PUF Protocols

Presented By
Urbi Chatterjee

Under the guidance of
Dr. Rajat Subhra Chakraborty
and
Dr. Debdeep Mukhopadhyay

Indian Institute of Technology Kharagpur.
Date:3rd February,2015

Contents

Introduction to PUF

- **Problem:**

  Storing **digital** information in a device in a way that is resistant to **physical attack** is difficult and expensive.

IBM 4758

Tamper-proof package containing a secure processor which has a secret key and memory

Tens of sensors, resistance, temperature, voltage, etc.

Continually battery-powered

~ $3000 for a 99 MHz processor and 128MB of memory

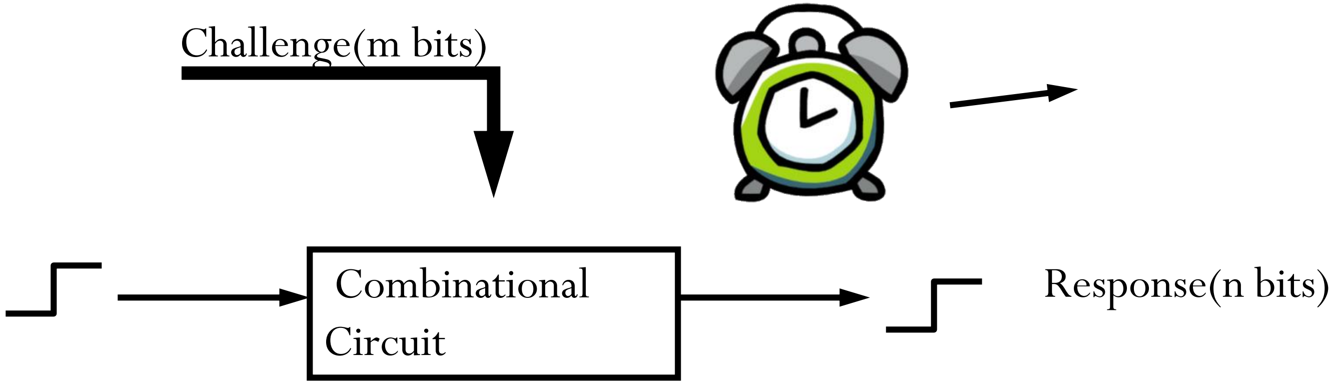
Introduction to PUF(Contd…)

Definition: Silicon PUF is a physical entity that is:

- Embodied in a physical system.
- Easy to evaluate.
- Hard to predict.
- Hardware equivalent to one way function.
- The functional mapping between input and output is instance-specific.

Proof of Concept

- Because of process variations, **no two Integrated Circuits are identical**
- Experiments in which **identical circuits with identical layouts** were placed on different FPGAs show that path delays vary enough across ICs to use them for identification.

