University of London
Imperial College of Science, Technology and Medicine
Department of Computing

Building Strong Authentication and Key Storage PUF Scheme using off-the-shelf SRAM

Ade Setyawan

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy in Computing of the University of London and
the Diploma of Imperial College, February 20, 2018

Chapter 1

Introduction

1.1 Trust and Security

If we look further into modern economy, one of the activity that we commonly do is saving our money on bank and collect our money from ATM machine. We rely on this so much that we sometimes forget why do we trust our money in another entity. One of the reason is because we believe that bank is capable of keeping our money save, regardless what might happen in the future. When we trust the bank, we also expect the bank to trust us, especially when we want to collect our money. One way to tell the bank that we are the entity who own that money, we can use our debit card. Sometimes we depend so much that when we lost it, our world die a little.

Since debit card is important, debit card providers, such as Mastercard and VISA, also provide some features to ensure that their product is safe. For example, in the first generation, they introduce the usage of magnetic stripe. In the latest generation they use EMV, also known as chip-and-pin technology [1]. This technology is safer than magnetic stripe because it support dynamic authentication, unlike the former one that only use a static data which make an attacker easily clone the data using inexpensive card reading device.

One important aspect of debit card which keep us on using it is its trustworthy. Why do we trust debit card? We trust it because we believe that it is really hard or even impossible to clone it without bank approval. Since we trust bank, we also trust its product. We believe that bank only associate one card with one entity, either a company or a user. In a more simple words, we are confident that debit card and the system around it is secure.

Since security is important, on designing a system that deal with sensitive informations, one need to ensure that the system is able to protect the information and property from unauthorized entity without sacrificing the system's functionality. The study dedicated to protect information from unauthorized access, loss or damage is called information security [2].

In information security, one way to protect an information is by using cryptographic techniques. A cryptographic technique commonly consist of a cryptographic algorithm and a key. On designing a cryptographic algorithm, one should follow Shannon's Maxim which says one should design systems under the assumption that the enemy will immediately gain complete knowledge with the design [3]. In another word, on a secure cryptographic technique, the only thing which need to be kept secret is the key. The algorithm itself is supposed to be publicly available without affecting the system security. The most common way to stored the key is by using a *non volatile memory* (NVM). NVM is a type of computer memory that keep intact its information even after turned off. An example of product which implement this approach is the debit card that is mentioned above. Debit card use its chip to store information.

Unfortunately, this NVM is prone to physical attack. Since the key is permanently stored in the memory, an attacker can use some technique to clone the memory, such as microprobing [4]. Attacker may also use a side channel information to retrieve any information about the key. There are numerous other techniques on this kind of attack. This attack can be even worse if someone that knows the system design is involved. Due to this problem, more secure, tamper-evident, tamper-proof solutions need to be presented.

1.2 Rise of PUF as a Security Solution

In 2001, Physical Unclonable Function (PUF) comes in handy as an inexpensive and yet effective security solution to overcome the mentioned problem above by a different way of generating and processing secret keys in security hardware. It was introduced by Pappu [6]. Unlike cryptographic algorithm security which usually rely on hard-to-solve mathematical problem, PUF ideas stems from using hardware features designed to utilize the physical random nanoscale disarray phenomena [7]. This disarray phenomena can be used as a derivation of keys without having to keep any security-critical information explicitly. This physical randomness is unclonable, even by the original manufacturer due to manufacturing process variations. Furthermore, since the secrets can only be produced when the PUF device is turned on, active manipulation of circuit structure will cause dysfunction of challenge-response mechanism and destroy the secret.

An example of PUF type is SRAM PUF. SRAM, stands for static random-access memory, is a type of semiconductor memory that uses bistable latching circuitry (flip-flop) to store each bit. When a static RAM (SRAM) is turned on, the memory cells have an undefined state [8]. Since the initialized bits of SRAM is random, these bits are a good candidate for PUF. The value of these bits itself is determined by the SRAM cell which consists of two cross-coupled inverters along with two access transistors. This concept was firstly introduced by Guajardo [9]. In order for SRAM to be used as a cryptographic security key, SRAM PUFs need to have certain characteristics such as the key generated by every SRAM should be reliable and unique. Reliable means the generated key should always be consistent, while unique refers to there should be no correlation between one device or another.

