Physically Uncloneable Functions in the Universal Composition Framework

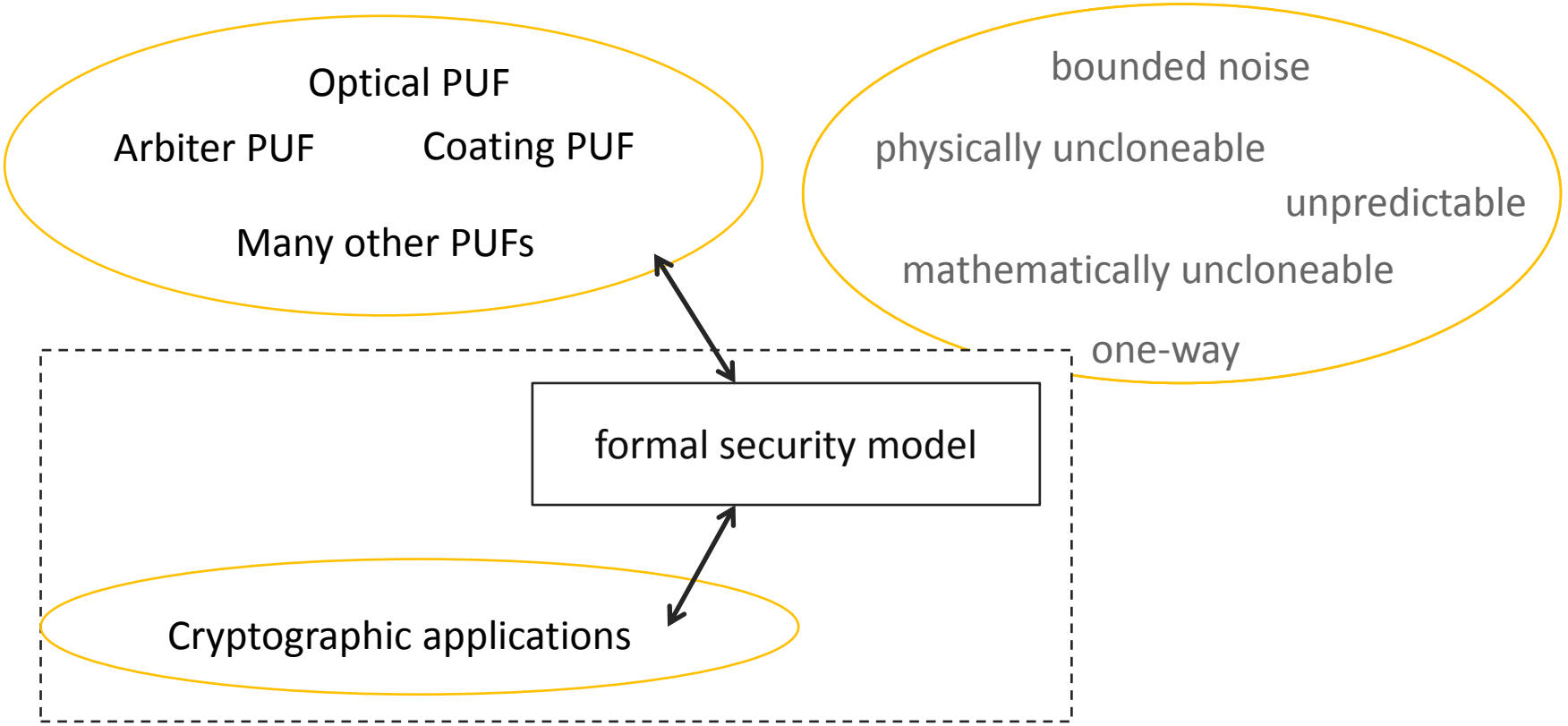
Christina Brzuska
Marc Fischlin
Heike Schröder
Stefan Katzenbeisser

Cryptoplexity
Cryptography & Complexity Theory
Technische Universität Darmstadt
www.cryptoplexity.de

CASED

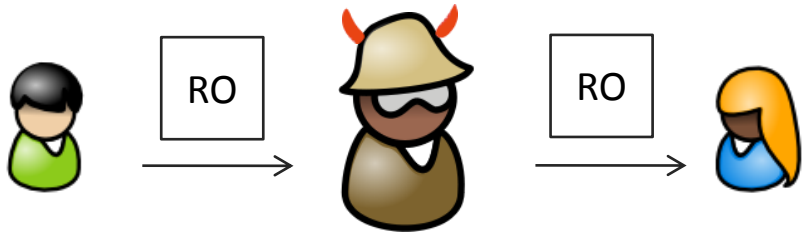
TECHNISCHE UNIVERSITÄT DARMSTADT

Security of PUFs



Applications + nice surprise

- Key agreement
- Oblivious Transfer
- Commitments
  - OT → COM at cost of one round
- All protocols UC-secure
- Canetti, Fischlin [CF01]: ~~UC-secure commitments~~