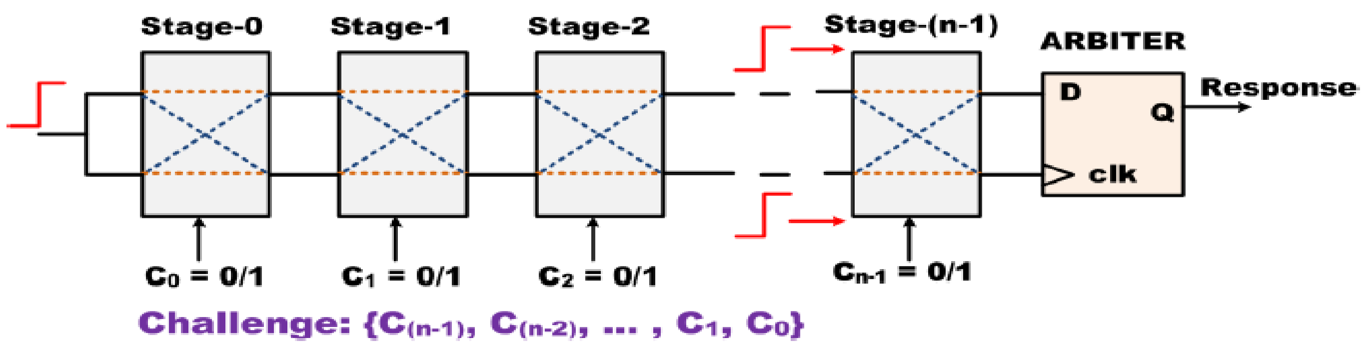


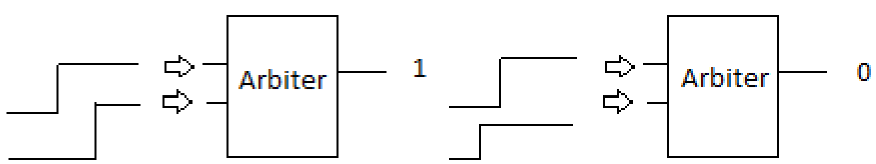
Desirable Physical Properties of PUF

- Large number of Challenge-Response Pair(CRP)
- Reliability
- Uniqueness
- Physical Unclonability
- Mathematical Unclonability

A Candidate Silicon PUF(Arbiter PUF)

- In APUF, each challenge creates two paths through the circuit that are excited simultaneously. The digital response is based on a **(timing) comparison of the path delays**.
- Path delays in an IC are **statistically distributed** due to random manufacturing variations.





Lim,D.,2004

Application of PUF

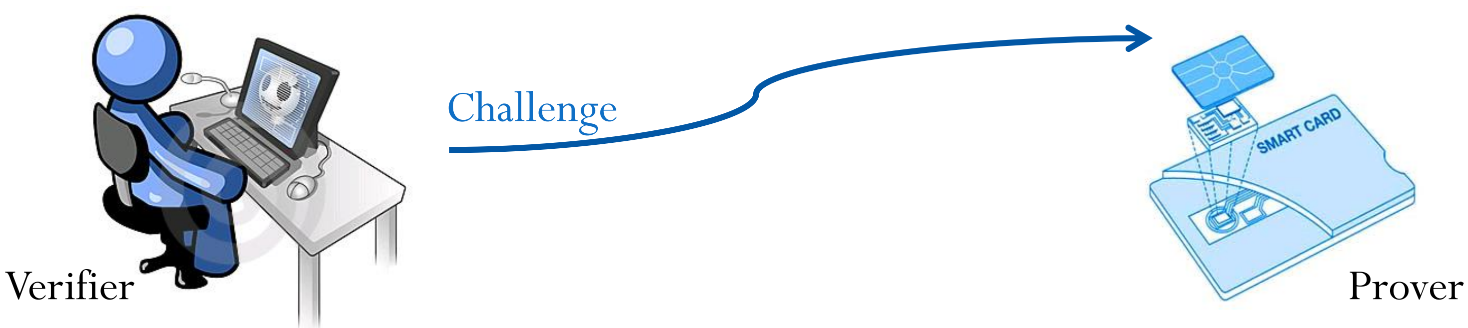
- IC anti-counterfeiting
- Device identification and authentication
- Binding hardware to software platforms
- Secure storage of cryptographic secrets
- Key-less secure communication

Slender PUF Protocol

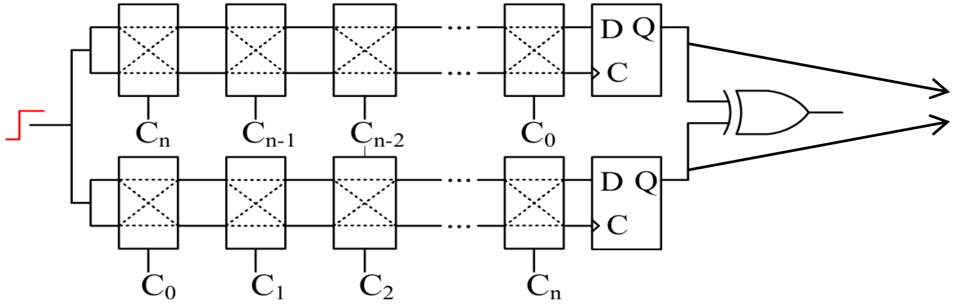
Majzoobi, M., Rostami, M., Koushanfar, F., Wallach, D.S.,
Devadas, S, 2012

Communicating parties

- Prover
  - Has PUF
  - Will be authenticated
- Verifier
  - Has a compact soft model of the PUF
    - Compute challenge/response pairs
  - Will authenticate the prover