Unfortunately, SRAM PUF is also problematic since it contains noise in its bit value. Many proposed solution has emerged, but it seems that its availability is still limited, expensive or only exclusive for some companies. SRAM PUF also consider as a weak PUF, which means that it has limited challenge-response pairs. Due to limited challenge-response pairs, it is argued as unsuitable for authentication scheme, only appropriate for key generation or random number generator [7].

1.3 Problem Statement

As mentioned above, SRAM PUF still has some problems that need to solve. A future where anyone can have their own SRAM PUF without having to buy from specific company is a wonderful possibility. If everyone can just buy an SRAM from the market and use it on top of an open system, this will revolutionize the security industry. This objective describes the motivation for this thesis.

The goal of this thesis is to build an open system where anyone can use their own SRAM as a PUF solution and provide a secure key storage and authentication function. The system will include a SRAM's stable bits analyzer, test system to check the SRAM's quality as a PUF, and a scheme of key storage and authentication as the main purpose of PUF itself.

As an attempt to reach this goal, several steps are expected to be done:

1. Get multiple type of SRAM available in the market
2. Investigate the characteristic of each SRAM
3. Investigate and choose the embedded platform where the system will be build
4. Design a system which able to determine the stable bits of SRAM automatically
5. Search and analyze existing secure key storage and authentication methods using SRAM PUF
6. Propose a system to enable secure PUF key storage and authentication using off-the-shelf SRAM
7. Construct the complete software, which contained the error correcting code
8. Evaluate the solution by experimenting on SRAM and conducting performance analysis
9. Explore possible improvements on the system for a future research

1.4 Contributions

This thesis provides the following contributions:

1. Analysis of time difference, voltage difference and neighbor values on the stability of SRAMs 23LC1024 and CY62256NLL
2. Result comparisons between two methods on looking for stable bits: data remanence and neighbor stability analysis on SRAMs 23LC1024 and CY62256NLL
3. A system to enable strong PUF authentication and key storage scheme using SRAM

1.5 Outlines

Chapter 2

Background Theory

2.1 PUF

A physical unclonable function is an entity that utilize manufacturing variability to produce a device-specific output. The idea to build PUF arise from 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 [7]. PUFs leverage this variability to derive secret information that is unique to the chip. This secret can be refer as a silicon biometric.

In addition, due to the manufacturing variability that defines the secret, one cannot manufacture two identical chips, even with full knowledge of the chips design. PUF architectures exploit manufacturing variability in multiple ways. For example, one can utilize the effect of gate delay, the power-on state of SRAM, threshold voltages, and many other physical characteristics to derive the secret.

Due to this feature, PUFs are a promising innovative primitive that are used for authentication and secret key storage without the requirement of secure hardware. Currently, the best practice for providing a secure memory or authentication source in such a mobile system is to place a secret key in a nonvolatile electrically erasable programmable read-only memory (EEPROM) or battery- backed static random-access memory (SRAM) and use hardware cryptographic operations such as digital signatures or encryption.

There are two main parts of PUF, physical part and operational part. Physical part refers to a physical system that is very difficult to clone due to uncontrollable process variations during manufacturing. Operational part means a set of challenges (PUF input) C_i has to be available to which the system responds with a set of sufficiently different responses (PUF output) R_i. This combination of challenge and response is called challenge-response-pair (CRP).

$$R_i <- PUF(C_i) \tag{2.1}$$

According to [7], to be qualified as PUF, a device should fulfilled several characteristics below :

- Reliable: A response to the same challenge should be able to be reproduced over time and over various range of conditions.
- Unpredictable: A response to a challenge on a PUF device should be unrelated to a response to another challenge from the same device or the same challenge from different device.
- Unclonable: Challenge-response pairs mapping of a device should be unique and cannot be duplicate.
- Physically Unbreakable: Any physical attempts to maliciously modify the device will result in malfunction or permanent damage.

2.1.1 PUFs Classification

Based on the number CRPs, PUFs can be divided into two categories:

- Strong PUFs

  Strong PUFs can be identified by having large number of CRPs. Strong PUFs typically used for authentication.
- Weak PUFs

  Contrary to strong PUFs, weak PUFs only have a small number of CRPs. Weak PUFs commonly used for key storage.

