Participants:

- **Soroush Mostofi Rad – 107361**
- **Mohammadreza Mohebbi Najmabad – 106732**
- **Pranav Deo – 104145**
- **Saeed Doozandeh – 107292**
- **Aurika Nurt – 106666**

Task 1.

Memory Based PUFs.

In general Memory based PUFs are better as they do not require any additional hardware, memory is present in virtually every system out there, and memory has become super cheap too in terms of production.
i.e., we have cost efficiency, plus we can make generate keys for less computation power (Lightweight), we have added benefit of generating keys even after production of the RAM chips are done.

1. **Describe the way in which SRAM PUFs work. What is their disadvantage and how could their reliability be improved?**

SRAM (Static Random-Access Memory) based: Use the design and working principle of the SRAM.

- Use of flip flop based latching circuits.
- The way that SRAMs are designed is that they have usually up to 6 MOSFETs (transistors).
- Each bit in an SRAM cell is stored on four transistors that are connected in two cross coupled inverters. The two additional transistors are access transistors which are used to provide access control during read and write operation.
- The SRAM cells as a result of above design have their preferred state [0 or 1] that they revert to as soon as they are powered up. This is due to differences between the inverters, the so-called larger inverter is what decides the preferred state.
- Due to manufacturing variations, each cell is different and thus has a different preferred state.
- As such, when we power on the device, we have a unique series/ pattern of 0s and 1s, and this pattern can be used a fingerprint for the device.
- Different SRAM chips will have different patterns/ fingerprints, again due to the manufacturing variations.

2. **Explain by means of an illustrative example how DRAM decaying PUFs work. What is their advantage/disadvantage compared to why one might prefer/avoid them over SRAM PUFs?**

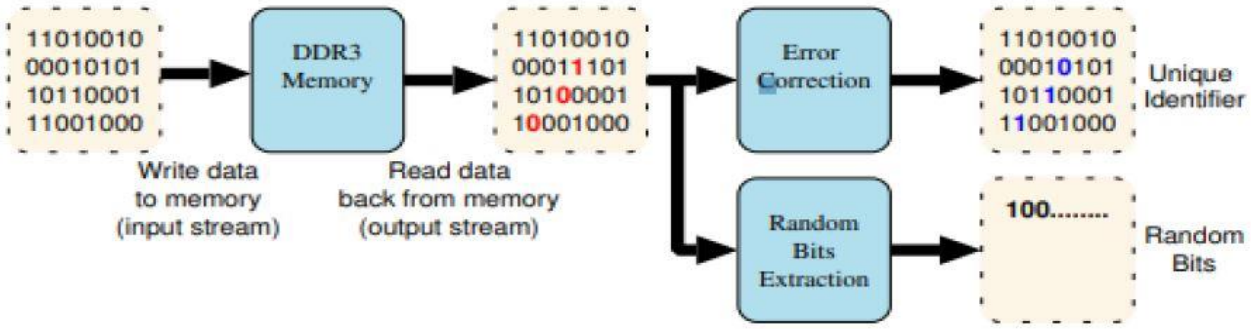
DRAMs cells consist of one transistor and once capacitor as opposed to four or six transistors in a SRAM cell.

Capacitors have a tendency to discharge over a period of time (dielectric leakage).

As such, in DRAM cells, written data is read over a period of time and then the stored data is re-written to avoid any loss of data.

This cycle of re-writing can be done as often as every 32 to 64ms depending on capacitor specifications and operating temperatures.

In reference to 1(#1), they have explored the use of DRAM based PUFs for TRNG and Unique Identifier formation.



**Pic source: paper mentioned in reference 1(#1).

- As said above, there is a loss of charge that happens in a node/ cell over time period.
- The loss can be due to leakage through non conducting transistors, gate leakage of access transistor, loss of capacitor charge etc.
- The leakage though is not always constant, and depends highly on the component characteristics. And as established, in silicon manufacturing processes, there is always production difference that led to slight variations in component characteristics.
- The authors fed DRAMs with some initial input data that was read back after some time from the memory array. The cells which tend to discharge have tendency to flip the input bit and then give an output.
- The data was compared with input data, and several such iterations were done.
- Here upon, authors determined two different types of cells
    1. Perfectible cells: Cells that have a predictive behavior to either not flip any
- bit or a perfectible behavior to discharge and flip a bit.
    2. Unpredictable Cells: Cells whose behaviors cannot be predictable.
- A unique fingerprint can then be generated by isolating the cells that have unpredictable behavior and using them on an error correction code.

