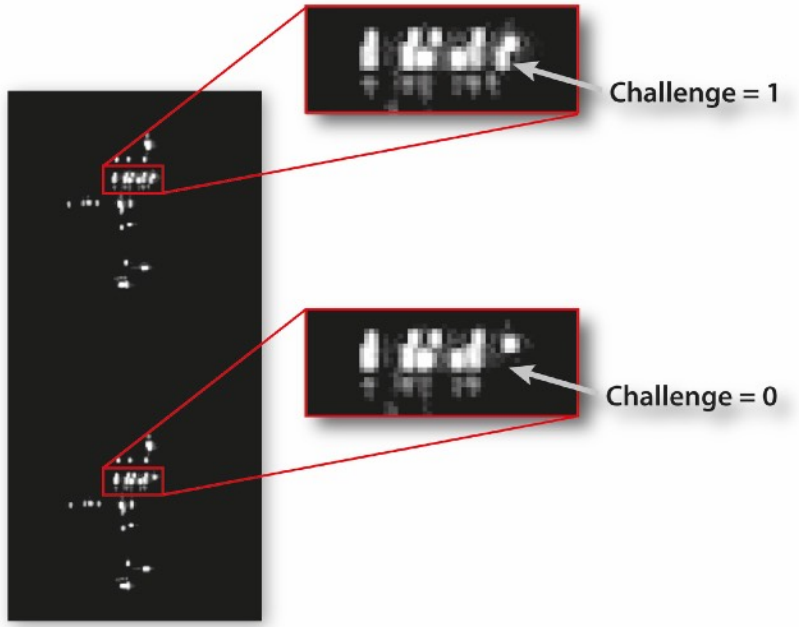
- PUF implementations are used during **configuration** to either decrypt the bitstream or authenticate the device to a RoT
- Assumption: No access to the challenges and the responses



[1] Tajik, Shahin. On the physical security of physically unclonable functions. Springer, 2018.

Optical Emission of Arbiter PUF





[1] Tajik et al. Emission Analysis of Hardware Implementations. DSD 2014: pp. 528-534
[2] Tajik et al. Physical Characterization of Arbiter PUFs. CHES 2014: pp. 493-509
[3] Tajik et al. Photonic Side Channel Analysis of Arbiter PUFs. Journal of Cryptology, 2017

- Measuring timing differences at the end of the chain on both paths
- **No response is needed!**
- Characterization of the PUF with **n+1** challenge: 1 reference challenge and n challenge with hamming distance one
- E.g.

C(0) = 0000…00 >> **Reference**

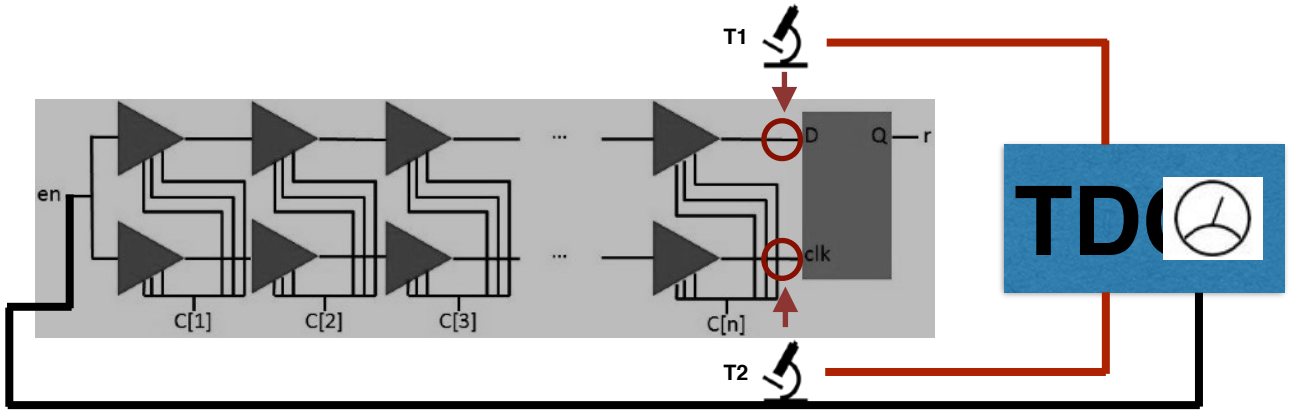
C(1) = 1000…00 >> **δ1 = +10 ps**

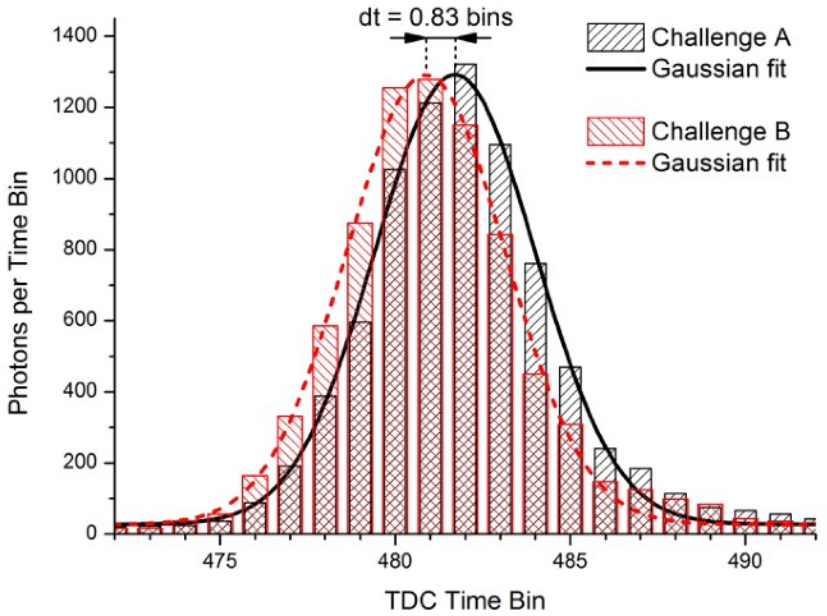
C(2) = 0100…00 >> **δ2 = +185 ps**

C(3) = 0010…00 >> **δ3 = -16 ps**

C(x) = 1110…00 >> **+179 ps**

measured value = +175 ps





- Measuring timing differences at the end of the chain on both paths
- **No response is needed!**
- Characterization of the PUF with **n+1** challenge: 1 reference challenge and n challenge with hamming distance one
- E.g.

C(0) = 0000…00 >> **Reference**

C(1) = 1000…00 >> **δ1 = +10 ps**

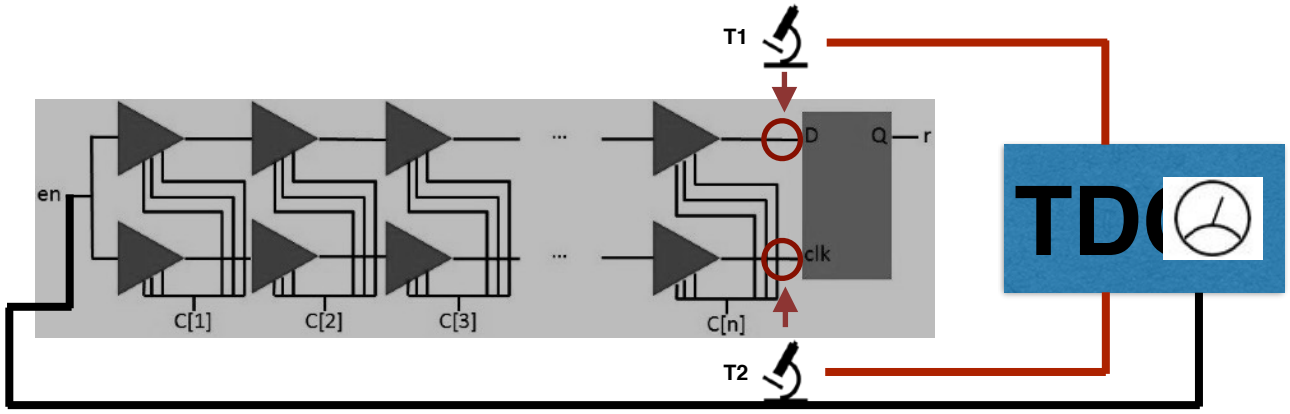
C(2) = 0100…00 >> **δ2 = +185 ps**

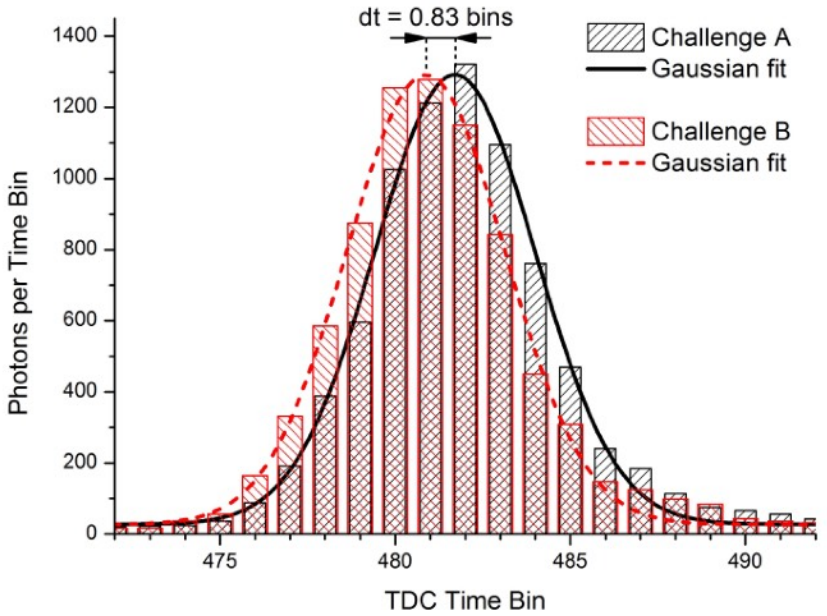
C(3) = 0010…00 >> **δ3 = -16 ps**

C(x) = 1110…00 >> +179 ps

measured value = +175 ps

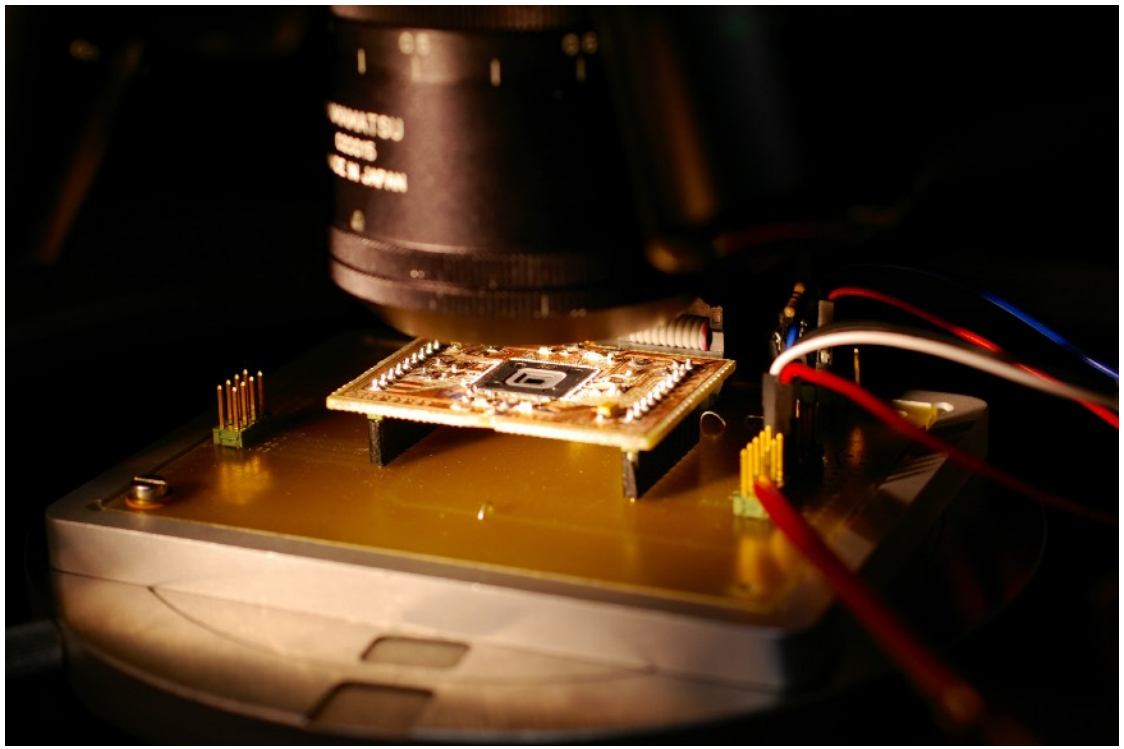
Linear increase in the required challenges with the number of PUFs in an XOR arbiter PUF!