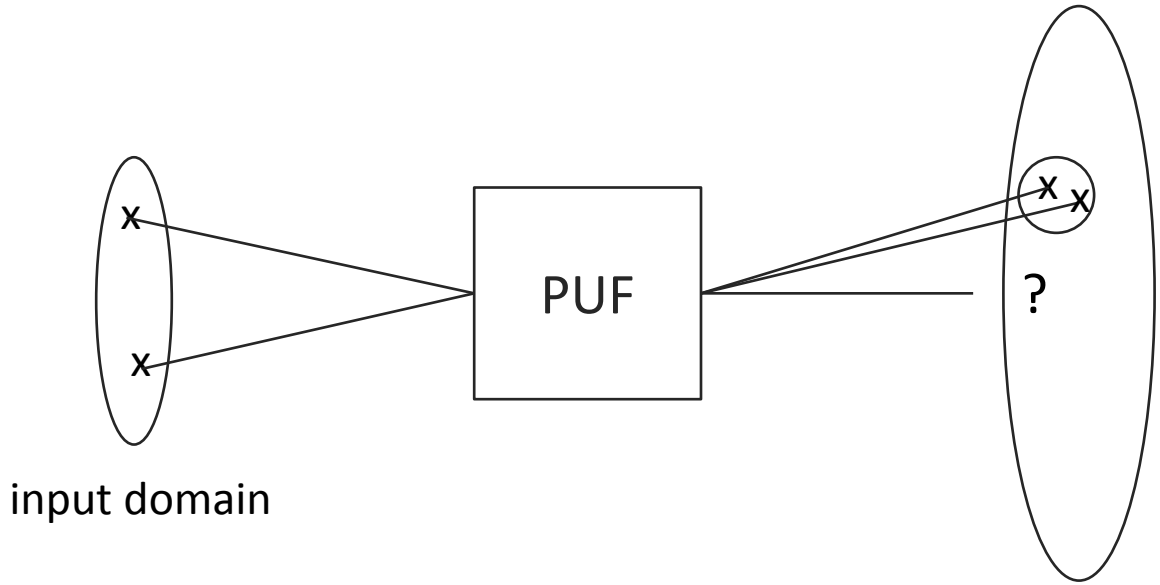
"random function in a box"

PUF ≈ NPRO

[CF01] extends to NPRO

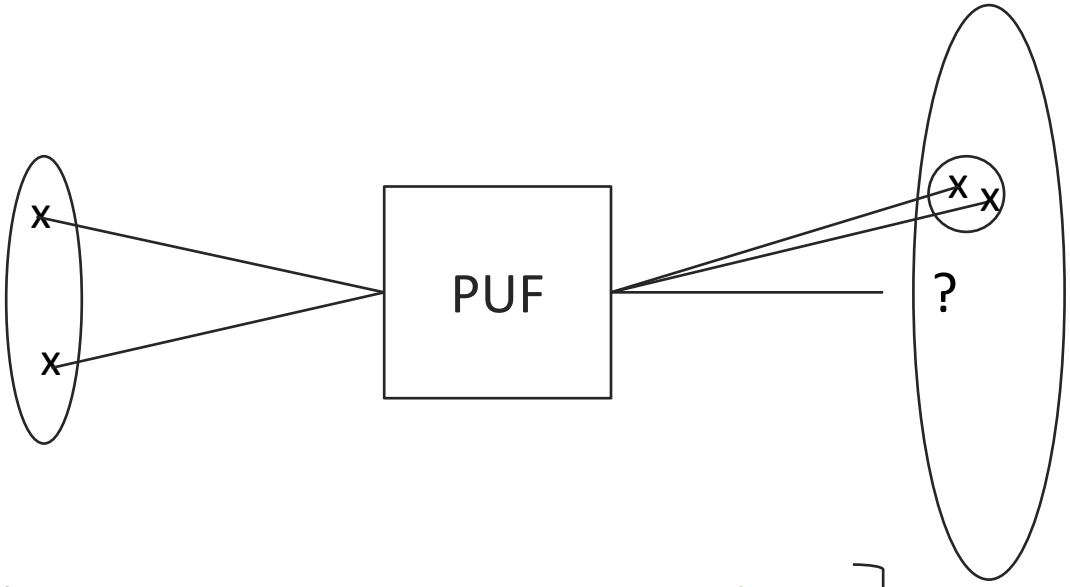
- UC-secure commitment scheme from PUFs (without cryptographic assumptions)