XOR-ed delay-based PUF model



Probing here for model building

M. Majzoobi, F. Koushanfar, and M. Potkonjak, 2008

Malicious parties

- Dishonest prover
  - Does not have access to the PUF
  - Wants to pass the authentication
- Eavesdropper
  - Taps the communication between prover and verifier
  - Tries to learn the secret
- Dishonest verifier
  - Does not have access to the PUF soft model
  - Tries to actively trick the prover to leak information

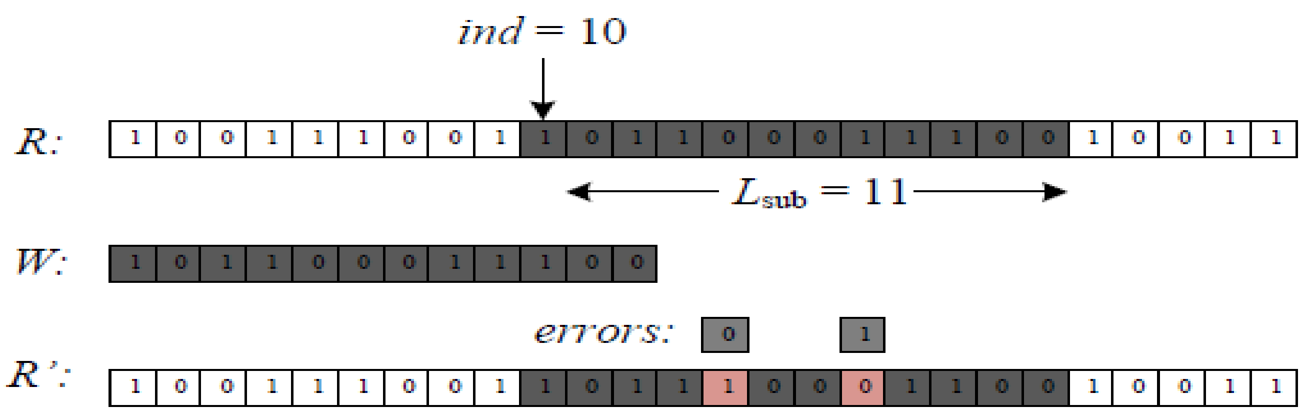
Slender PUF Protocol

Verifier		**Prover**
$Nonce_v$	$\longrightarrow$	
	$\longleftarrow$	$Nonce_p$
Seed = {$Nonce_v$, $Nonce_p$}		Seed = {$Nonce_v$, $Nonce_p$}
C = G(Seed)		C = G(Seed)
$\boldsymbol{R}$' = PUF_model(C)		$\boldsymbol{R}$ = PUF(C)
	$\longleftarrow$	W = sub-seq (ind, L_{sub}, R)
T =match(**R'**, W, e)		
Auth. pass: T = true?		

Verifier

Prover

Slender PUF Protocol



PUF modeling error

It reveals minimum information about original response sequence

Slender PUF Attack Analysis

List of Design Parameters

Parameter notation	Description
L	Length of PUF response string
L_{sub}	Length of PUF response substring
L_{n}	Length of the nonce
ind	Index value, $0 \leq ind < L$
N_{min}	Minimum number CRPs needed to train the PUF model with a misclassification rate of less than ϵ
k	Number of XORed PUF outputs
N	Number of PUF switch stages
th	Matching distance threshold
ϵ	PUF modeling misclassification rate
p_{err}	Probability of error in PUF responses

PUF Modeling Attack

- Minimum number (N_{min}) of direct CRPs required to model a linear PUF with a given level of accuracy.
- Attacker needs to correctly guess <ind> to discover L_{sub} (0 to L-1).
- If $L_{sub} > N_{min}$, then attacker can break the system with O(L) number of attempts.
- If $L_{sub} < N_{min}$, then N_{min}/L_{sub} multiple rounds of authentication needs to be launched to obtain at least N_{min} challenge response pairs. The number of rounds will of the following order:

$$O(L^{\frac{N_{min}}{L_{sub}}})$$

PUF Modeling Attack

- Set $L_{\text{sub}} = 500$, $L = 1024$
- 500000/500=**1000** rounds of protocol needed
- In each one, *ind* is unknown
- $1024^{500000/500} = \mathbf{1024^{1000}}$ models needed to be built

$$2^{10000}$$

- Strict avalanche criteria in the design of PRNG to avoid correlation attacks (using XORed delay based PUF).