Laser Fault Attack on XOR and RO PUF [1]

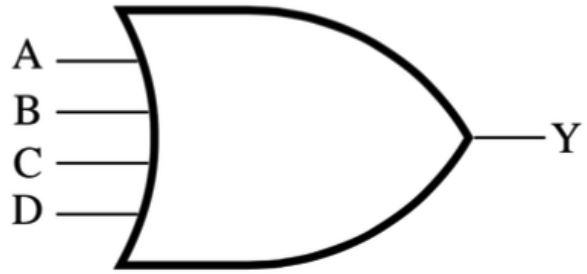
- **XOR arbiter PUF:**
  - **Simplifying ML attacks by deactivating all Arbiter chains except one!**
  - **Iterating the same approach until all Arbiter PUFs are learnt!**
- **RO PUF:**
  - **Reducing the Entropy of PUF by deactivating several ROs!**



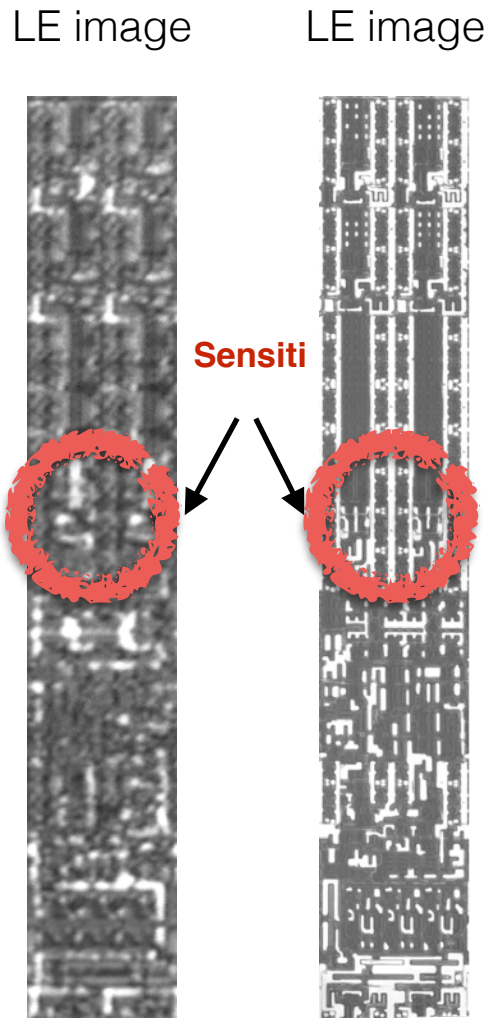
[1] Shahin Tajik, Heiko Lohrke, Fatemeh Ganji, Jean-Pierre Seifert, Christian Boit: **Laser Fault Attack on Physically Unclonable Functions.** FDTC 2015: pp. 85-96

Fault Injection into the Lookup Tables

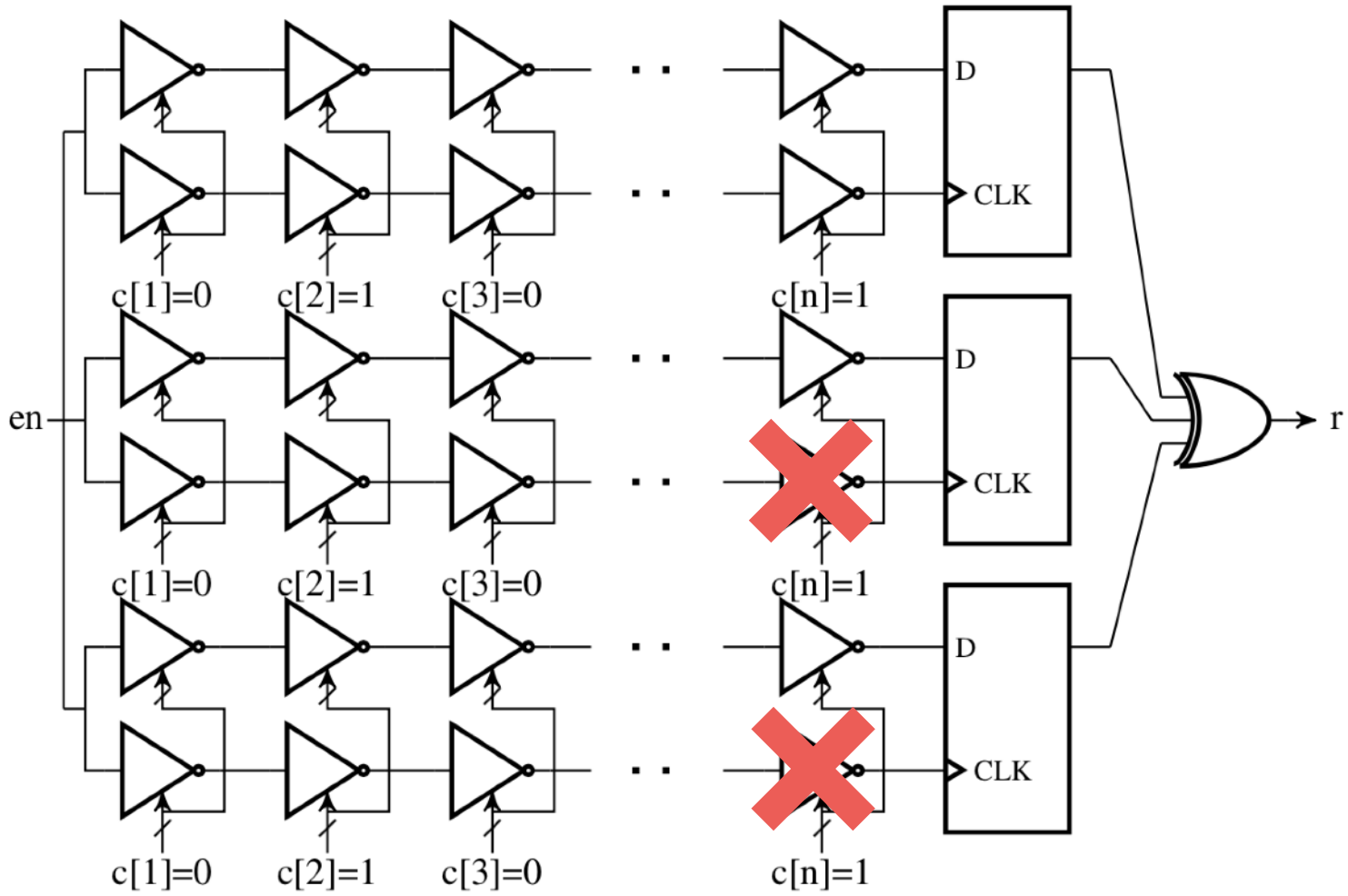
CD \ AB	00	01	10	11
00	0	1	1	1
01	1	1	1	1
10	1	1	1	1
11	1	1	1	1



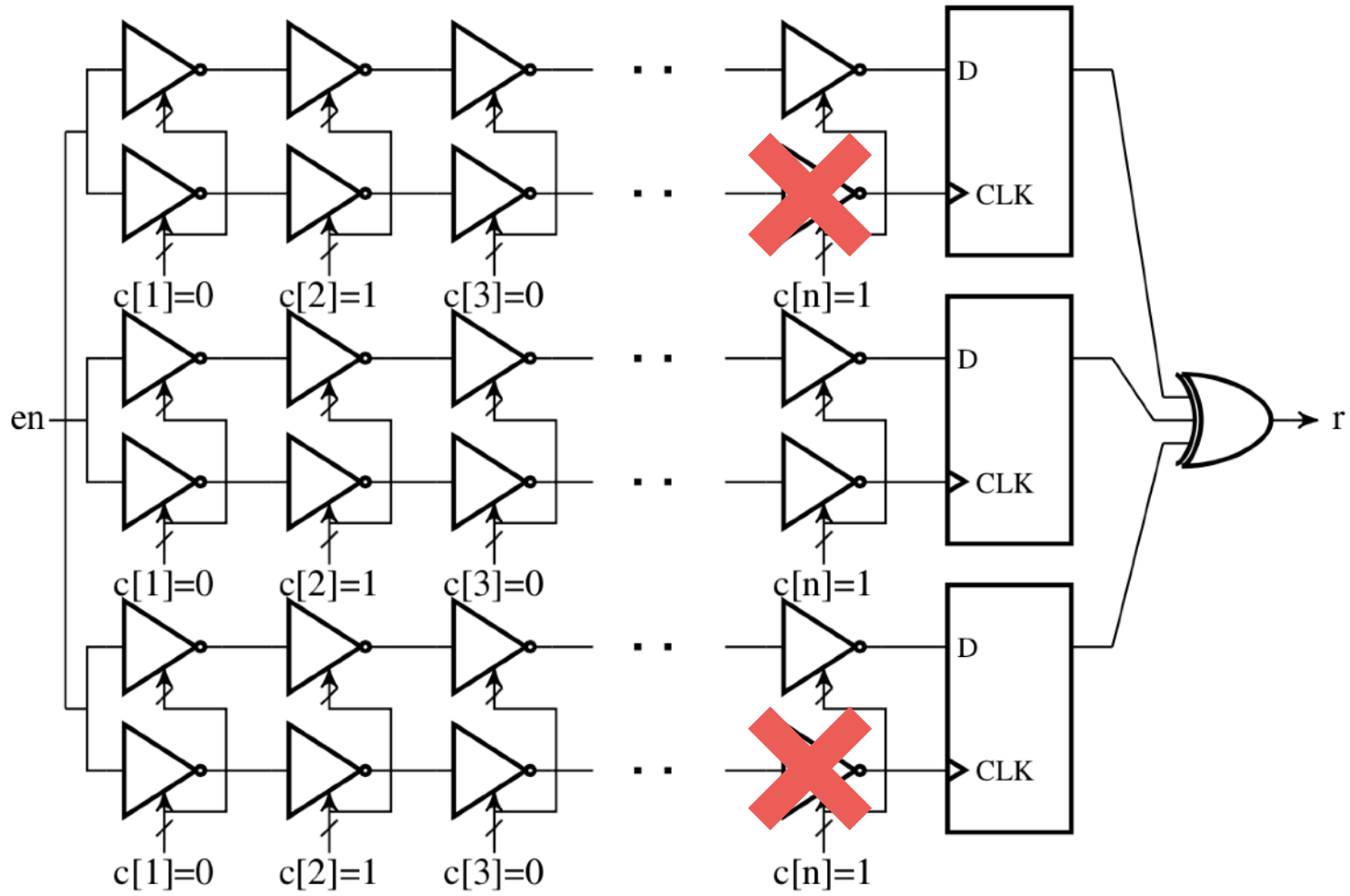
CD \ AB	00	01	10	11
00	1	1	1	1
01	1	1	1	1
10	1	1	1	1
11	1	1	1	0