Beside number of CRPs, PUFs can also categorized based on their physical design. There are two major category, extrinsic and intrinsic.

Extrinsic means that it need extra hardware added to the PUF component. There are two subcategories in extrinsic PUFs, non electronic and analog electronic PUFs. Some example in non electronic PUFs are optical PUF, paper PUF, CD PUF, RF-DNA PUF, magnetic PUF, and acoustic PUF. Some design instances in analog electronic PUFs are VT PUF, power distribution PUF, coating PUF, and LC PUF.

In intrinsic, the PUF component has to be available naturally during the manufacturing process. In addition, PUF and the measurement equipment should be fully integrated in intrinsic PUF. There are two subcategories in intrinsic PUFs, delay based and memory based PUFs. An example of delay based PUF is arbiter PUF. The main principle of arbiter PUF is to introduce a digital race condition on two paths on a chip and have an arbiter circuit to decide which one won the race. As in memory based PUFs, some examples of this design are SRAM PUF, butterfly PUF and latch PUF. SRAM PUF utilized the random physical mismatch in the cell caused by manufacturing variability determines the power up behavior (can be zero, one, or no preference). Butterfly PUF use the effect of cross coupling between two transparent data latches. Using the clear functionalities of the latches, an unstable state can be introduced after which the circuit converges back to one of the two stable states. In latch PUF, the concept is based on using two NOR gates which are cross coupled. These gates will converge to a stable state depending on the internal mismatch between the electronic components.

Both weak and strong PUFs rely on analog physical properties of the fabricated circuit to derive secret information. Naturally, these analog properties have noise and variability associated with them. To handle the noise, PUF designs usually employ multiple error-correction techniques to correct these bits, improving reliability. However, when using the error correction techniques, one should pay attention because some error-correcting techniques probably leak some bits of the secret key, since they require the computation and public storage of syndrome bits.

2.1.2 Intra and Inter Hamming Distances

As explained before, PUF main purpose is dedicated for identification, shown by having a device specific output. To help defining this idea, there are two types of hamming distance commonly use, intra-chip ($\mathrm{HD}_{\mathrm{intra}}$) and inter-chip ($\mathrm{HD}_{\mathrm{inter}}$) hamming distance. Inter-chip hamming distance is the distance between two responses resulting from applying a challenge once to two different PUFs device. Intra-chip hamming distance refers to difference between the two responses resulting from applying a challenge twice to a PUF device [10]. Hamming distance itself is the number of positions at which the corresponding symbols are different on two equal length strings. In ideal PUFs, the $\mathrm{HD}_{\mathrm{intra}}$ is 0% and $\mathrm{HD}_{\mathrm{inter}}$ is 50%. Due to noises, normally PUF devices has $\mathrm{HD}_{\mathrm{intra}} \leq 10\%$ and $\mathrm{HD}_{\mathrm{inter}}$ 50%.