Random Guessing Attack

- Dishonest Prover

$$P_{\text{auth,guessing}} \leq L \times \sum_{i=L_{\text{sub}}-th}^{L_{\text{sub}}} \binom{L_{\text{sub}}}{i} \frac{1}{2}^{i} \cdot \frac{1}{2}^{L_{\text{sub}}-i}$$

- Honest Prover

$$P_{\text{auth,honest}} \simeq \sum_{i=L_{\text{sub}}-th}^{L_{\text{sub}}} \binom{L_{\text{sub}}}{i} (1-p_{\text{err}})^{i} \cdot p_{\text{err}}{}^{L_{\text{sub}}-i}$$

Compromising Random Seed

- seed = $\{Nonce_v \ \ Nonce_p\}$
- A dishonest verifier can manipulate an honest prover and the same seed is used over and over during authentication rounds, then the generated response sequence (superstring) will always be the same.
- A dishonest prover (verifier) may keep his/her portion of the seed constant to reduce the entropy of seed.

Replaying Attack

- A dishonest prover may mount an attack by recording the substrings associated with each used Seed by eavesdropping on the communication channel between the legitimate prover and verifier.
- He can repeatedly contact the legitimate verifier for authentication and then matching the generated Seeds to its pre-recorded database.
- The chance that the whole seed collides is: $1/(2^{L_n})$

Exploiting non-idealities of PRNG and PUF

- An attacker may resort to exploiting the statistical bias in a non-ideal PRNG or PUF.
- Can predict pattern in generated responses.
- Leak information about location index of the response substring.
- Must follow the avalanche criteria.

Converse PUF- Based Authentication Protocol

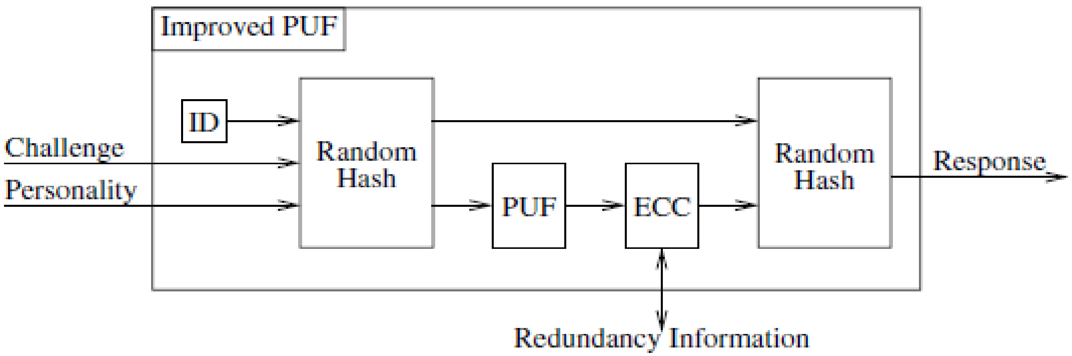
Unal Kocabas, Andreas Peter, Stefan Katzenbeisser, Ahmad-Reza Sadeghi, 2012

Communicating parties

- Prover
  - Has CRP Database
  - Will be authenticated
- Verifier
  - Has a PUF
  - Will authenticate the prover

Controlled PUF

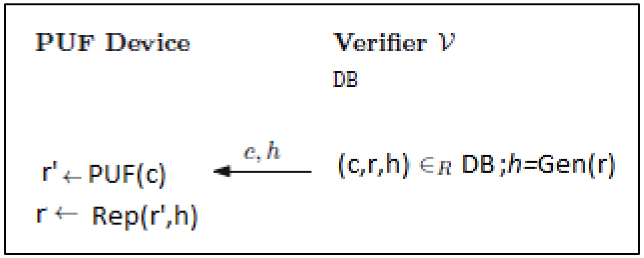
- A CPUF is a combination of a PUF and a control layer in which the PUF is inseparably embedded. The control layer completely shields of the PUF inputs and outputs from the outside world. Any communication with the PUF has to occur through the control layer electronics. Any attempt to force the components apart will damage the PUF.