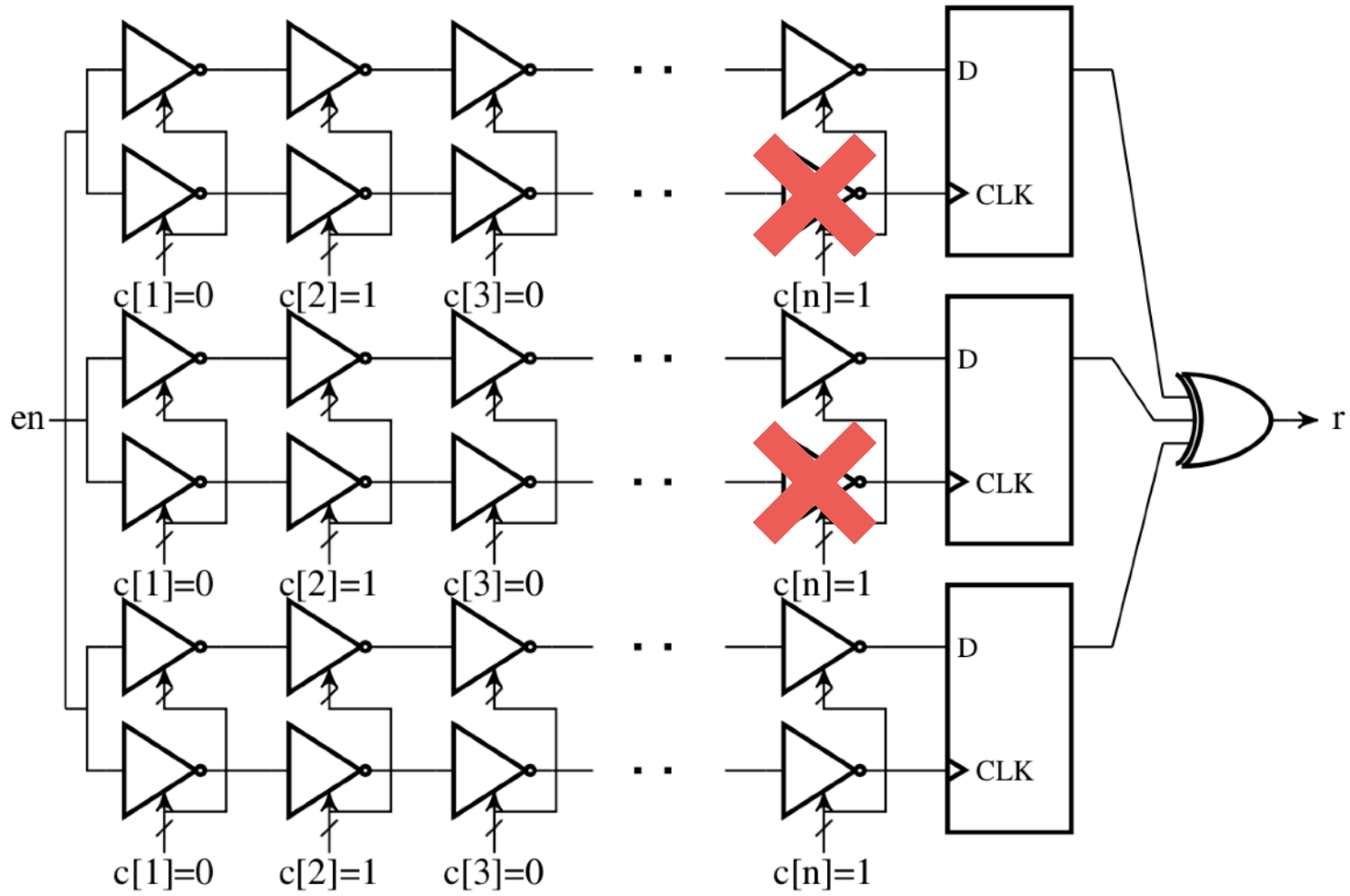
Launching ML Attack



Launching ML Attack



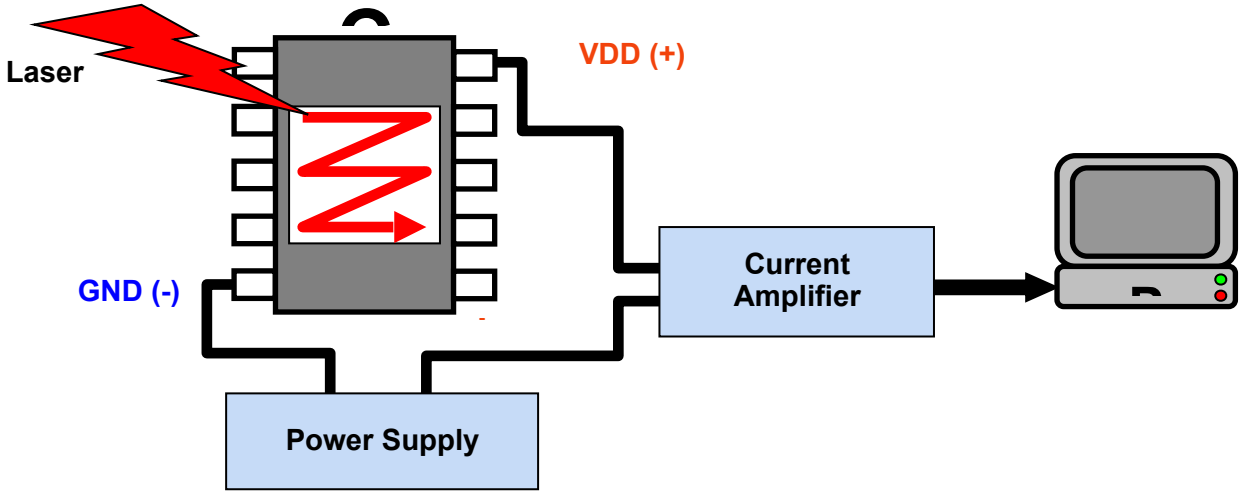
Launching ML Attack

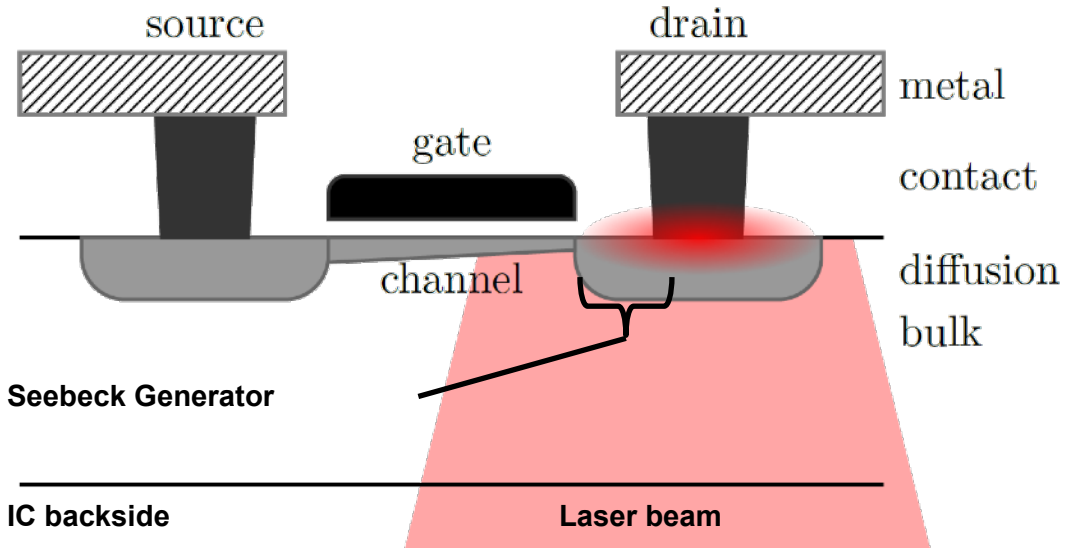


Attack against SRAM PUF

- Two obstacles for the attacker
  - How can the SRAM be read when no electrical access to the SRAM is available from the outside?
  - How can the CRP behavior of one PUF be digitally and physically cloned on another device?

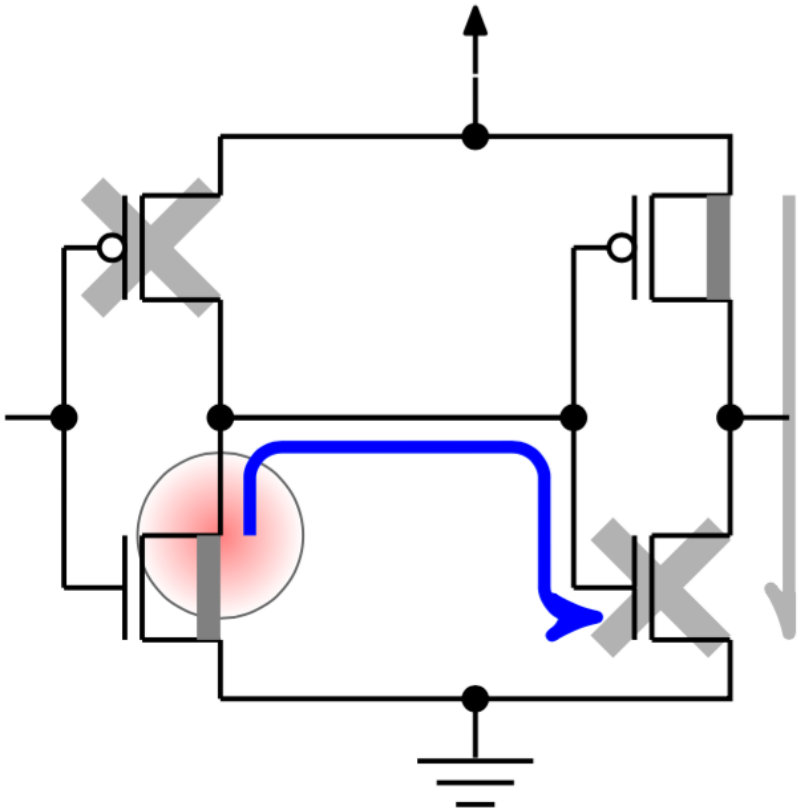
Laser Stimulation



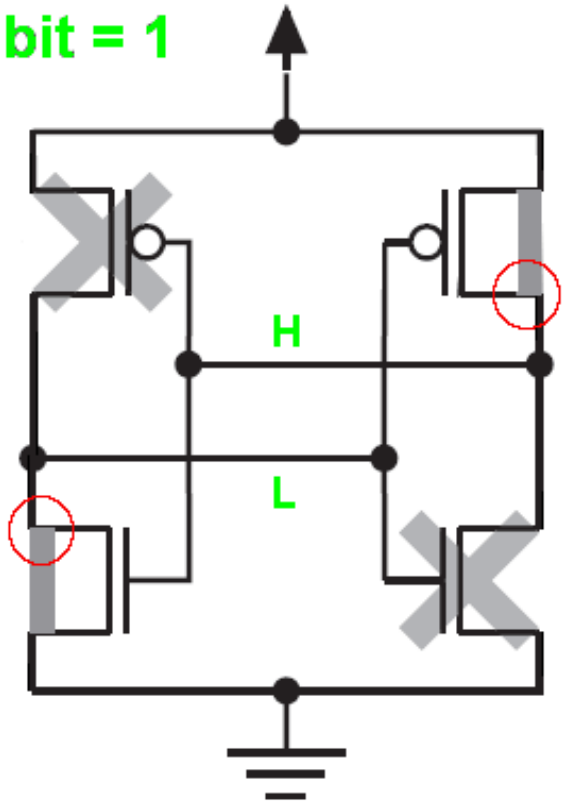


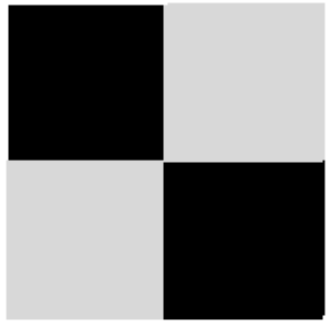
- The chip is scanned with a laser beam with either thermal or photoeletric interaction (TLS/PLS)
- The laser beam generates a Seebeck voltage (TLS) by creating a temperature gradient across dissimilar materials

Thermal Laser Stimulation (TLS) of SRAM Cells



- The Seebeck voltage changes current flow through the “off” transistors >> leakage current increases

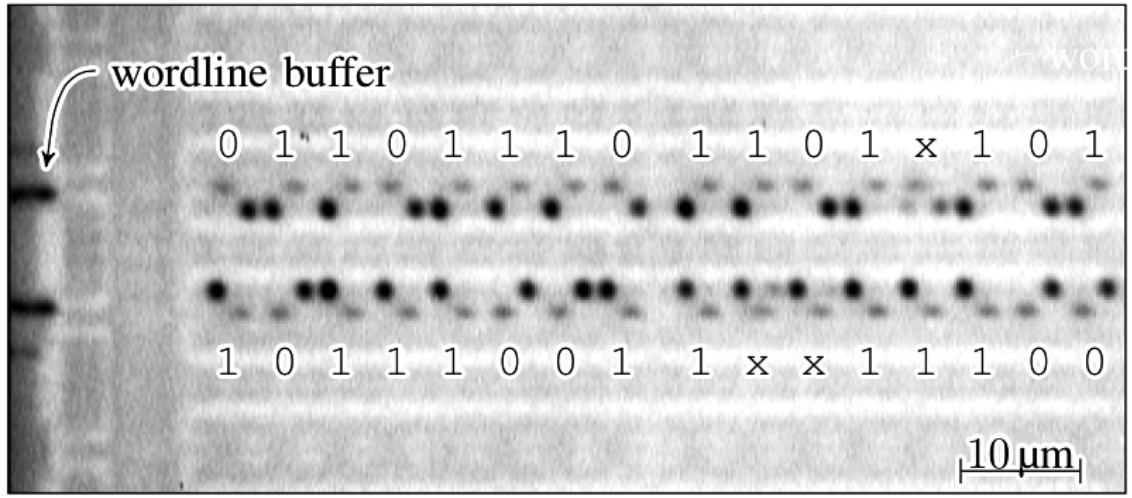




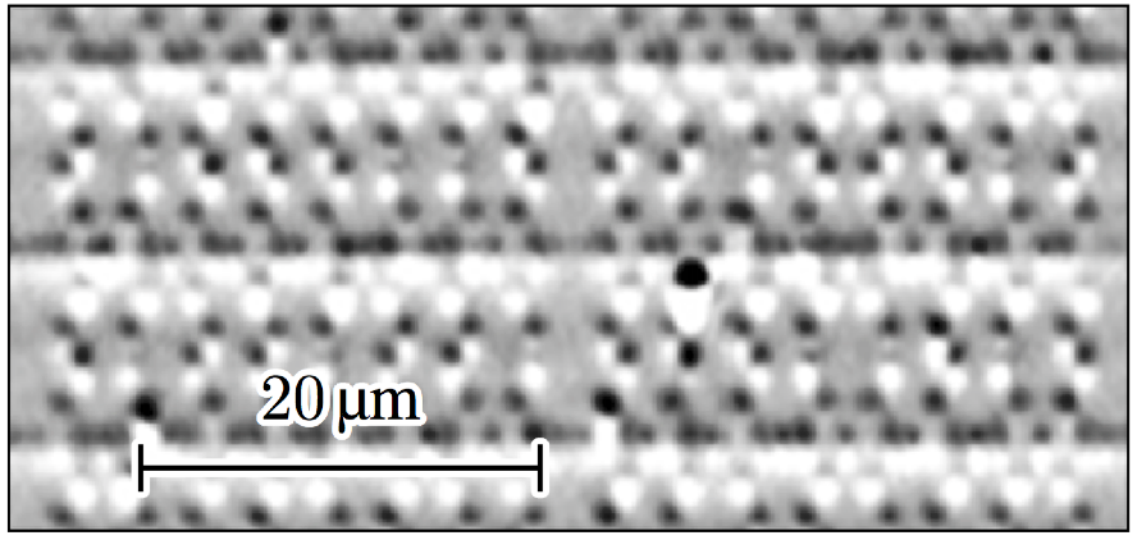
- Reaction of different areas of SRAM cells to TLS, depending on the stored value