Comparison:

- DRAM pufs are cheaper to produce than SRAM pufs.
- DRAM pufs are more robust than SRAM pufs and tend to be more reliable in outputting constant stream.
- In general DRAM pufs are more stable over a long period of time as compared to SRAM pufs.

References:

[1]
C. Keller, F. Gürkaynak, H. Kaeslin and N. Felber
"Dynamic memory-based physically unclonable function for the generation of unique identifiers and true random numbers"
2014 IEEE International Symposium on Circuits and Systems (ISCAS), 2014, pp. 2740-2743, doi: 10.1109/ISCAS.2014.6865740.
link: https://ieeexplore.ieee.org/abstract/document/6865740?casa_token=vnbCZbgLCCcAAAAA:OiKImv5xRISjWV9wEqEqCYvVkiuMQh39ra8_0BOkFcqu8oLDp3um2qp58uJiZnLSrzyJoBKO8g
[2]
https://link.springer.com/chapter/10.1007/978-3-662-53140-2_21

3. **Consider briefly how memory-based combination PUFs could be used to enable device authentication in embedded systems? A general description of the architecture or approach is sufficient (you do not have to go into the smallest detail).**
   - The IoT infrastructure is heavily reliant on the deployment of embedded systems. The concept of Root of Trust is important in IoT to authentication, storing of sensitive data, and facilitating secure communications.
   - A PUF could be used to generate a root key which protects all other stored keys which are used for authentication purposes.

   Considering a challenge response-based mechanism for device authentication:
   - verifying entity is placed that authenticates devices that have RAM based PUFs.
   - The verifying entity initially sends a set of challenges to all the devices and stores the responses in its database.
   - Upon actual implementation, the entity sends a random challenge to the device, the device then computes a PUF response based on its ram based PUF, and computes the challenge.
   - The hash of the challenge is then sent to the entity, which can crosscheck it against the database, and authenticate device if hash matches (a certain amount of threshold tolerance can also be allowed by the entity to account for puf response noise).

Task 2.
PUF-metrics.

1. **List which PUF-related metrics do you know? Also, for each metric, answer what properties of PUF responses do they measure and what are their optimum values?**

The metrics that I know (stated in the course lecture):

a) Hamming weight of the response
   1. Compute Hamming weights of different responses of the same PUF
   2. Ideally: relative value of 0.5 demonstrates no bias in response

b) Intra Hamming distance of responses:
   1. Compute Hamming distances between repeatedly measured PUF response
   2. Relative value close to 0 indicates stability (robustness)

c) Inter Hamming distance of responses
   1. Compute Hamming distances between responses of different PUFs to the same challenge
   2. Relative value close to 0.5 indicates unique responses per device (unclonability)

d) Shannon (binary) entropy (found on internet – no idea)

e) min-entropy (found on internet - no idea)

2. **On StudIP you can find four csv-files containing measurements of a DRAM decay based PUFs. The data set consists of two measurements per chip (Memory1 x.csv for memory one and Memory2 x.csv for memory two). Each csv-file consists of two columns, the first column is the address of the cell and the second column the corresponding value. All of the four measurements were gathered with a DRAM decay time of 240 seconds and an ambient temperature of 50 degree. Write a program which parses the measure data and calculates the metrics described in the previous task. Finally evaluate the calculated metrics. What do you observe?**

Algorithm 1. `Enroll`

1 For $i \in \{1, \ldots, N\}$ do the following experiment:
  Charge the DRAM. Let it decay for time t_1. Let $\mathcal{F}_i$ be the set of addresses of the decayed cells;
2 $\mathcal{F} = \mathcal{F}_1 \cap \cdots \cap \mathcal{F}_N$;
3 Charge the DRAM. Let it decay for time t_3;
4 Let $\mathcal{S}$ be the set of addresses of the cells that have not yet decayed;
5 Randomly pick n_f elements from $\mathcal{F}$ and n_s elements from $\mathcal{S}$, with $n_f + n_s = n$. Construct a vector r by putting the n elements in a random order.
  Construct $x \in \{0,1\}^n$ such that $x_i = 1$ if $r_i \in \mathcal{F}$ and $x_i = 0$ if $r_i \in \mathcal{S}$;
6 $w = \mathtt{Syn}(x)$;
7 Generate random p. Compute $\mathcal{K} = \mathtt{KeyDeriv}(x, p)$;
8 Store (r, w, p) in memory.