Gassend, B., Clarke, D.E., van Dijk, M., Devadas, S,2008

Fuzzy Extractor

- Fuzzy extractors consist of a secure sketch, which maps similar PUF responses to the same value, and a randomness extractor, which extracts full-entropy bit-strings from a partially random source. It works in two phases:
  - in the generation phase some helper data h = Gen(r) is computed from PUF response r.
  - in the reproduction phase to recover r = Rep(r', h) from a distorted PUF response r' = r + e, where e is the error caused by noise.
- An important property: after observing one single h, there is still some min-entropy left in r, which means that h can be stored and transferred publicly without disclosing the full PUF response.



Dodis,Y., Reyzin, L., Smith, A,2004

Enrolment Phase

Creating P's database D

For a database $\mathcal{D}$ of size ρ, repeat the protocol ρ times

Prover $\mathcal{P}$ (Creates one element in $\mathcal{D}$)		**Verifier $\mathcal{V}$** (Hosts the PUF)
Choose random PUF challenge C		
	$\xrightarrow{\text{PUF challenge } C}$	
		$(\gamma, h) \longleftarrow \mathsf{FE.Gen}(\mathbf{PUF}(C))$
	$\xleftarrow{R(C,h) := H(\gamma), h}$	
Append $(C, h, R(C,h))$ to $\mathcal{D}$		

Authentication Phase

Prover $\mathcal{P}$ (Holds database $\mathcal{D}$ of size ρ)		**Verifier $\mathcal{V}$** (Hosts the PUF)
		Choose random value $0^n \neq \Delta \xleftarrow{U} \{0,1\}^n$
	$\xleftarrow{\quad \Delta \quad}$	
Find two entries in $\mathcal{D}$: $(C_1, h_1, R(C_1, h_1)), (C_2, h_2, R(C_2, h_2))$ with $\Delta = R(C_1, h_1) \oplus R(C_2, h_2)$ If none found, choose random $C_1 = C_2$ and $h_1 = h_2$		
	$\xrightarrow{(C_1, h_1), (C_2, h_2)}$	
		Compute $R(C_1, h_1) = H(\mathsf{FE.Rep}(\mathbf{PUF}(C_1), h_1))$ and $R(C_2, h_2) = H(\mathsf{FE.Rep}(\mathbf{PUF}(C_2), h_2))$ If $R(C_1, h_1) \oplus R(C_2, h_2) = \Delta$ set $M = \top$, else set $M = \bot$
	$\xleftarrow{\text{Message } M}$	
$\left[\begin{array}{l}\text{Compute shared key} \\ K = \mathcal{H}(R(C_1, h_1) \Vert R(C_2, h_2))\end{array}\right]$		
		$\left[\begin{array}{l}\text{Compute shared key} \\ K = \mathcal{H}(R(C_1, h_1) \Vert R(C_2, h_2))\end{array}\right]$

Probability of Successful Authentication

- For a set M, let $\binom{M}{2}$ denote the set of all subsets of cardinality 2 of M, whereas the elements of this set is denoted by unordered pairs (R1,R2) (excluding duplicate values).
- Consider the set $\binom{\{0,1\}^n}{2}$ has $\binom{2^n}{2}$ many elements.
- For authentication, set $\binom{\mathcal{D}}{2}$ has exactly $\binom{\rho}{2}$ elements taken uniformly at random from $\binom{\{0,1\}^n}{2}$.
- Now we consider

  $$\mathcal{D}^{\oplus} = \{R_1 \oplus R_2 \mid (R_1, R_2) \in \binom{\mathcal{D}}{2}\}$$
- The probability that we hit on Δ when XOR-ing R1 and R2 is :

$$q := \frac{2^n}{\binom{2^n}{2}} = \frac{2}{2^n - 1}$$

Probability of Successful Authentication

- Therefore, the probability of successful authentication is:

$$\mathsf{Succ}^{\mathrm{Auth}}_{\mathcal{P},n}(\rho) = \Pr\left[\Delta \in \tilde{\mathcal{D}}^{\oplus}\right]$$

- The probability of having exactly s successes is given by the binomial probability formula:

$$\Pr\left[s \text{ successes in } \binom{\rho}{2} \text{ trials}\right] = \binom{\binom{\rho}{2}}{s} q^s (1-q)^{\binom{\rho}{2}-s}.$$

- Therefore, the probability of having s = 0 successes is

$$(1-q)^{\frac{\rho^2-\rho}{2}}.$$

$$\mathsf{Succ}^{\mathrm{Auth}}_{\mathcal{P},n}(\rho) = 1 - \Pr\left[0 \text{ successes in } \binom{\rho}{2} \text{ trials}\right] = 1 - \left(1 - \frac{2}{2^n - 1}\right)^{\frac{\rho^2-\rho}{2}}.$$