Physically Uncloneable Functions (PUF)



- Fuzzy measurement with bounded noise
- High(er) entropy for fresh challenge values

Physically Uncloneable Functions (PUF)



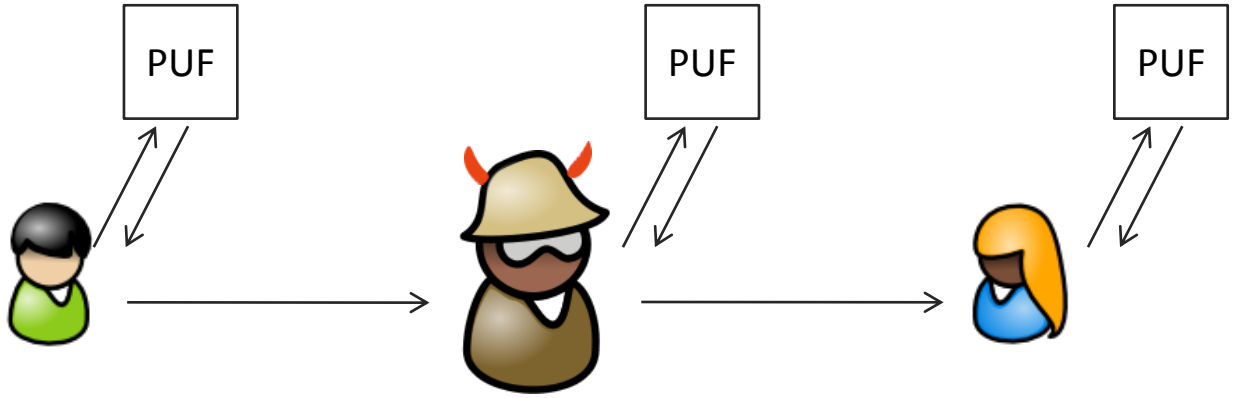
- Bounded noise — ok
- High(er) entropy for fresh challenge values — ok

} Our Definition

- Superpolynomially big input domain — some PUFs

} Needed for applications

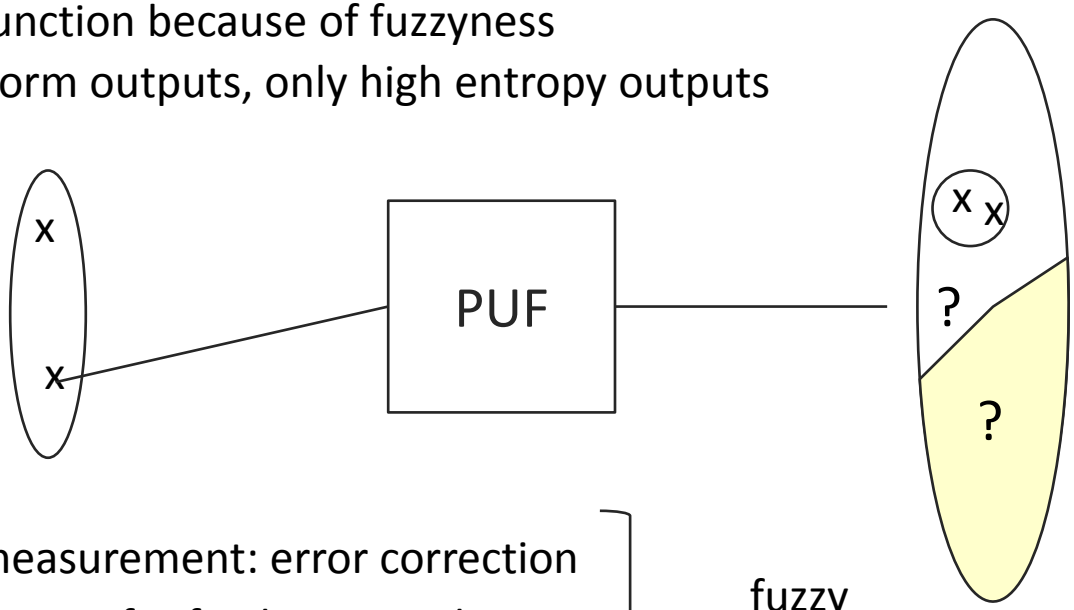
Attack Model



- If input domain is small, the adversary can measure the whole PUF
- Small input domain can be used for key storage, weaker attack model