Reading SRAM of ATMega328P

Reading SRAM using Photon Emission

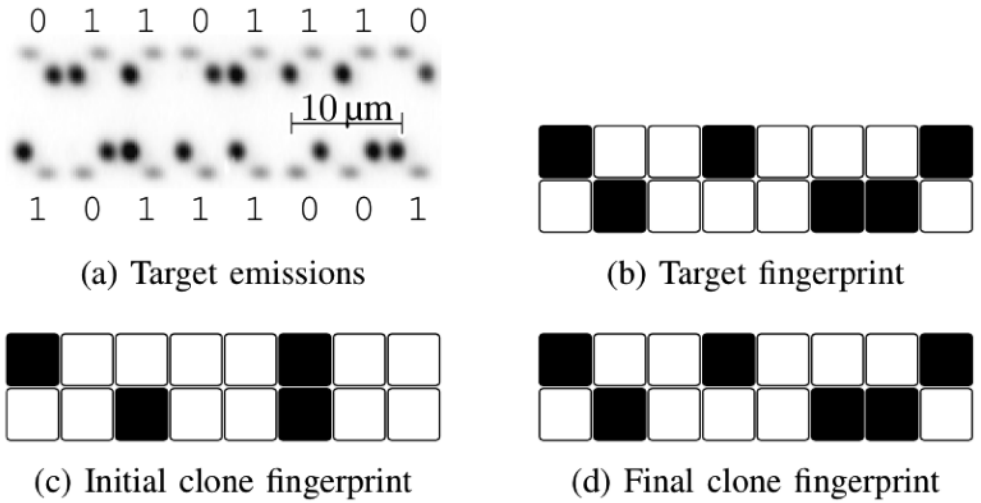


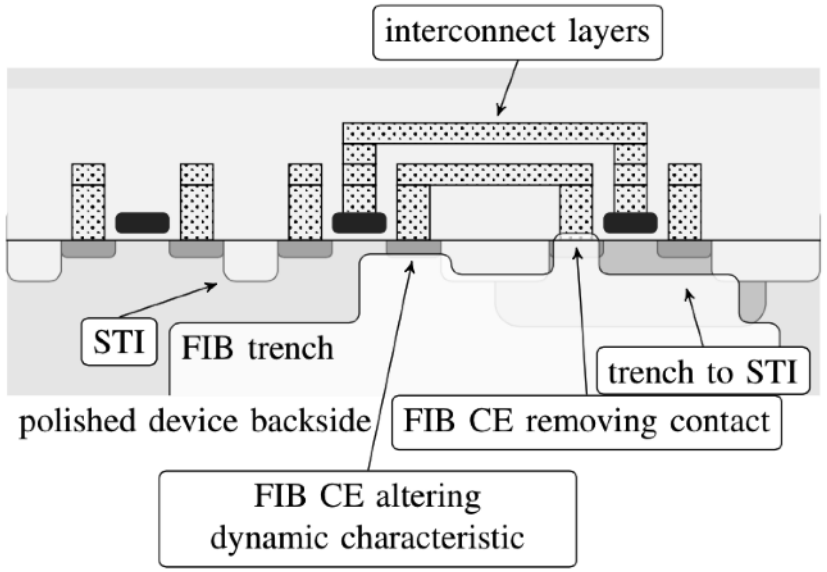
Reading SRAM using laser stimulation



Cloning SRAM PUF

- **Trenching silicon with FIB**
- **Removing contacts or trimming transistors by FIB to change their power-up states**





[1] Helfmeier, C., Boit, C., Nedospasov, D., & Seifert, J. P. (2013, June). Cloning physically unclonable functions. In *2013 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)* (pp. 1-6). IEEE.

Hard PUFs in Commercial FPGAs

Hard PUF Examples

SRAM PUF for
Microsemi SmartFusion2 and IGLOO2 Models

INTRINSIC ID

SRAM PUF for Intel/Altera Stratix 10

INTRINSIC ID

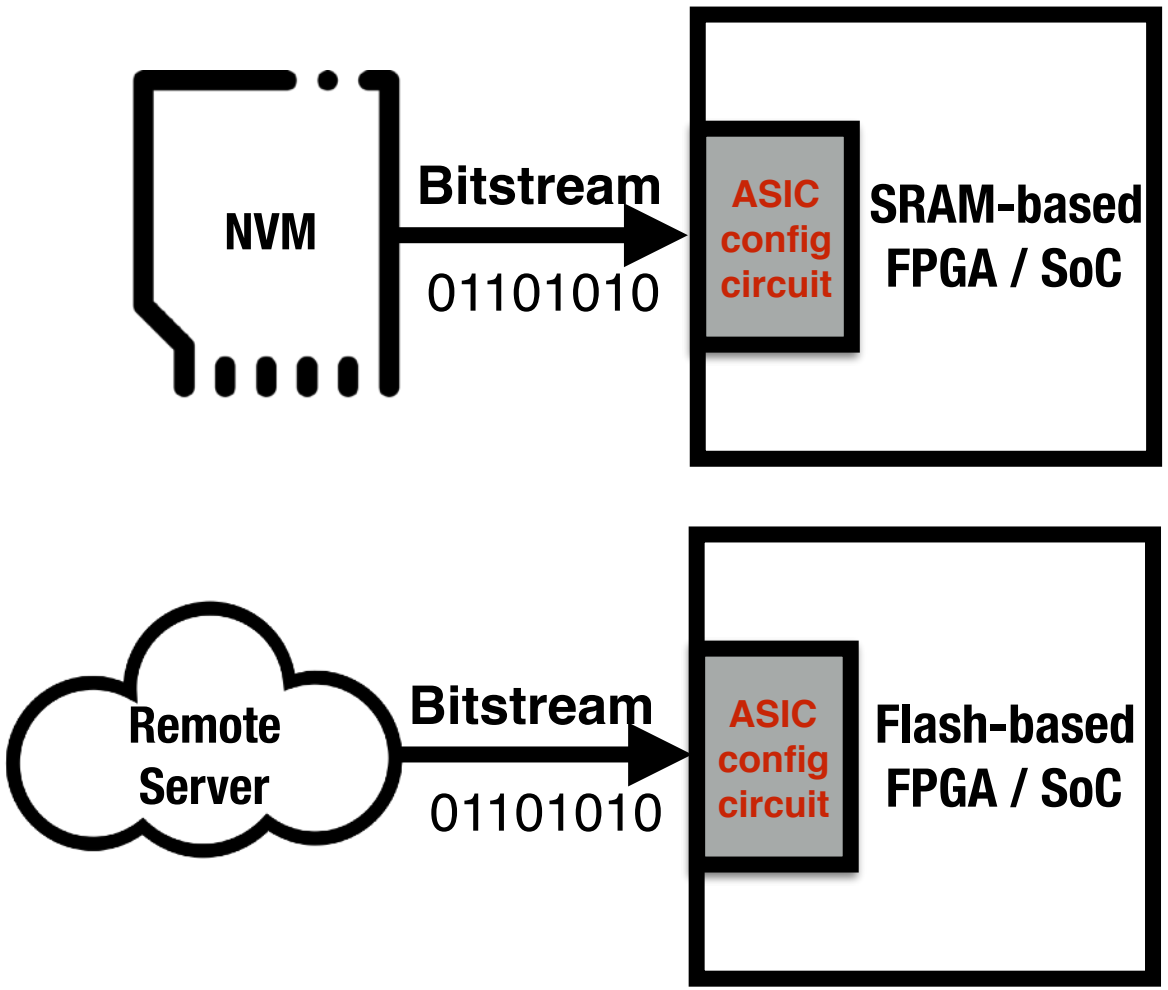
RO PUF for Xilinx Ultrascale+

Why Vendors Trust PUFs?

- The main assumption of attacks against PUFs: Access to the Challenge and Responses is available!
- Since the implemented soft or hard PUFs inside of FPGAs are controlled PUFs, where a non-invasive electrical access to the challenges and responses of the PUFs is restricted by either physical or algorithmic countermeasures, most of the reported modeling and semi-invasive attacks are ineffective.
- In this case the unprocessed challenges can be transmitted with the first stage boot loader to the FPGA.
- The response of the PUF will also be generated and processed inside the device and cannot be observed in a non-invasive way.

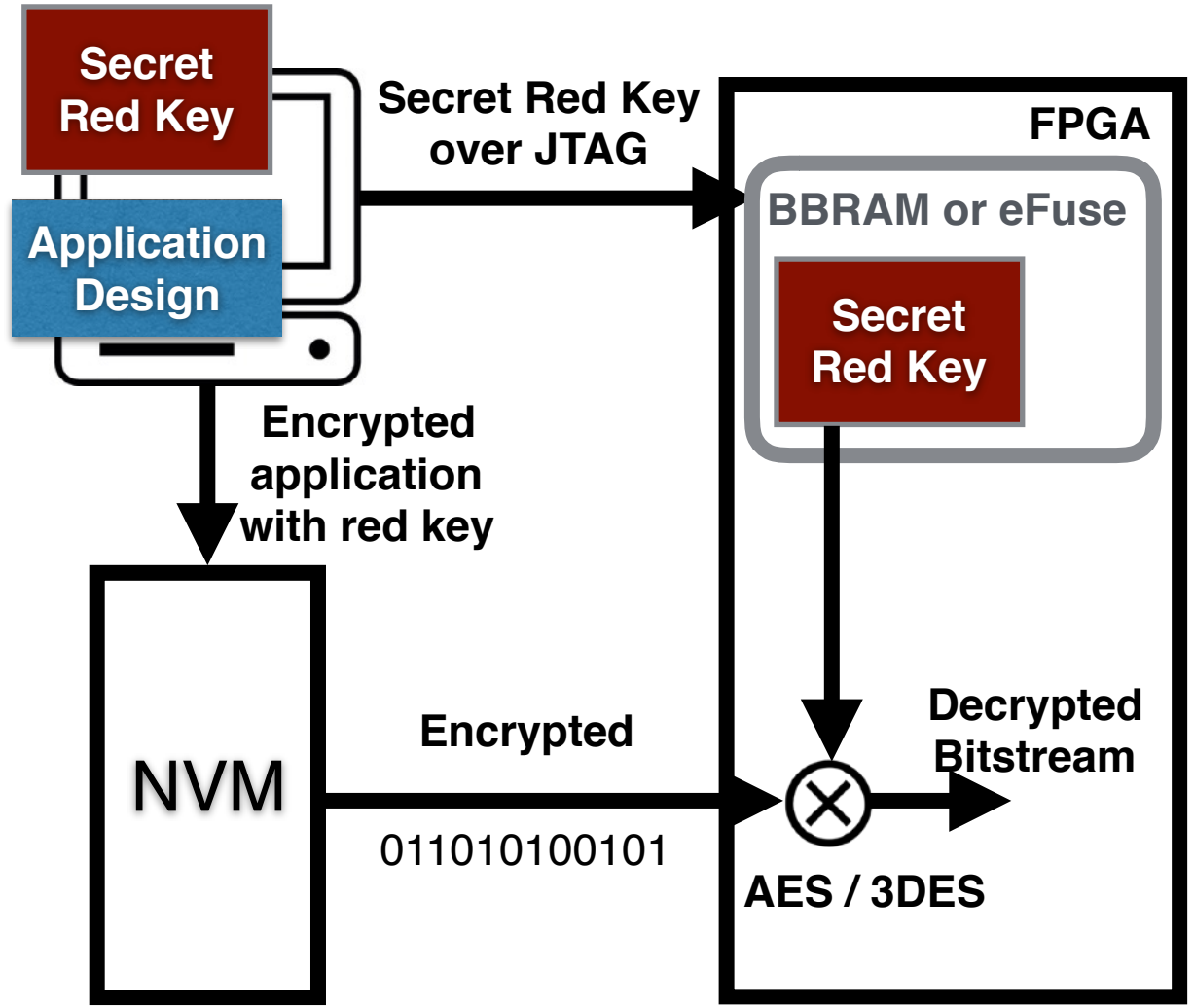
Security of FPGAs

- **Bitstream:** configuration data containing Intellectual Property (IP) and secrets for reconfigurable hardware
- The bitstream can be loaded in the field (**adversarial environment**)
- **Threats:** cloning, reverse-engineer, tampering or spoofing