2.2 Fuzzy Extractor

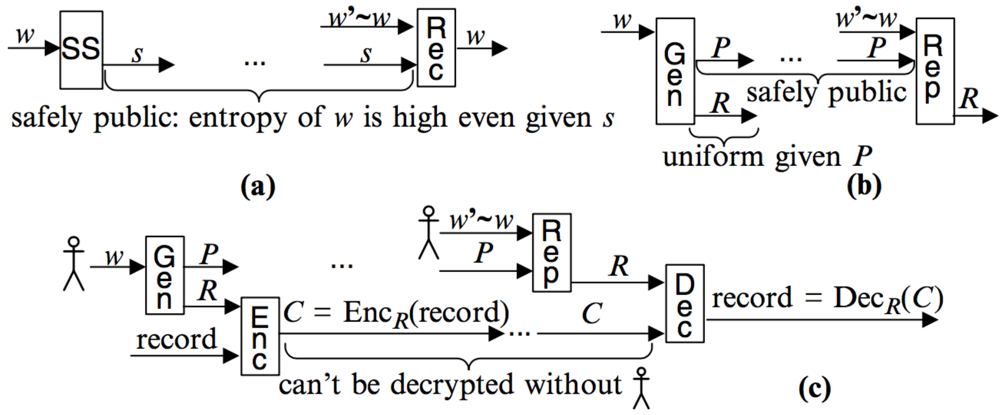

Figure 2.1: Fuzzy Extractor Scheme

2.3 Helper Data Algorithms

2.4 PUF Application

Key Generator

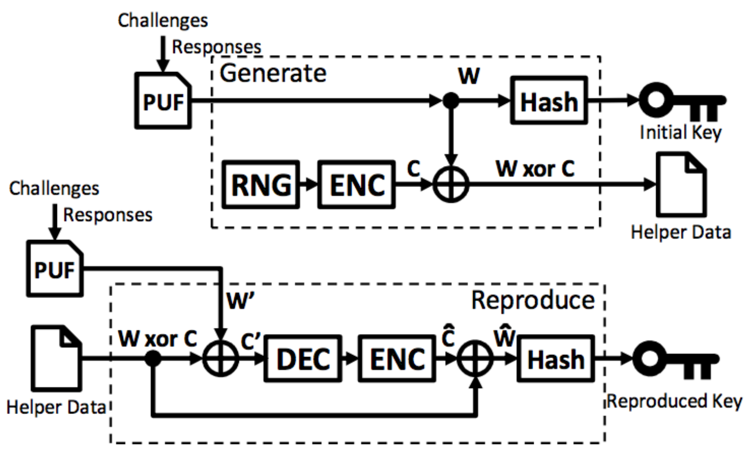

Figure 2.2: Scheme of Stable Key Generation

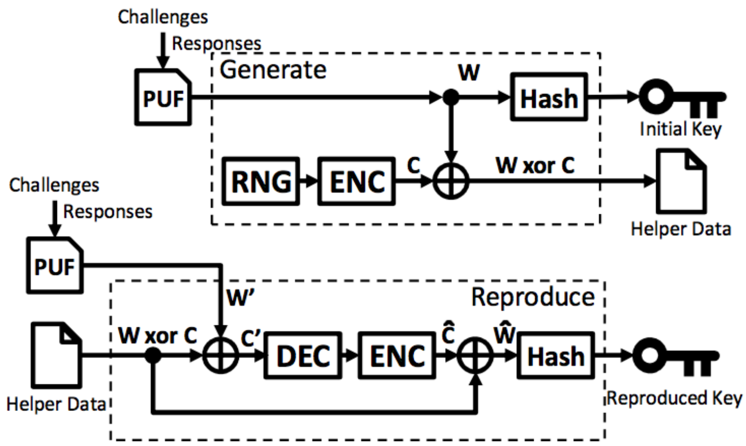

Figure 2.3: Scheme of Stable Key Generation

2.5 Key Storage using PUF Scheme

In a basic key storage scheme using PUF, it usually divided into two phases. In the first phase, generally called enrollment, the helper data is constructed by XOR-ing PUF bits and the encoded key. In the second phase, which called reconstruction, the helper data is XOR-ed with the PUF bits, followed by decoding the result to reconstruct the key.

2.6 Authentication using PUF Scheme

In a basic authentication scheme using PUF, it usually divided into two phases. In the first phase, generally called enrollment, CRPs are generated and stored in a CRP database. In the second phase, usually referred to verification, a challenge from the CRP database is applied to the PUF and the response produced by the PUF is compared with the corresponding response from the database.

Chapter 3

SRAM PUF

SRAM PUFs use existing SRAM blocks to generate chip-specific data. After powering-up the circuit the cells stabilize at a state which is defined by the mismatches between the involved transistors. Thus, each SRAM cell provides one bit of output data. A SRAM cell has two stable states (used to store a 1 or a 0), and positive feedback to force the cell into one of these two states and, once it is there, prevent the cell from transitioning out of this state accidentally.

The SRAM PUF was first proposed by Guajardo and Holcomb in 2007.