Security Analysis

- It considers a passive adversary A to see a bounded number of protocol transcripts.
- κ = the (bit-) entropy of the output of the FE in the authentication protocol. The protocol is called (t, κ, ε)-secure (against passive adversaries), if for any probabilistic polynomial time (PPT) adversary A who gets to see t transcripts

  $\tau_i = (\Delta_i, (C_i, C_i'), (h_i, h_i'))$, where $\Delta_i = R(C_i, h_i) \oplus R(C_i', h_i')$, for $i = 1, \ldots, t$, successfully authenticates herself with probability at most ε, i.e.,

  $\Pr[A(\tau_1, \ldots, \tau_t) = ((C, C'), (h, h')) \mid \Delta = R(C, h) \oplus R(C', h')] \leq \varepsilon$

  where the probability is taken over the random coin tosses of A and rand $\Delta \xleftarrow{U} \{0,1\}^n$. We denote this success probability of A by $Succ_{A,n,\kappa}(t)$.

Success Probability of a Worst case Adversary

- After knowing t transcripts, A's database is a list of 2t PUF-challenges C1, . . ., C2t where A knows for at least t pairs the value $R(Ci) \oplus R(Cj) = \Delta i,j$.
- If C1 is fixed, the adversary A gets the following system of t equations: $R(C1) \oplus R(Cj) = \Delta 1,j$ for all $j = 2, \ldots, t + 1$.
- Adding any two of these yields a new equation of the form $R(Ci) \oplus R(Cj) = \Delta i,j$ for $2 \leq i < j \leq t+1$. This means that the adversary can construct up to $\binom{t}{2} - t$ additional Δ-values, called A-checkable.
- The worst case occurs where there are exactly $\binom{t}{2}$ *A-checkable* *Δ-values.*

Success Probability of a Worst case Adversary

- there are only 2^n different Δ-values in total. The adversary can successfully authenticate if:

$$\binom{t}{2} = \frac{t^2-t}{2} = 2^n$$

- t is a positive root of degree two polynomial $X^2 - X - 2^{n+1}$
- The condition will be satisfied if: $t = \frac{1}{2} + \frac{1}{2}\sqrt{1+2^{n+3}}$
- If $\binom{t}{2} \leq 2^n$ $\Pr_{\Delta \xleftarrow{U} \{0,1\}^n}[\Delta \text{ is } \mathcal{A}\text{-checkable}] = \frac{\binom{t}{2}}{2^n} = \frac{t^2-t}{2^{n+1}}$

- Therefore,

$$\Pr_{\Delta \xleftarrow{U} \{0,1\}^n}[\Delta \text{ is not } \mathcal{A}\text{-checkable}] = 1 - \frac{t^2-t}{2^{n+1}} = \frac{2^{n+1}-t^2+t}{2^{n+1}}$$

Success Probability of a Worst case Adversary

- the probability of guessing correctly (meaning that R(C1) $\bigoplus$ R(C2) = Δ)is upper bounded by the probability of guessing two outputs γ1, γ2 of the FE such that H(γ1) $\bigoplus$ H(γ2) = Δ, which is 1/(2^κ). So if Δ is not A-checkable, the success probability of A is less or equal to

$$\frac{2^{n+1}-t^2+t}{2^{n+1}} \cdot \frac{1}{2^\kappa}$$

- The worst case success probability is:

$$\mathsf{Succ}^{\mathrm{wc}}_{\mathcal{A},n,\kappa}(t) = \begin{cases} 1 & , \textit{ if } t > \left\lfloor \frac{1}{2} + \frac{1}{2}\sqrt{1+2^{n+3}} \right\rfloor \\ \frac{(2^\kappa-1)t^2-(2^\kappa-1)t+2^{n+1}}{2^{n+\kappa+1}} & , \textit{ else} \end{cases}$$

Conclusion

- In 2014, Ingrid et. al. had shown that none of the proposed PUF protocols are free from the all kinds of physical attacks.
- Our motivation is to come up with novel PUF architecture that will be also immune to modelling attacks as well as other physical attacks.
- Secondly, we will also try design security protocols assuming that the PUFs are not immune to Modelling Attacks.

Thank You