Algorithm 2. `Rec`

1 Read (r', w', p');
2 Charge the DRAM. Let it decay for time t_2;
3 Construct $x' \in \{0,1\}^n$ such that $x'_i = 1$ if the cell at address r'_i is decayed and 0 otherwise;
4 $\hat{x} = x' \oplus \mathtt{SynDec}(w' \oplus \mathtt{Syn}(x'))$;
5 $\hat{\mathcal{K}} = \mathtt{KeyDeriv}(\hat{x}, p')$.

Algorithm 3. `Mutual Authentication` Let $\mathcal{N}_{id}$ Denote the Set of all Memory Cells in PUF_{id}.

1 $\mathcal{P}$ initiates contact;
2 $\mathcal{V}$ sends id to $\mathcal{P}$;
3 $\mathcal{P}$ performs the following actions:
  Set $x = c_{id}$. Perform a measurement of PUF_{id} at decay time t_x. The result is a set of addresses $s^{id}(t_x)$ of decayed cells. Randomly select addresses into a set $\mathcal{B} \subset \mathcal{N}_{id} \setminus s^{id}(t_x)$ of size $|\mathcal{B}| = 2\epsilon_t(x+1) - l_x^{id}$ and construct a vector z by randomly permuting $s^{id}(t_x) \cup \mathcal{B}$. Construct a bit string $a \in \{0,1\}^{2\epsilon_t(x+1)}$ such that $a_i = 1$ if $z_i \in s^{id}(t_x)$ and $a_i = 0$ otherwise. Increase c_{id} and send x, z to $\mathcal{V}$;
4 $\mathcal{V}$ performs the following actions:
  Continue only if $x \geq c'_{id}$ and z has length $2\epsilon_t(x+1)$; else abort. Construct $a' \in \{0,1\}^{2\epsilon_t(x+1)}$ such that $a'_i = 1$ if $z_i \in s_e^{id}(t_x)$. If the fractional Hamming weight of a' is larger than $\frac{1}{2}(1 - \Delta_1)$, then set $c'_{id} = x + 1$ and send a', else abort;
5 $\mathcal{P}$ checks if the fractional Hamming distance between a and a' is smaller than Δ_2. If not, $\mathcal{P}$ aborts.

References:

https://pure.tue.nl/ws/files/133415079/08332526.pdf

Task 3.

Error Correction.

1. **A PUF is measured twice and returns the following sequences:**
   Apply the indices to stable bits method over these two measurements.

In the indices to stable bits method, we basically identify what the stable bits are and this use only the stable bits.
In following example, the samples measurement count of two is technically insufficient to evaluate if the stable bits are indeed 100 percent stable.

Measurement 1:
10100100011000011111001101010000011001111100010101110010011001 10
Measurement 2:
01101110011010010001001111010010011001111100010101110010011001 10
Mask: ** * ***** *** ***** *****
****
Response: 10 1 00110 001 10011 10100
00110011111000101011100100110011 0

2. **Describe the Code Offset Method. What is its advantage and disadvantage?**

In code offset method,

- The initial puf measurement is taken, say X .
- A random word W is chosen carefully depending on what kind errors are to be expected, the frequency of errors and the failure probability.
- A code word is calculated as code word c = X xor W and we store W & hash(X xor W)
- A new measurement of PUF is taken, say Y .
- We can now find a new code while correcting Y, new code c_new = Y xor W , and compute the hash(c_new) .

Advantage:

- Usually corrects all the errors in the code.