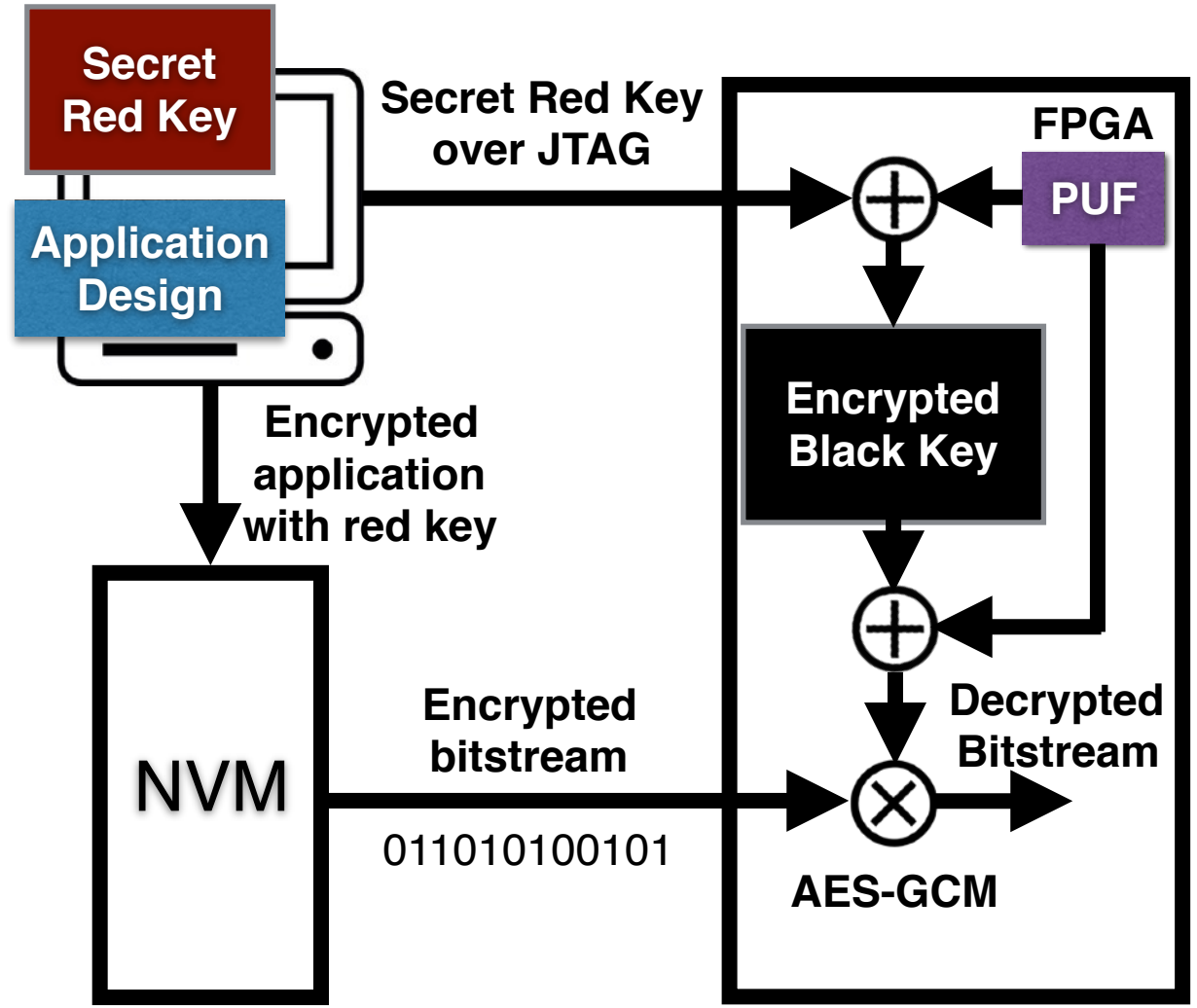
PUF-based Bitstream Configuration

- Loading the decryption key in **plaintext** into FPGA through JTAG
- Storing the red key in Battery Backed RAM (battery needed) or eFuses (no battery)
- Loading the encrypted bitstream in the field



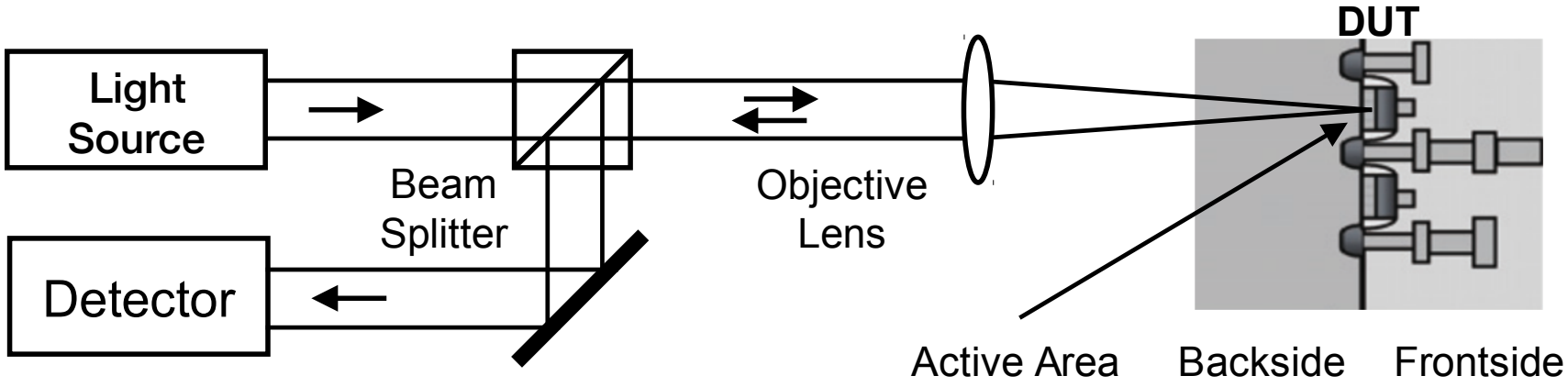
PUF-based Bitstream Configuration

- Loading the decryption key (Red) in **plaintext** into FPGA through JTAG
- Encrypting the Red Key with PUF responses to generate **Black Key**
- Storing Black Key and PUF configuration in NVM.
- Tampering Black Key by semi-invasive attacks does not divulge the red key



[1] Peterson, E.: White Paper WP468: Leveraging Asymmetric Authentication to En- hance Security-Critical Applications Using Zynq-7000 All Programmable SoCs. Xilinx, Inc. San Jose, CA (2015)

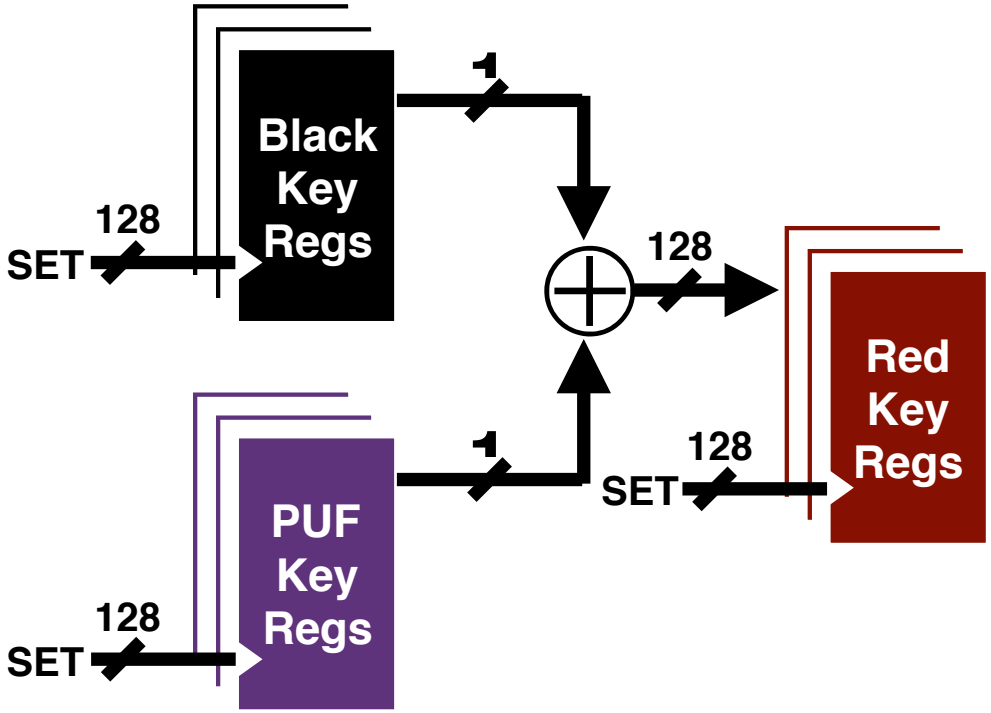
Optical Probing [1]

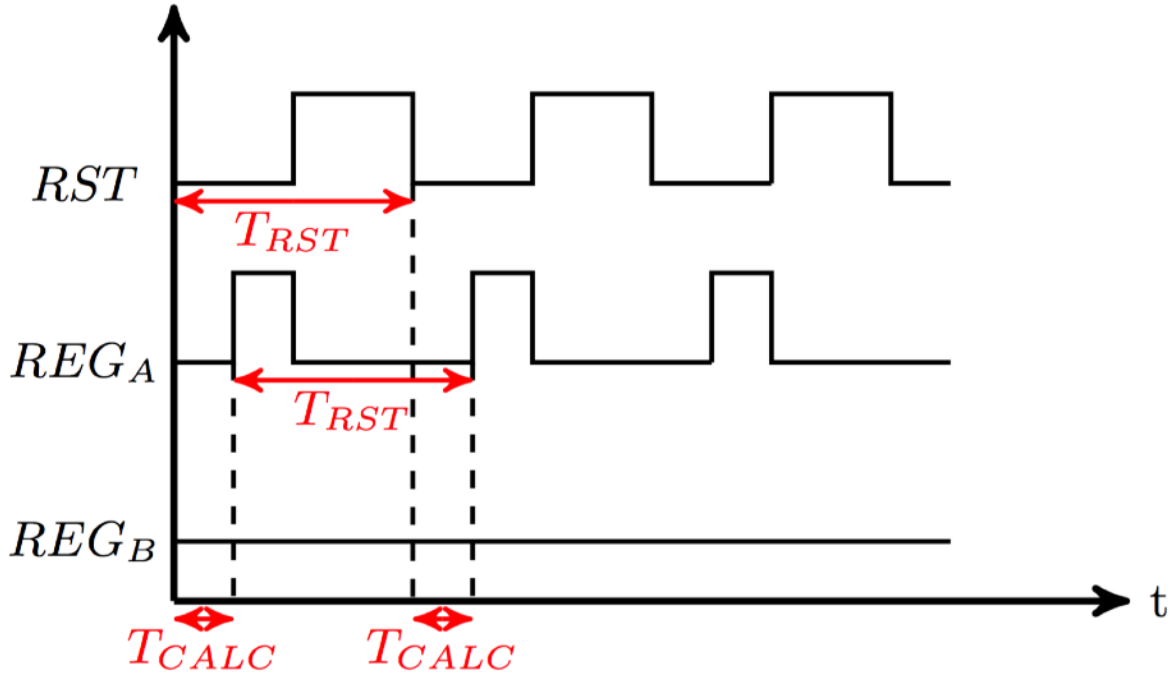


- Changes in the absorption coefficient and the refractive index of device in active area by electrical field and current.
- **Electro-Optical Probing (EOP)** or **Laser Voltage Probing (LVP)**: Optical beam intensity altered by voltage/current —> probing of electrical signals on the node
- **Electro-Optical Frequency Mapping (EOFM)** or **Laser Voltage Imaging (LVI)**: Feeding the reflected signal to a detector with a narrow band frequency filter while scanning the laser—> detecting node switching with this frequency