PUF → NPRO: Fuzzy Extractors [DRS04]

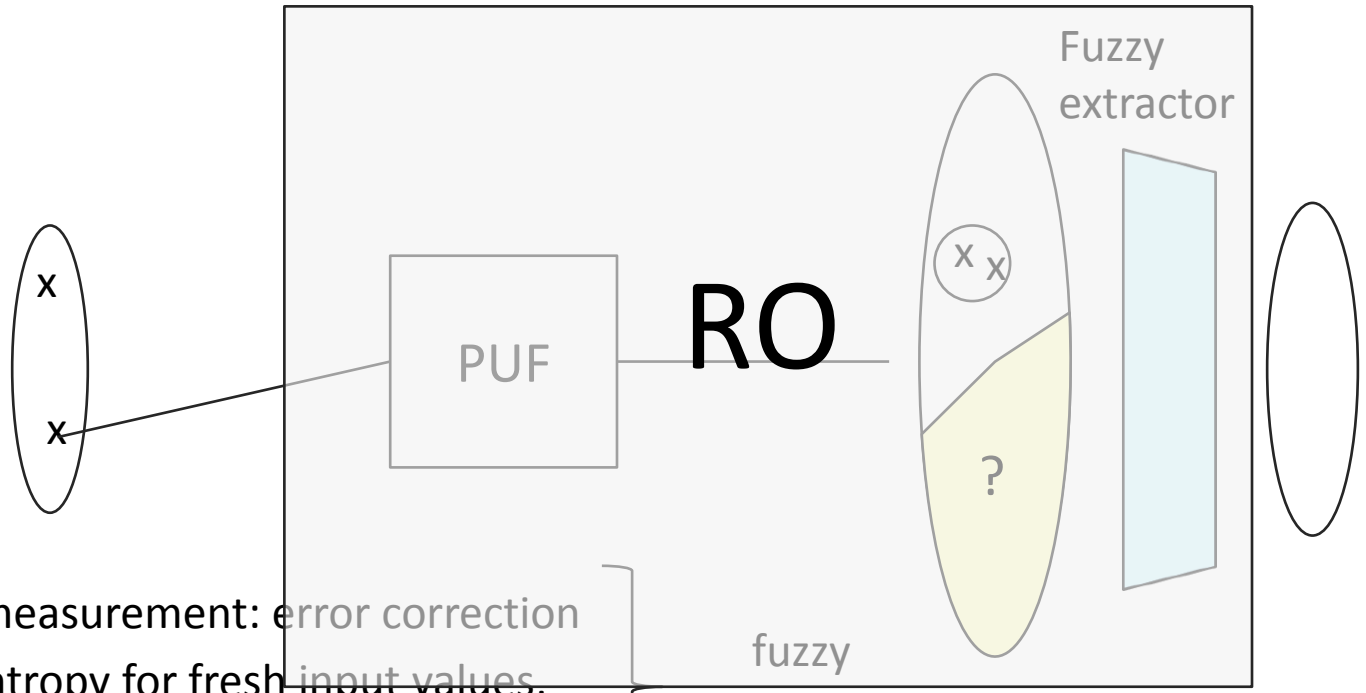
- Goal: random function in a box
- Not a function because of fuzzyness
- No uniform outputs, only high entropy outputs



- Fuzzy measurement: error correction
- High entropy for fresh input values, but not uniform: smooth out entropy