A write operation forces the SRAM cell to transition toward one of the two states. However, if the device pow- ers up and no write operation has occurred, the SRAM cell exists in a metastable state where theoretically, the feed- back pushing the cell toward the 1 state equals the feed- back pushing the cell toward the 0 state, thereby keeping the cell in this metastable state indefinitely. In actual implementations, however, one feedback loop is always slightly stronger than the other due to small transistor threshold mismatches resulting from process variation. Natural thermal and shot noise trigger the positive feed- back loop, and the cell relaxes into either the 1 or 0 state depending on this process variation. Note that since the final state depends on the difference between two feedback loops, the measurement is differential. Therefore, common mode noise such as die temperature, power supply fluctuations, and common mode process variations should not strongly impact the transition.

Like other strong and weak PUF implementations, the SRAM PUF is also sensitive to noise. If the two feedback loops of the SRAM cell are sufficiently close, then random noise or other small environmental fluctuations can result in an output bit flip. Therefore, error correction of this output will be necessary.

3.1 Error Correcting Codes

As mentioned in the introduction, error-correcting codes (ECC) is useful in SRAM PUF scheme due to noise available in SRAM initialized values. Error-correcting codes are a class of schemes for encoding messages in an attempt to enable message recovery when there is noise introduced in the sending or receiving of the message. ECC can be divided into two subcategory, hard-decision and soft-decision. Hard-decision operates on fixed set of possible values (usually 0 or 1 in a binary code), while the inputs to a soft-decision decoder may take on a whole range of values in-between (usually refers to float value).

There are some well-known ECC, such as in hard-decision code, Reed-Solomon code and BCH code; while in soft-decision, Viterbi code and turbo code. Soft-decision code has an advantage over hard-decision code where it can process extra information which indicates the reliability of each input data point and used to form better estimates of the original data. But it has drawback where one should provide a probability function on the data (on SRAM, a probability function on each cell should be provided) to enable a good decoding result. This is a problem if applied on this thesis goal where the system should work on any SRAM off-the-market. Calculating the probability on each SRAM cell will take an extra step, over complicate the system and the procedure on using the constructed system. Thus, the hard-decision code is preferred.

One of the popular hard-decision error correcting code is BCH codes. BCH, stands for *BoseChaudhuriHocquenghem*, codes are a family of cyclic error correcting codes which constructed using polynomials over a finite field and work in binary field. BCH codes are a very flexible set of codes in that within certain bounds there is a great amount of choice in code parameters and are relatively efficient in message length and error correction. The code parameters are as follows:

- q: The number of symbols used (e.g., in binary field, $q = 2$)

- m: The power to which to raise q to generate a Galois Field for the construction of the code.

- d: The minimum Hamming distance between distinct codewords.

These parameters lead to several derived parameters which are standard parameters of linear codes:

- n: The block length of the code; for our special case, $n = q * m1$

- t: The number of errors that can be corrected, $d \geq 2t + 1$

- k: The number of message bits in a codeword, $k \geq n - mt$

Both BCH codes and Reed-Solomon codes have the capability to correct multiple errors. Reed-Solomon codes is also a flexible ECC and have similar parameters as BCH codes, e.g. n, k, d. Unlike BCH codes, Reed-Solomon codes can work in both binary and non-binary fields. Reed-Solomon codes also perform better in correcting burst errors while BCH codes are better in fixing random errors. BCH codes has an advantage where it requires less computing resource when working on same parameter compared to Reed-Solomon codes.

3.2 Phenomena Increasing the Noise

[11]

- Voltage Noise: When an input pattern is applied to a circuit, it will create a large number of switchings in the circuit. The switching will increase dynamic power and cause voltage drop on power lines and voltage increase on ground lines. This effect is known as power supply noise. When the voltage reaching a gate changes, it will change the delay characteristics of the gate.

- Temperature: Increase in power also increases the temperature over time depending on the type and number of patterns applied to the circuit. Temperature distribution depends on the location of switchings and distribution of power consumption in the circuit.

- Crosstalk: As technology feature size scales, interconnect spacing and width are also being reduced. However, in order to keep the resistance low, the thickness of the wires is not scaled at the same rate. This produces tall sidewalls between long parallel interconnects separated by very little space, which creates a parasitic coupling capacitance between wires. Due to this fact, crosstalk has become a significant contributor to signal integrity problems in modern designs.
- Aging:

3.3 Bit Selection Algorithm

first important step on using sram puf is looking for stable bits. stable bits are used as a basic identifier in PUF, thus it is important to ensure that the PUF result is always the same throughout its lifetime. since not every SRAM cell is stable, we need to take special caution on deciding which SRAM cell is gonna be our bits to use in PUF. In here, we use two known algorithm to search for stable bits.