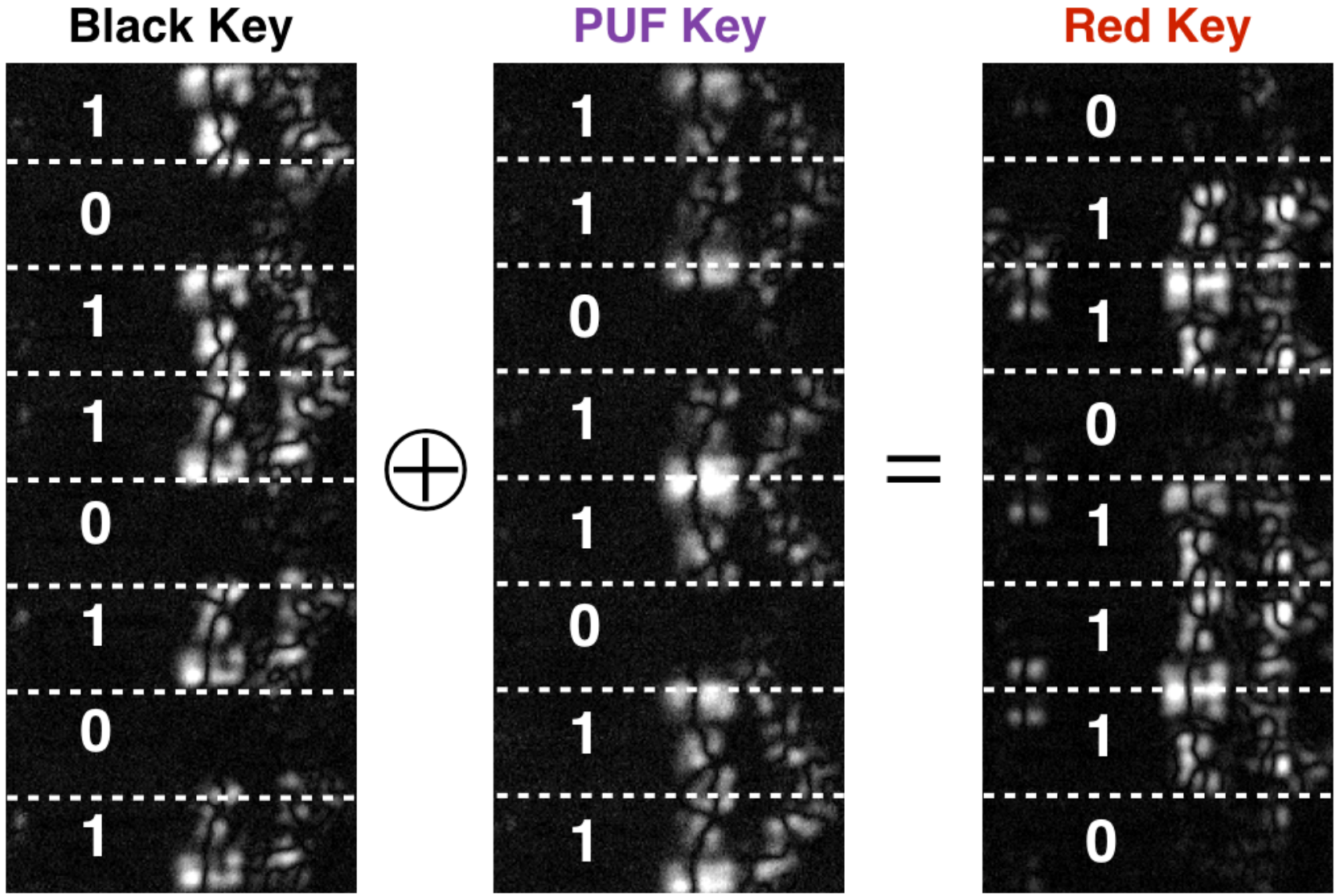
[1] Lohrke, H., Tajik, S., Boit, C., & Seifert, J. P. (2016, August). No place to hide: contactless probing of secret data on FPGAs. In *International Conference on Cryptographic Hardware and Embedded Systems* (pp. 147-167). Springer, Berlin, Heidelberg.