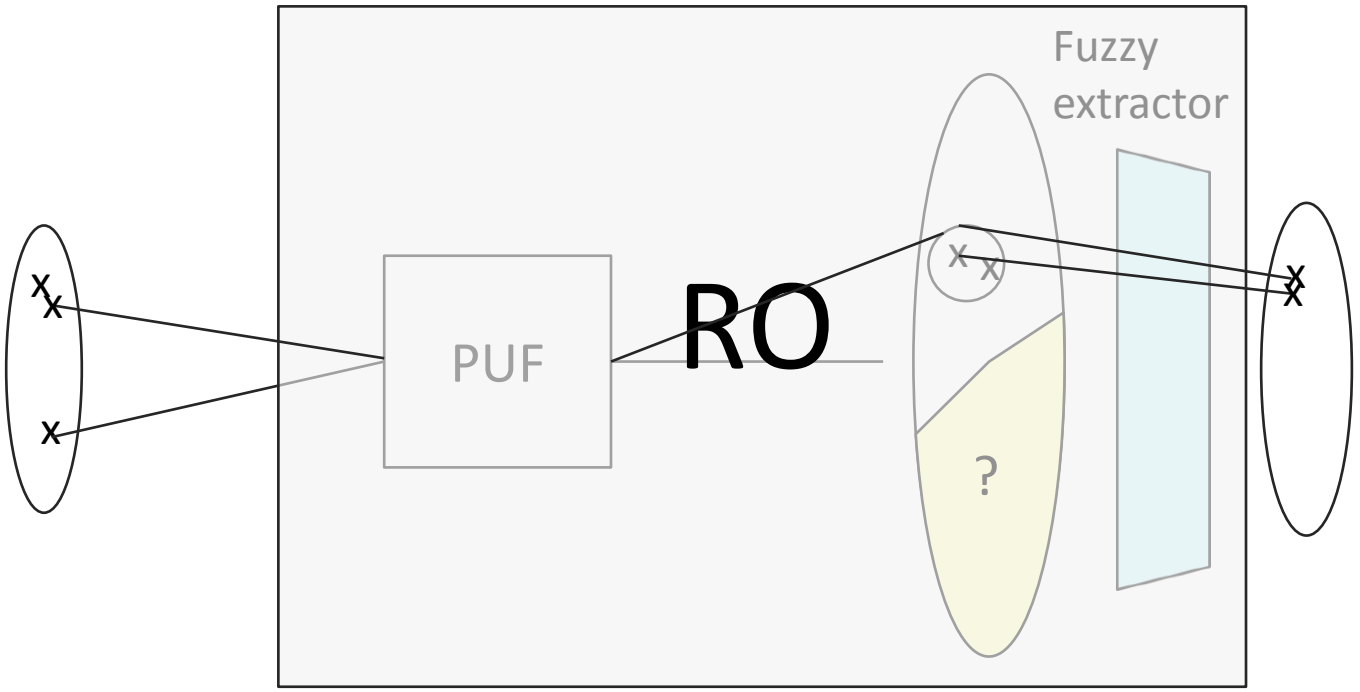
fuzzy extractor

PUF → NPRO: Fuzzy Extractors [DRS04]



- Fuzzy measurement: error correction
- High entropy for fresh input values, but not uniform: smooth out entropy

PUF → NPRO: Fuzzy Extractors [DRS04]

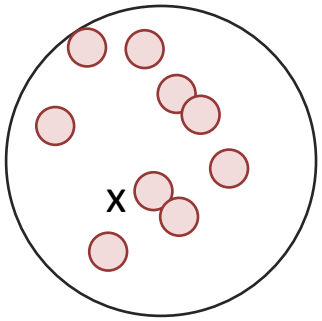


- Almost: If two input values are close, output values might still related
- It two input values are far away, outputs are uniform and independent

3 Main Properties for Application

- Correctness: Fuzzy(PUF(.)) is a mathematical function
- Well-Spread Domain:

input domain

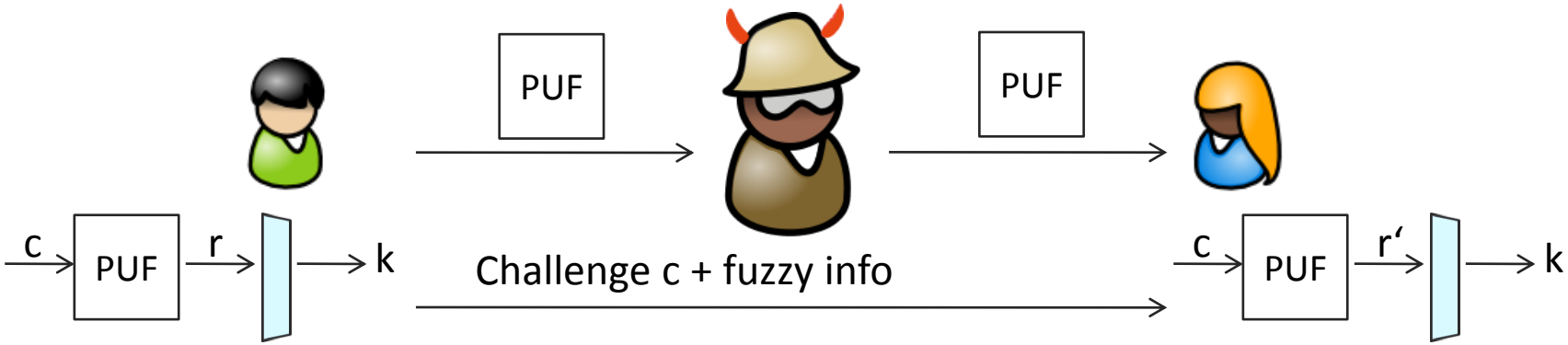


random inputs value is usually “far away“ from previous ones

- Uniform outputs:

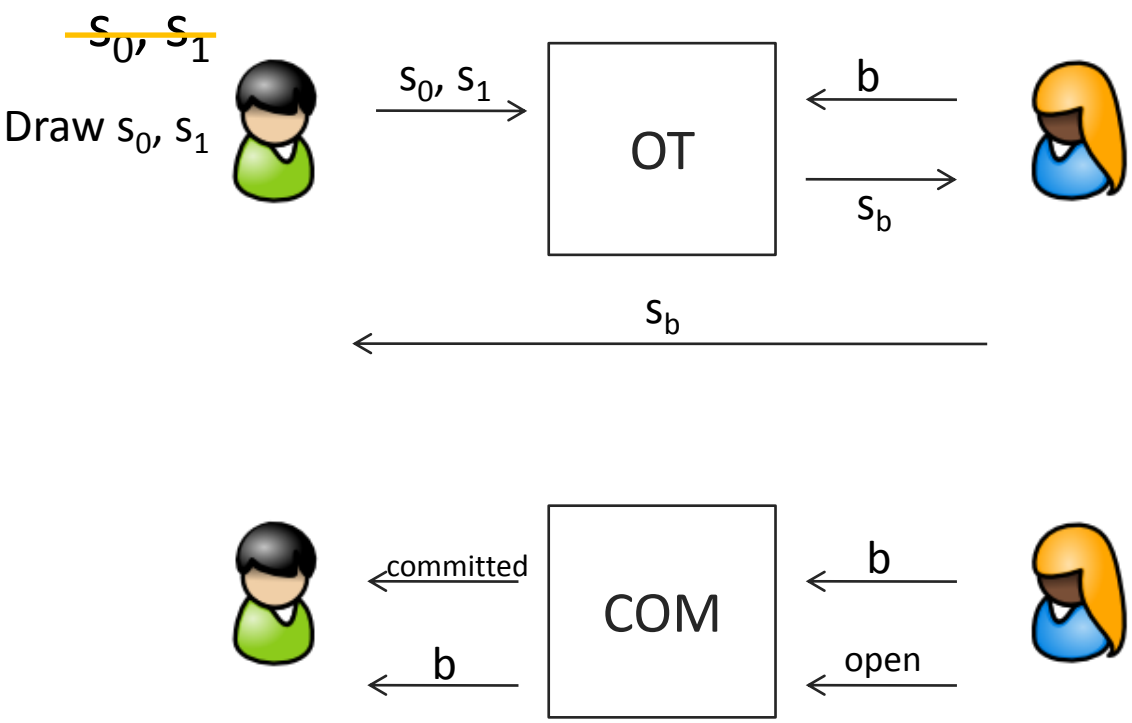
  For values, that are far away from previous ones, the output is uniform (probability over: PUF generation, evaluation, fuzzy extractor)