3.3.1 Neighbour Analysis

The first algorithm is use the rank of total stable neighbors [12]. The cells that are most stable across environmental conditions are surrounded by more stable cells during enrollment. A stable cell surrounded by more stable cells has a tendency to become more stable because its neighboring cells are likely to experience similar aging stress and operating conditions. The more stable neighbor cells it has, the higher weight it gets. After determining the weight of each cell, a heuristic algorithm that greedily chooses cells for the PUF ID/key with weight greater than a threshold is used.

Before the algorithm is performed, one should collect lots data of SRAM cells value first. The data should be retrieved in various condition, for example different voltages, temperatures, and time differences between enrollment. Afterwards, we use Temporal majority voting (TMV) to calculate all stable bits in SRAM. Last, the neighbor analysis algorithm is performed to get the most stable bits in SRAM.

3.3.2 Data Remanence

Another bits selection algorithm is by using data remanence of SRAM cell [13]. This approach requires only two remanence tests: writing 1 (or 0) to the entire array and momentarily shutting down the power until a few cells flip. The cells that are easily flipped are the most robust cells when written with the opposite data. Strong 1's are bits that are flipped fast after 0 is written to its location. On the contrary, if 1 is written to a bit location and the bit flipped fast, it means that the bit is a strong 0.

Chapter 4

Proposed System

4.1 Automated PUF Profiling System

To increase the experiment's efficiency, an automated PUF profiling system are constructed. The system consists of a PC, act as a master, and an Arduino connected to an external SRAM which act as a client. A custom protocol was designed to communicate between them. It is specifically designed to be generic and usable for all types of PUF profiling measurements. The software on Arduino side waits for measurement commands sent by PC on the serial link after booting. The designed protocol are dedicated for voltage control, read bytes, write bytes, and memory disable/enable.

4.2 BCH Codes as Error Correcting Codes

As mentioned in the previous chapter, BCH codes are a flexible ECC shown by multiple parameter available. The only fixed parameter is q since the problem is in binary form ($q = 2$). These parameter should be determined with several considerations, such as, the inner hamming distance of SRAMs and memory available on Arduino Mega 2560.

Here are the chosen parameters:

- n: 63

- k: 7
- d: 31
- t: 15

These parameters is chosen based on several reasons, which are:

1. Large n requires large memory and computing capability especially during the polynomial multiplication on decoding. Large program will not fit in Arduino Mega 2560. For example, Arduino memory is insufficient for $n = 511$. Even though larger n also means a higher capability on correcting the code, one should choose a parameter which fit in Arduino without sacrificing the program error capability. Thus $n = 63$ is chosen.
2. The value k, d, and t is correlated to each other. k is chosen to be 7 to enable $t = 15$. This combination will be able to correct 23.8% of a block of data (correct 15 bits in 63 bits of data).

Since 63 bits is required to encode 7 bits of data, if according to the key storage scheme

4.3 Key Storage Scheme

The key storage scheme is shown in Figure 4.1. This scheme can be used to store 7 bits of key which require 63 SRAM bits. If the full length of key is 256 bits, there will be 37 blocks similar like this figure and 2331 SRAM bits required.