Key Extraction using LVI

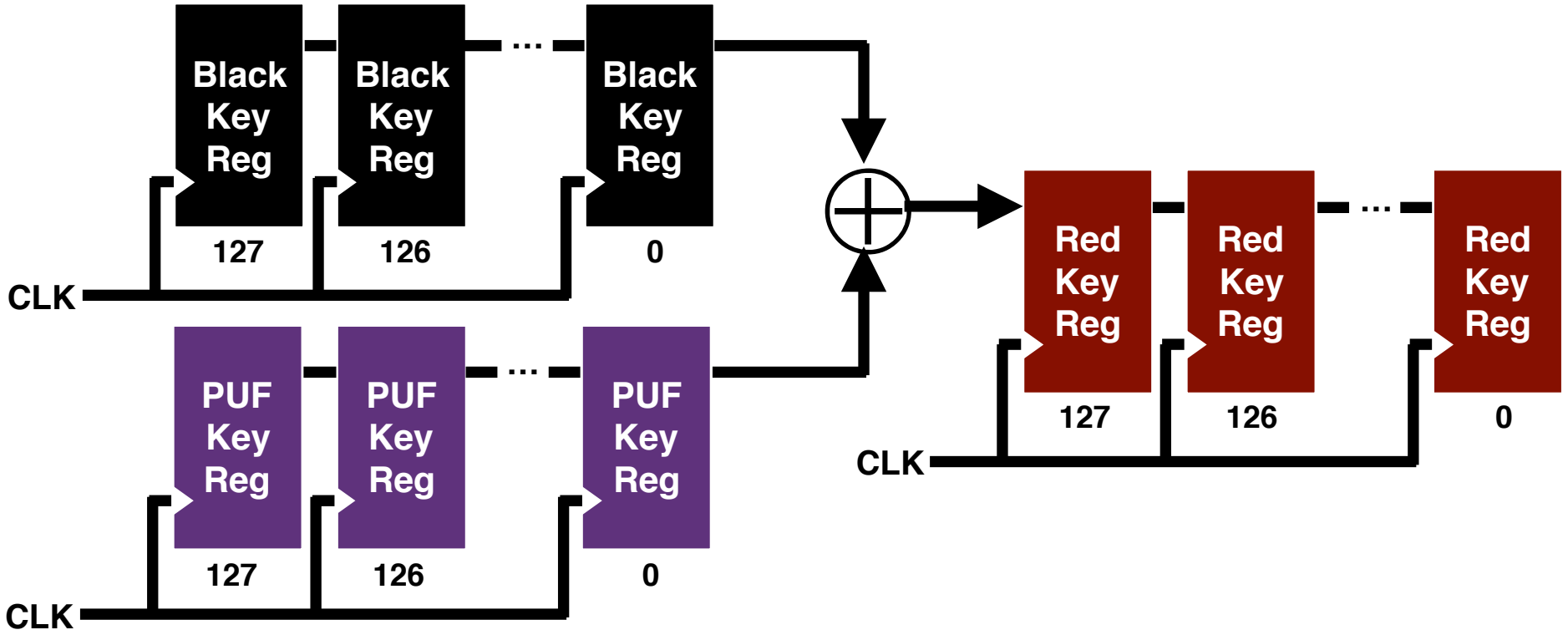




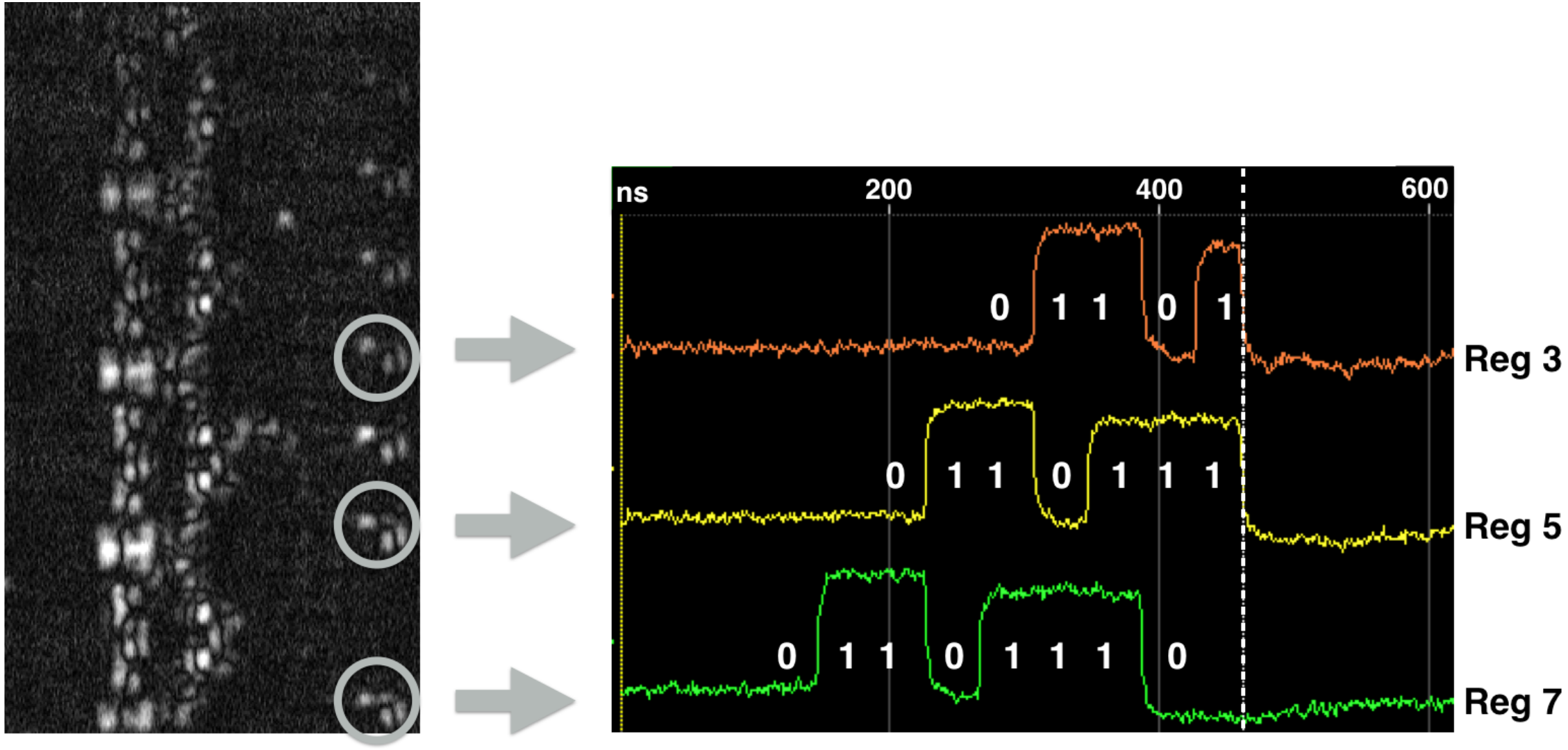
Key Extraction using LVI



Key Extraction using LVP



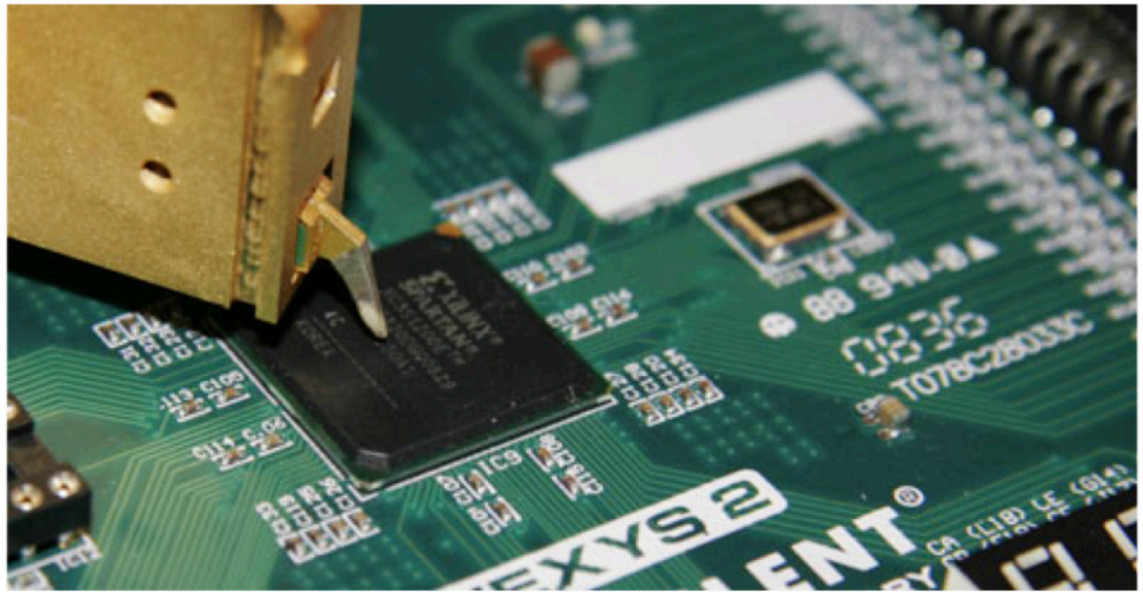
Key Extraction using LVP

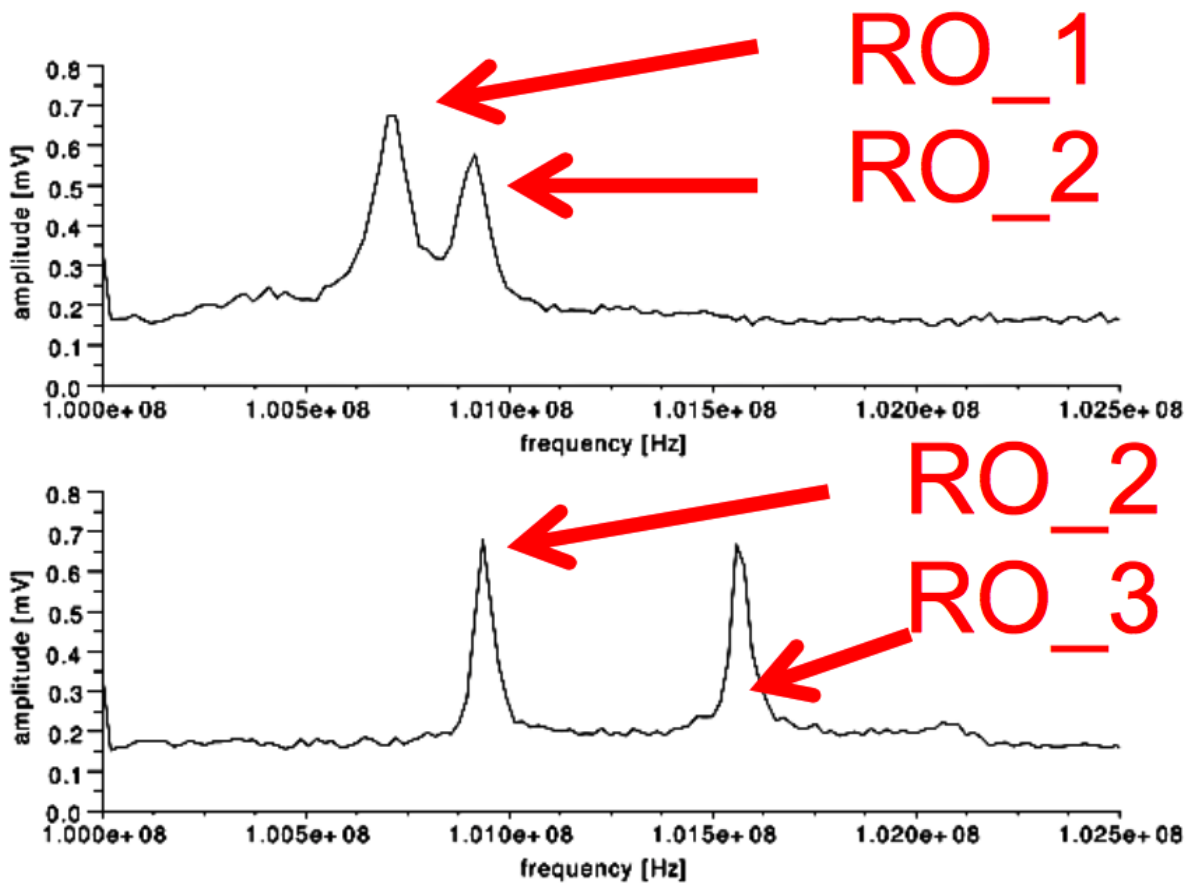


Side-Channel Analysis

EM Side-Channel Analysis of RO PUFs [1]

- Measuring the oscillation frequency of ROs by electromagnetic radiations
- **Assumption:** Attacker has access to CRPs

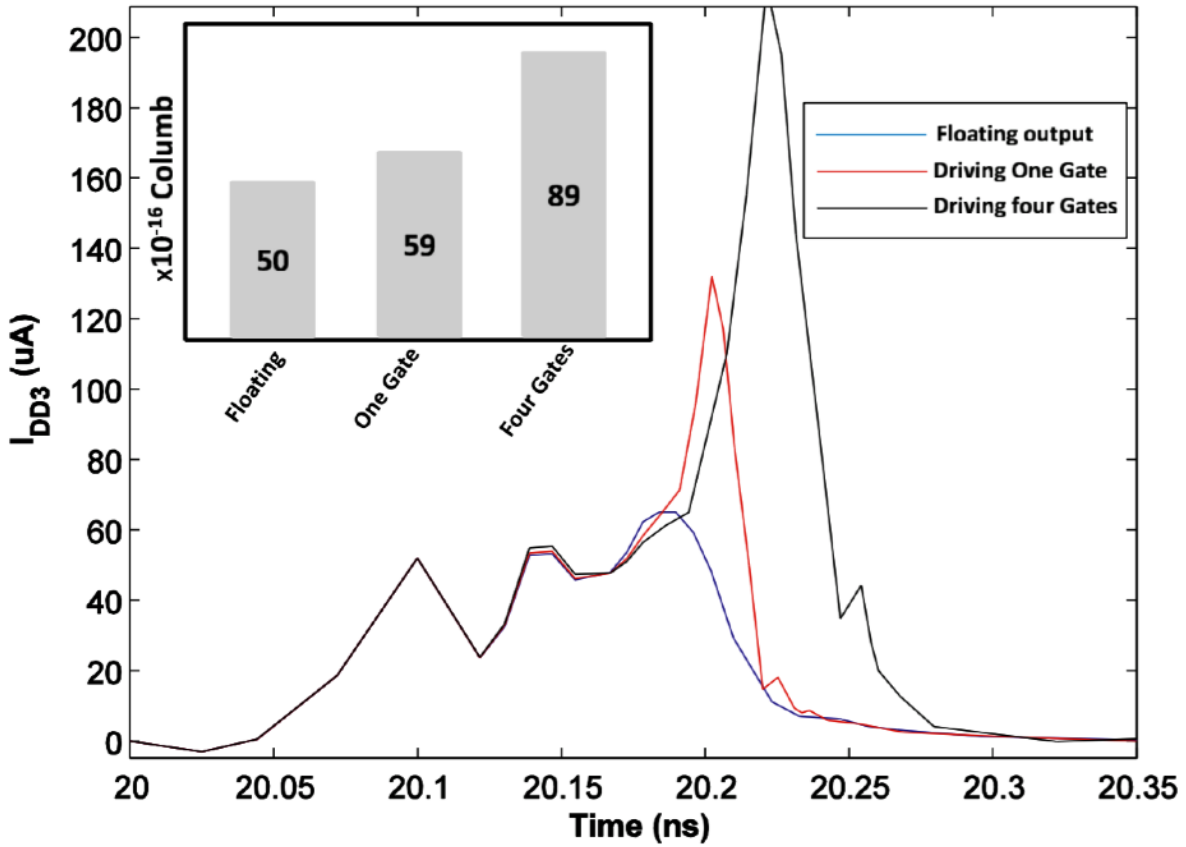




[1] Merli, Dominik, et al. "Side-channel analysis of PUFs and fuzzy extractors." International Conference on Trust and Trustworthy Computing. Springer, Berlin, Heidelberg, 2011.

Hybrid Attacks (SCA + ML) [1]

- Combination of ML techniques with SCA: Breaking up to 16 XOR arbiter PUF
- Power side channel
- Power consumption peaks when the flip-flops gives out 1
- Timing side channel

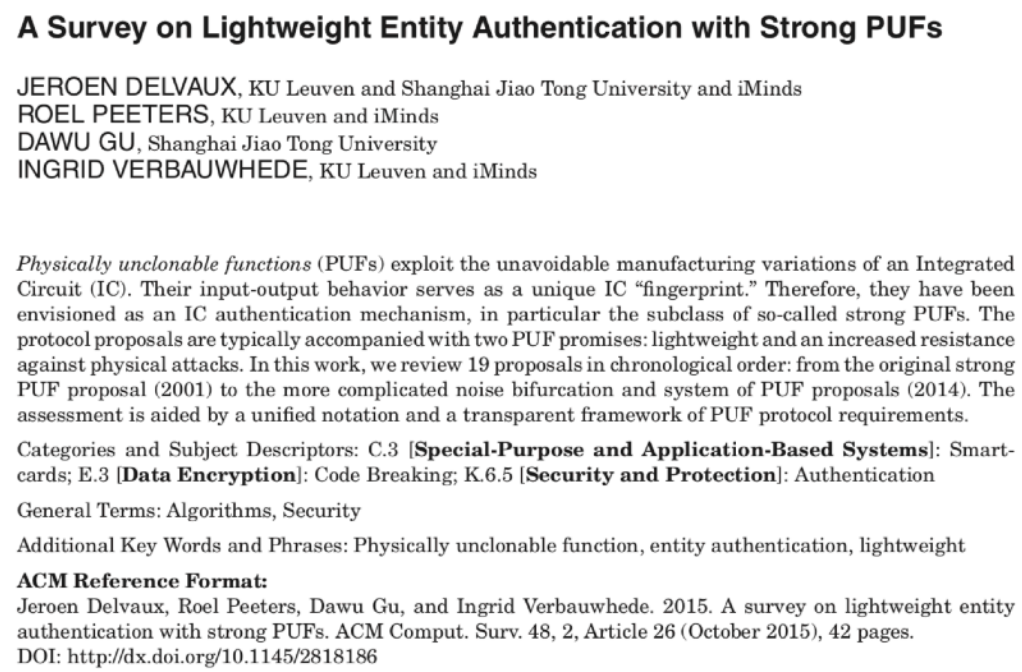


[1] Rührmair, Ulrich, et al. "Efficient power and timing side channels for physical unclonable functions." *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, Berlin, Heidelberg, 2014.

Protocol-level Countermeasures

Protocol-level Countermeasures

- Several protocol-level countermeasures against ML and physical attacks have been proposed, however, **almost all of them have been broken!**
- Very good security analysis of PUF protocols:



A Survey on Lightweight Entity Authentication with Strong PUFs

JEROEN DELVAUX, KU Leuven and Shanghai Jiao Tong University and iMinds
ROEL PEETERS, KU Leuven and iMinds
DAWU GU, Shanghai Jiao Tong University
INGRID VERBAUWHEDE, KU Leuven and iMinds

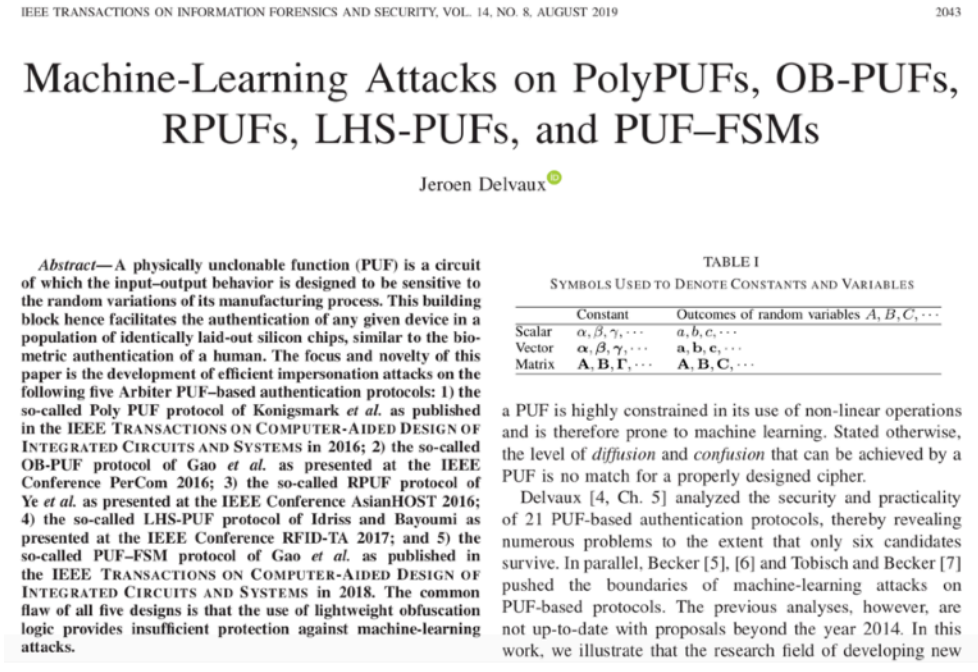
Physically unclonable functions (PUFs) exploit the unavoidable manufacturing variations of an Integrated Circuit (IC). Their input-output behavior serves as a unique IC "fingerprint." Therefore, they have been envisioned as an IC authentication mechanism, in particular the subclass of so-called strong PUFs. The protocol proposals are typically accompanied with two PUF promises: lightweight and an increased resistance against physical attacks. In this work, we review 19 proposals in chronological order: from the original strong PUF proposal (2001) to the more complicated noise bifurcation and system of PUF proposals (2014). The assessment is aided by a unified notation and a transparent framework of PUF protocol requirements.