Warm-Up: Key Agreement



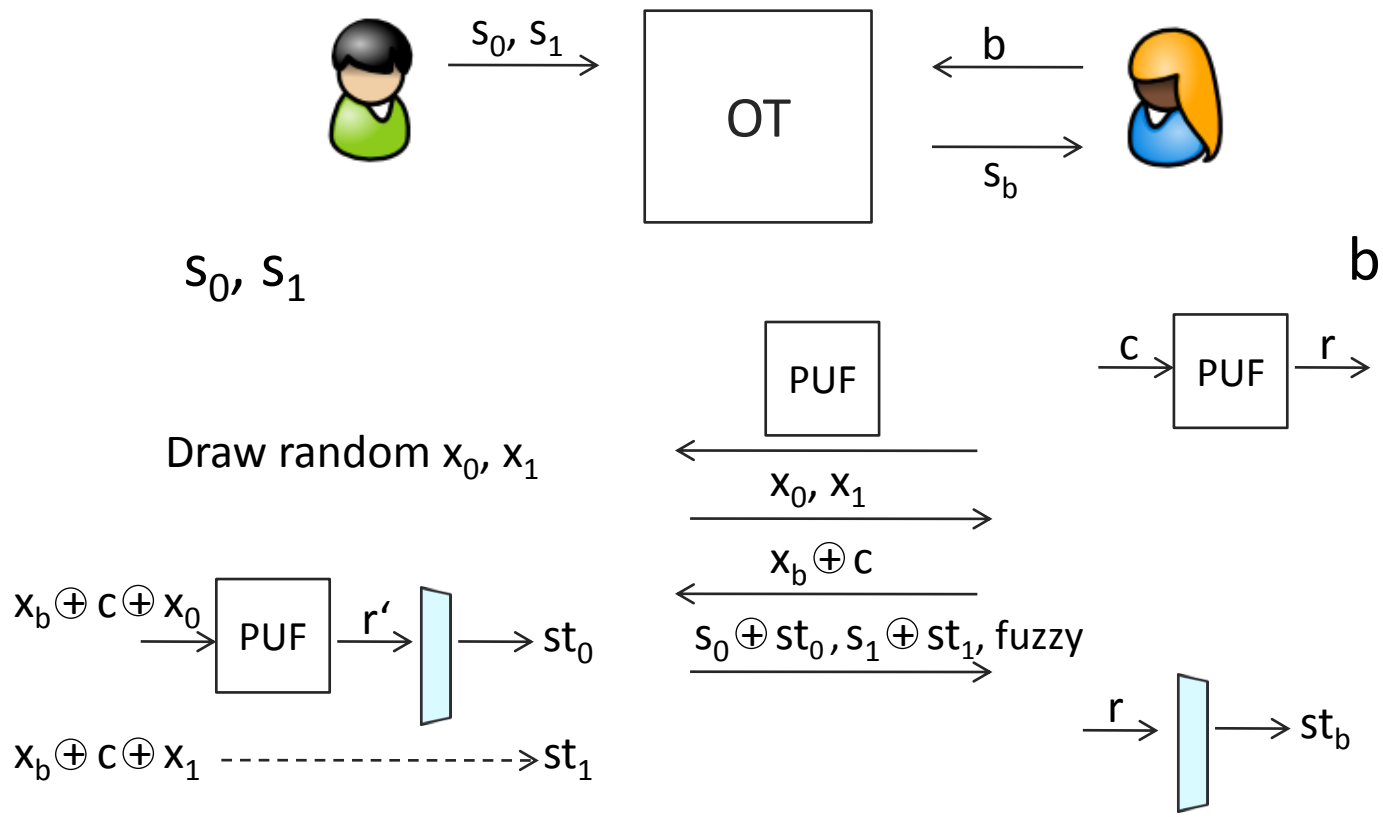
- They compute the same key due to **correctness** property of Fuzzy(PUF(.)).
- When adversary measures PUF, c is information-theoretically hidden. Due to the **well-spread domain property**, adversary only queries about values that are far away from c.
- Therefore, the value k is random from the point of view of the adversary due to the **uniform outputs** property.

OT → COM

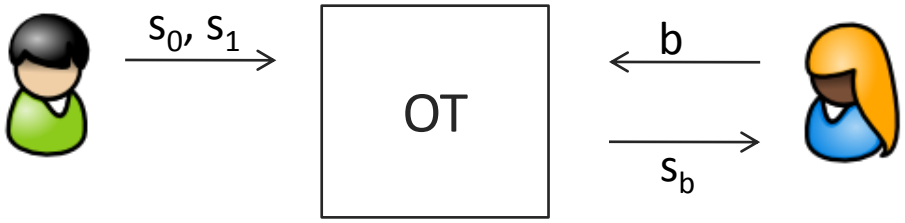


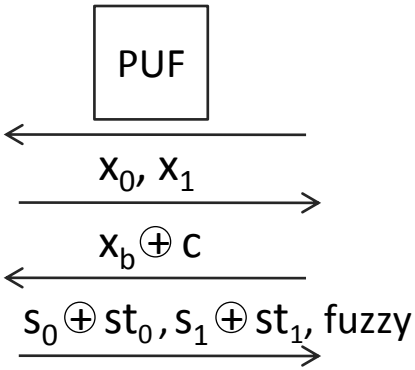
- Bit b is secret.
- Receiver learns only s_b
- Main idea:
  OT receiver is
  COM sender

Oblivious Transfer (OT)



Oblivious Transfer (OT)





- Security against receiver via PUF properties
- Security against sender information-theoretic.
- Hence, the resulting **commitment** scheme is secure against adaptive corruptions