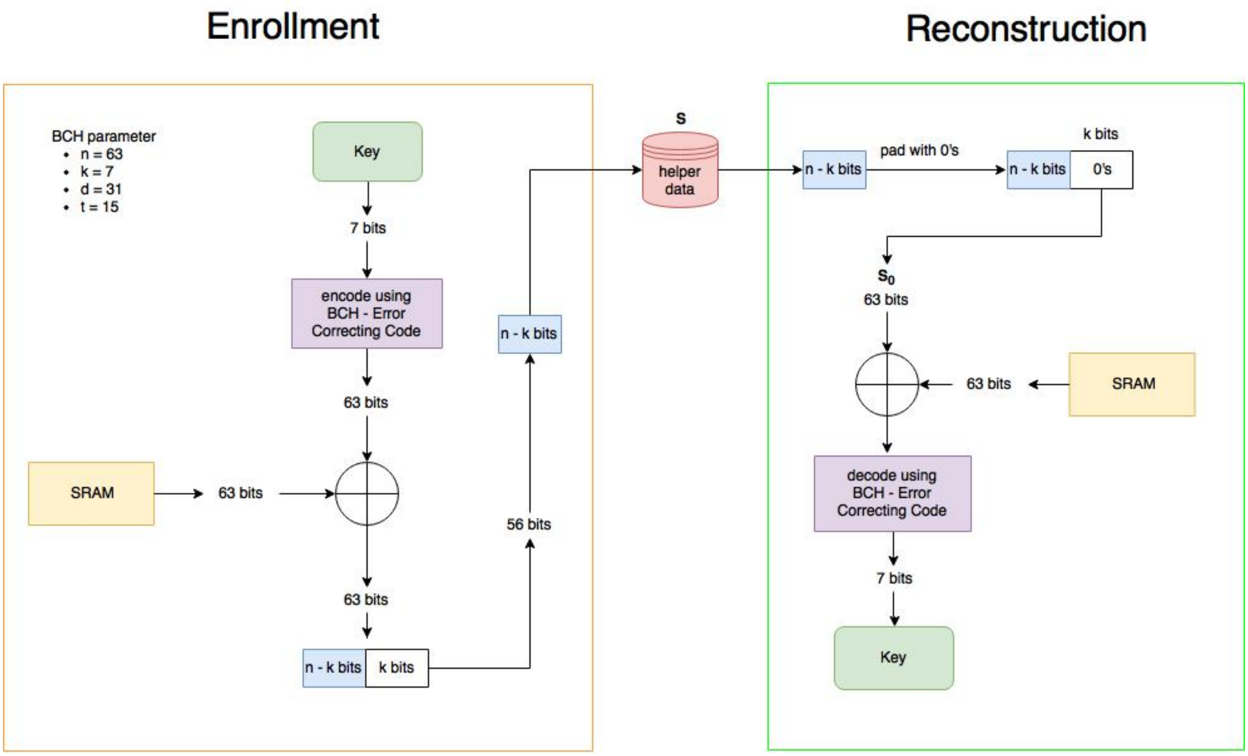

Figure 4.1: Scheme for Key Storage

4.4 Authentication Scheme

we used hmac function. the PUF bits is used as a key for the hmac function.

to create challenge response pairs, the server will give a random bits as input/challenge. the output of the function is the response.

using this scheme, it will be a strong PUF authentication scheme because there will be so many CRPs. The limit is the capability limit of the HMAC itself times the 2^n where n is the bits length of the PUF bits.

4.5 Defense Against Attacks

4.5.1 Man-in-the-middle Attack

Chapter 5

Experiment

This chapter contains explanation on experiment setup and results.

5.1 Chosen SRAM

In an attempt to achieve the thesis goals, the first step is looking for SRAM. There are numerous SRAM types available in the market. To Main requirements on the SRAM are easy to get (a simple google search should show some e-commerce websites to buy from), can be bought in small quantity (≤ 5 pieces), stand-alone component (available without buying extra component, e.g. not embedded in an FPGA), inexpensive (cost less than 5), reasonable memory size ($\geq$ 64kb). These criteria are chosen due to some product only sold to a company or someone that willing to buy in a big quantity or has to be custom made. There are two SRAM types purchased and tested here; Microchip 23LC 1024 and Cypress CY62256NLL. On each SRAM, there are several experiment performed to determine if these SRAMs are a suitable candidate for PUF, such as calculating HD_{intra} and HD_{inter} given the whole memory value as the challenge.

5.1.1 Microchip 23LC1024

The Microchip Technology Inc. 23A1024/23LC1024 is a 1024 Kbit Serial SRAM device. This SRAM is really popular shown by many references available online and several github repository intended just to access this SRAM. The reason of its popularity can be traced to its cheap price, small size and easy to use feature. The price is ranging from 1.5-3.5. This device has eight pins which contribute significantly on its small footprint. It is easy to use because it provides SPI connection which simplified the communication, and has three modes available; SPI (Serial Peripheral Interface), SDI (Serial Dual Interface) and SQI (Serial Quad Interface). Its voltage range also quite large, ranging from 2.5-5.5V. Figure 5.1 shows the Microchip 23LC1024.