Categories and Subject Descriptors: C.3 [**Special-Purpose and Application-Based Systems**]: Smart-cards; E.3 [**Data Encryption**]: Code Breaking; K.6.5 [**Security and Protection**]: Authentication

General Terms: Algorithms, Security

Additional Key Words and Phrases: Physically unclonable function, entity authentication, lightweight

ACM Reference Format:
Jeroen Delvaux, Roel Peeters, Dawu Gu, and Ingrid Verbauwhede. 2015. A survey on lightweight entity authentication with strong PUFs. ACM Comput. Surv. 48, 2, Article 26 (October 2015), 42 pages.
DOI: http://dx.doi.org/10.1145/2818186



Machine-Learning Attacks on PolyPUFs, OB-PUFs, RPUFs, LHS-PUFs, and PUF–FSMs

Jeroen Delvaux

***Abstract*— A physically unclonable function (PUF) is a circuit of which the input–output behavior is designed to be sensitive to the random variations of its manufacturing process. This building block hence facilitates the authentication of any given device in a population of identically laid-out silicon chips, similar to the biometric authentication of a human. The focus and novelty of this paper is the development of efficient impersonation attacks on the following five Arbiter PUF–based authentication protocols: 1) the so-called Poly PUF protocol of Konigsmark *et al.* as published in the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS in 2016; 2) the so-called OB-PUF protocol of Gao *et al.* as presented at the IEEE Conference PerCom 2016; 3) the so-called RPUF protocol of Ye *et al.* as presented at the IEEE Conference AsianHOST 2016; 4) the so-called LHS-PUF protocol of Idriss and Bayoumi as presented at the IEEE Conference RFID-TA 2017; and 5) the so-called PUF–FSM protocol of Gao *et al.* as published in the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS in 2018. The common flaw of all five designs is that the use of lightweight obfuscation logic provides insufficient protection against machine-learning attacks.**

TABLE I
SYMBOLS USED TO DENOTE CONSTANTS AND VARIABLES

	Constant	Outcomes of random variables $A, B, C, \cdots$
Scalar	$\alpha, \beta, \gamma, \cdots$	$a, b, c, \cdots$
Vector	$\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \cdots$	$\mathbf{a}, \mathbf{b}, \mathbf{c}, \cdots$
Matrix	$\mathbf{A}, \mathbf{B}, \boldsymbol{\Gamma}, \cdots$	$\mathbf{A}, \mathbf{B}, \mathbf{C}, \cdots$

a PUF is highly constrained in its use of non-linear operations and is therefore prone to machine learning. Stated otherwise, the level of *diffusion* and *confusion* that can be achieved by a PUF is no match for a properly designed cipher.

Delvaux [4, Ch. 5] analyzed the security and practicality of 21 PUF-based authentication protocols, thereby revealing numerous problems to the extent that only six candidates survive. In parallel, Becker [5], [6] and Tobisch and Becker [7] pushed the boundaries of machine-learning attacks on PUF-based protocols. The previous analyses, however, are not up-to-date with proposals beyond the year 2014. In this work, we illustrate that the research field of developing new

Lockdown Protocol

- Restricting the feeding of arbitrary challenges to the PUF.
- Preventing repeatable feeding of the same challenge to prevent side-channel leakage.

146 IEEE TRANSACTIONS ON MULTI-SCALE COMPUTING SYSTEMS, VOL. 2, NO. 3, JULY-SEPTEMBER 2016

A Lockdown Technique to Prevent Machine Learning on PUFs for Lightweight Authentication

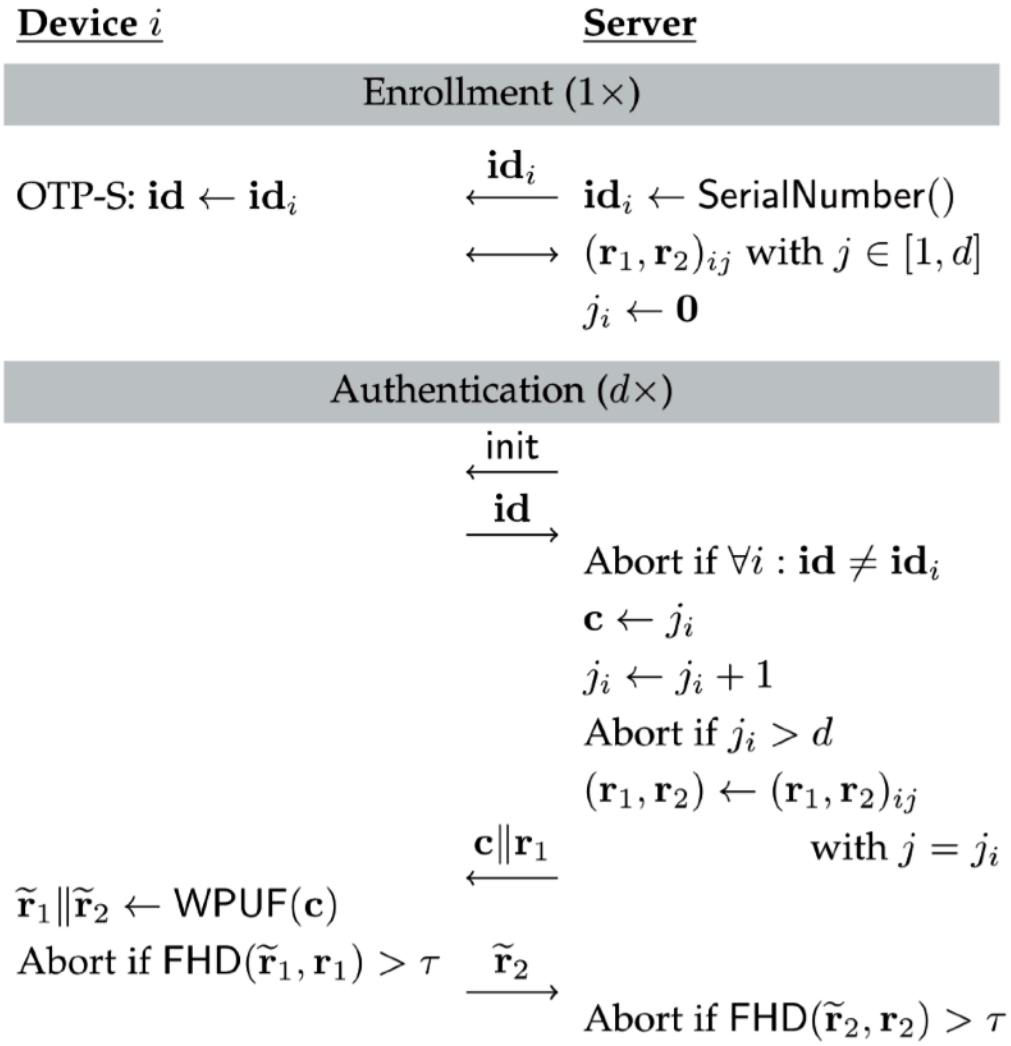
Meng-Day (Mandel) Yu, *Member, IEEE*, Matthias Hiller, Jeroen Delvaux,
Richard Sowell, Srinivas Devadas, *Fellow, IEEE*, and Ingrid Verbauwhede, *Fellow, IEEE*

Abstract—We present a lightweight PUF-based authentication approach that is practical in settings where a server authenticates a device, and for use cases where the number of authentications is limited over a device's lifetime. Our scheme uses a server-managed challenge/response pair (CRP) *lockdown* protocol: unlike prior approaches, an adaptive chosen-challenge adversary with machine learning capabilities *cannot* obtain new CRPs without the server's implicit permission. The adversary is faced with the problem of deriving a PUF model with a limited amount of machine learning training data. Our system-level approach allows a so-called strong PUF to be used for lightweight authentication in a manner that is *heuristically secure* against today's best machine learning methods through a worst-case CRP exposure algorithmic validation. We also present a degenerate instantiation using a weak PUF that is secure against *computationally unrestricted* adversaries, which includes *any* learning adversary, for practical device lifetimes and read-out rates. We validate our approach using silicon PUF data, and demonstrate the feasibility of supporting 10, 1,000, and 1M authentications, including practical configurations that are not learnable with polynomial resources, e.g., the number of CRPs and the attack runtime, using recent results based on the *probably-approximately-correct* (PAC) complexity-theoretic framework.

Index Terms—Physical unclonable function, authentication, machine learning, heuristic security, computationally unrestricted adversary, probably approximately correct (PAC) learning

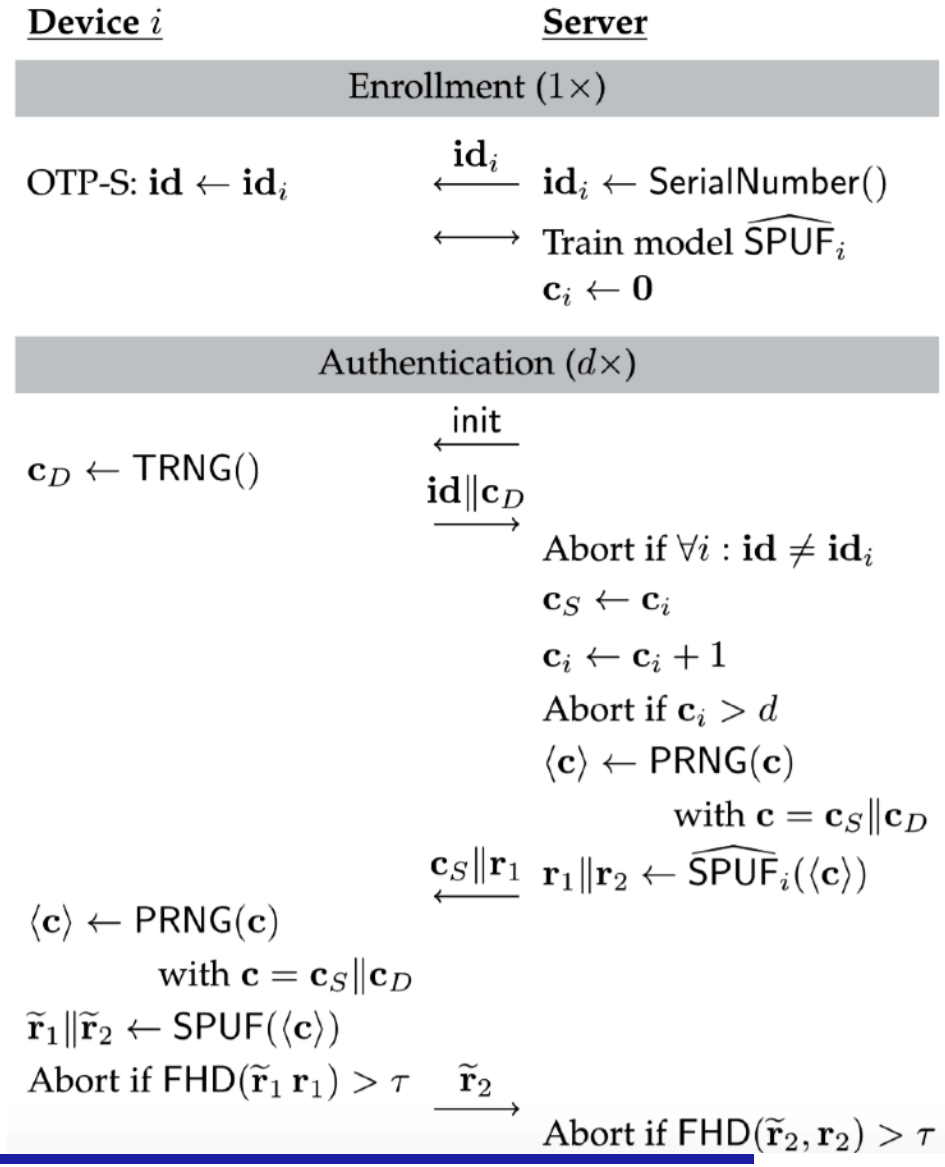
Lockdown Protocol 1

- Due to the lockdown, the adversary cannot issue a not-yet- seen packet c||r1 (not-yet-released by the server) and get the corresponding returning response packet r2'; as a result, the challenges can be deterministically generated, e.g., using a counter. Every authentication is unique since the challenge is non-repeating, based on a server-side counter.
- **WPUF:** Weak PUF like SRAM or RO PUF



Lockdown Protocol 2

- During an authentication event, the server obtains a device identifier id packet which now also includes a challenge c_D from the device; the device-side challenge is to allow a challenge exchange, so neither the device nor the server can unilaterally determine all the bits of <c>.
- It prevent repeated measurements to, for example, obtain photonic information
- **SPUF:** Strong PUF like XOR Arbiter PUF
- Server keeps a simulated SPUF (In enrollment phase, individual chains of an XOR arbiter PUF can be modelled by giving access to response of all chains. The simulated model will be stored in the database)



Thank you