Summary

- For active adversaries, we need PUFs with:
  - Bounded noise
  - High(er) entropy for fresh challenge values
  - Superpolynomially big input domain
- Properties of PUF+fuzzy extractor:
  - Correctness
  - Well-spread domain
  - Uniform outputs
- Get efficient provably secure protocols without cryptographic assumptions
  - Key Agreement
  - Oblivious Transfer
  - Commitments: one-round transformation from OT
- PUF + Fuzzy $\approx$ NPRO, but apparently, one can do even more with PUFs

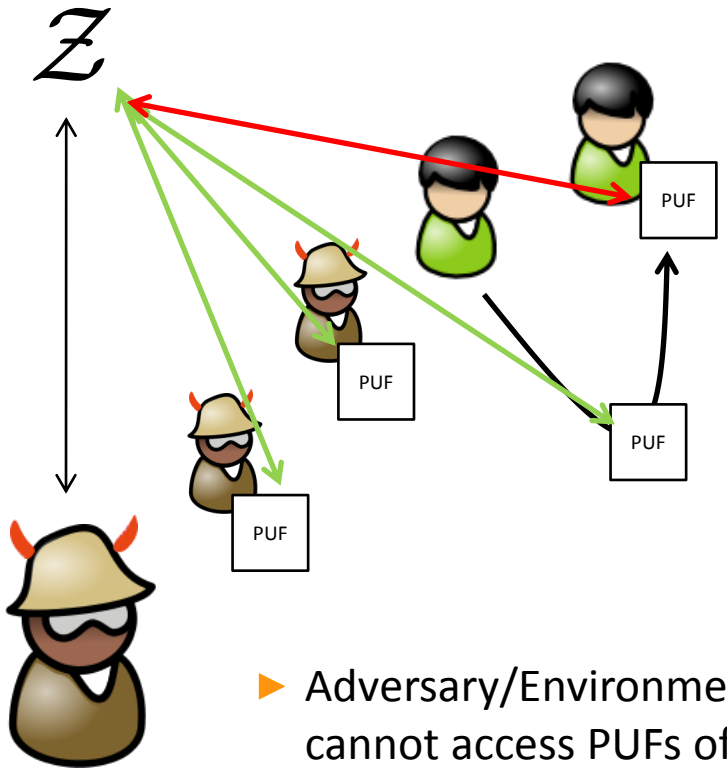
Thank you

Marc Fischlin

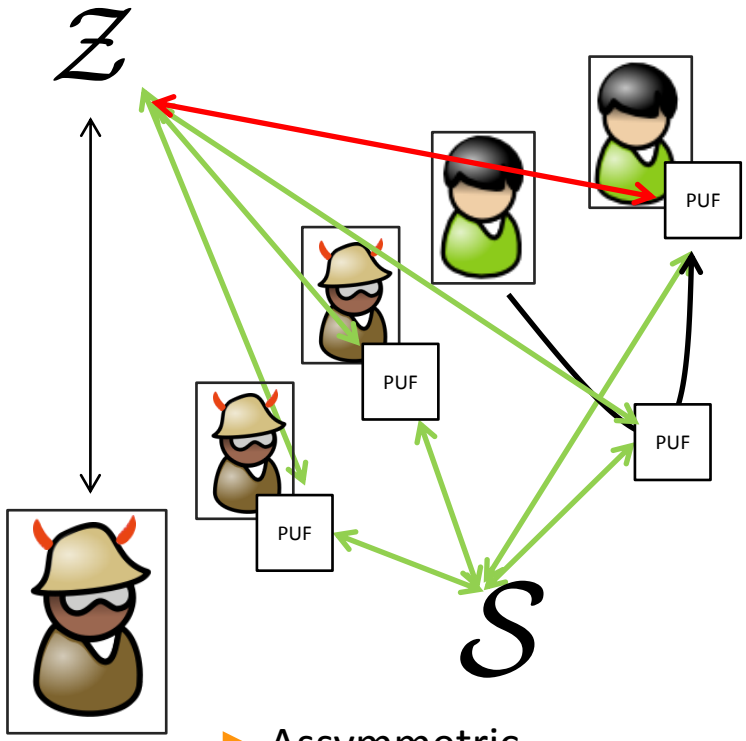
Heike Schröder

Stefan Katzenbeisser

On NPRO and PUFs in UC



- Adversary/Environment cannot access PUFs of honest parties



- Assymetric
- No programming abilities