Disadvantage:

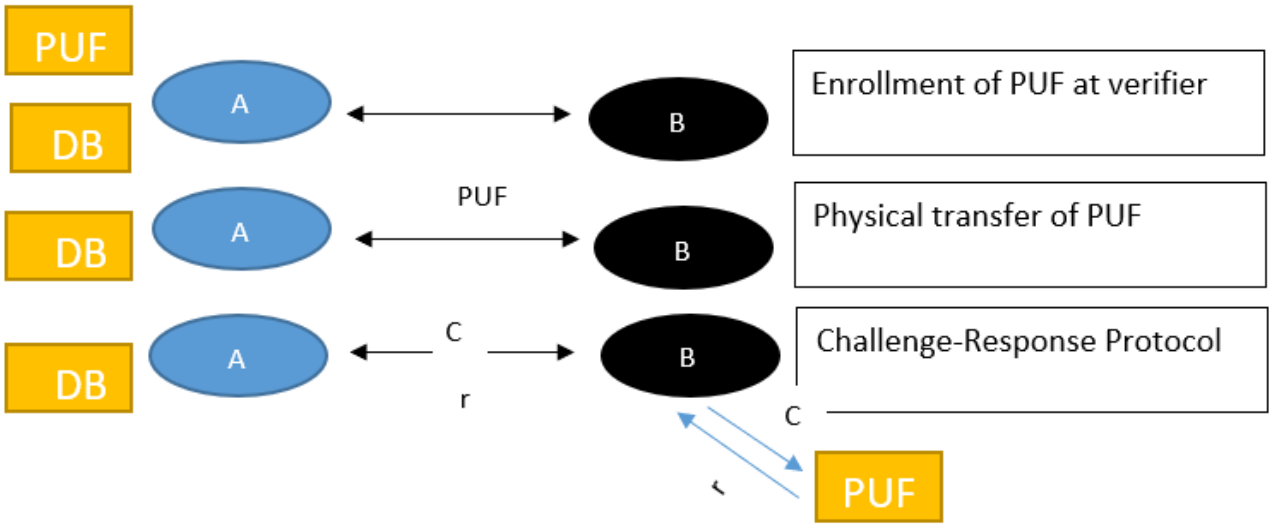
- Implementation is slow compared to other methods.
- Right code word has to be chosen carefully.

Task 4.

PUF protocols and practical applications.

1. **Describe a basic authentication protocol based on a Physical Unclonable Function with an explanatory drawing. In this protocol, a server should authenticate the client. In addition, there should be a registration phase where the server collects information about the PUF and then sends the PUF to the client.**

A is the client and B is the server:



According to the picture above

a) A sends some challenges to PUF and receives responses. It stores all the responses in a database
b) A sends the physical PUF to B
c) A sends a challenge to B and B send the challenge to PUF. Then B gets a response and sends it to A. A check whether the response is the same as the one stored in database and then if yes, understands that B is the right person and is authenticated.

- Assumption is that response is stable

2. Think about two application examples in the area of the Internet of Things that could be based on the key agreement protocol you have just defined. Also describe how the protocol is used in detail in each example.

Key agreement between two constrained IoT devices that have never met each other is an essential feature to provide in order to establish trust among its users. Physical Unclonable Functions (PUFs) on a device represent a low-cost primitive exploiting the unique random patterns in the device allowing it to generate a unique response for a given challenge. These so-called challenge-response pairs (CRPs) are first shared with the verifier and later used in the authentication process. The advantage of a PUF at the IoT is that even when the key material is extracted, an attacker cannot take over the identity of the tampered device. However, in practical applications, the verifier, orchestrating the authentication among the two IoT nodes, represents a cluster node in the field, who might be vulnerable for corruption or attacks, leading to the leakage of the CRPs. Possessing a huge number of CRPs allows its usage in machine learning algorithms reveal the behavior of the PUF. We propose a very efficient method to provide authentication between two IoT devices using PUFs and a common trusted cluster node, where the CRPs are not stored in an explicit way. Even when the attacker is able to get access to the database, the stored information related to the CRPs will not be usable input for any type of learning algorithm. The proposed scheme uses only elliptic curve multiplications and additions, instead of the compute intensive pairing operations as an alternative scheme recently proposed in the literature.

Two applications to name:

a. Security
b. Maching learning

References:

https://www.researchgate.net/publication/337932420_PUF-Based_Authentication_and_Key_Exchange_for_Internet_of_Things