Figure 5.1: SRAM 23LC1024

There are ten Microchip 23LC1024 SRAMs that were available during experiment. To check whether this SRAM is a justifiable candidate for PUF, several testing are performed. First, the number of 1's and 0's in memory after a start is calculated. Unfortunately, the average distribution of 1's and 0's are not similar, 1's occupy 70% and 0's fill the remaining 30%.

Second, HD_{intra} and HD_{inter} are calculated on both chips. The calculation are done using twenty data of chip memory values on each chip which retrieved on room temperature, 5V input and 10 seconds interval between each enrollment. From these chips, the average HD_{intra} is 5.75% and the average HD_{inter} is 42.54%.

Third, the effect of voltage variation on the HD_{intra} and HD_{inter} are also evaluated. The calculation are done using chip memory values on each chip which retrieved on room temperature and 10 seconds interval between each enrollment. The voltage range is between 2.5V and 5V with 0.1V increase on a step. On each step, there are three data enrolled. Using these data, voltage variation results in an average HD_{intra} 5.14% and an average HD_{inter} 38.98%.

5.1.2 Cypress CY62256NLL

The Cypress CY62256NLL is a 256k bit SRAM device. Even though this device is less popular than Microchip 23LC1024, it's still widely used. One of the reason is because this device has an automatic power-down feature, reducing the power consumption by 99.9 percent when

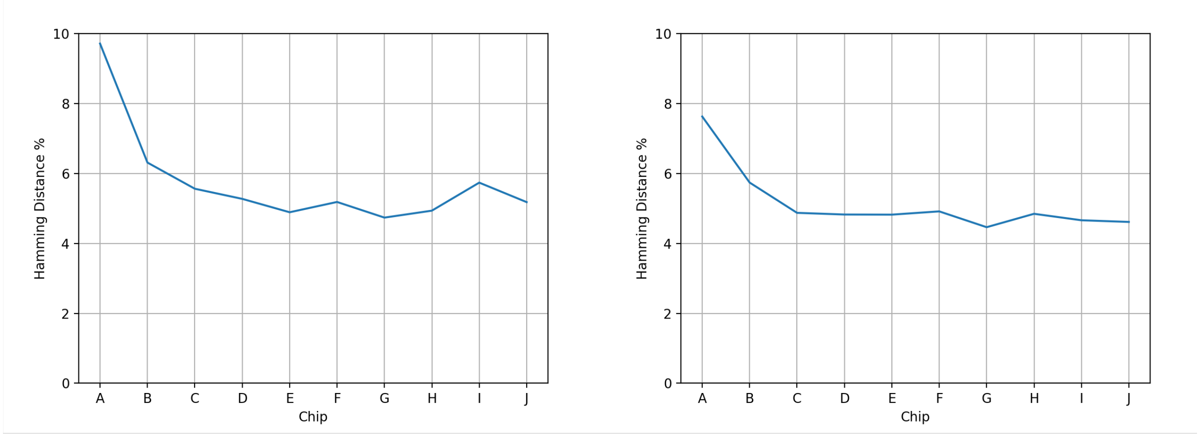

Figure 5.2: HD_{intra} of ten SRAM Microchip 23LC1024. The left is HD_{intra} with constant voltage, the right one is tested based on the voltage variation

deselected. Unlike 23LC1024, CY62256NLL doesn't have a SPI connection which complicate the communication. To communicate, one should utilize its twenty eight pins available. Since it has many pins, it also participate on its significantly larger size compared to 23LC1024. This device is developed using 90nm. Its voltage range is ranging from 4.5V-5.5V. Figure 5.3 shows Cypress CY62256NLL.

Figure 5.3: SRAM CY62256NLL

There are five Cypress CY62256NLL SRAMs that were available during experiment. To check whether this SRAM is a justifiable candidate for PUF, several testing are performed. First, the number of 1's and 0's in an initialization is counted. Fortunately, unlike the 23LC1024, the average distribution of 1's and 0's are similar, both occupy 50% of total bits available.

Next, HD_{intra} and HD_{inter} are calculated on both chips. The calculation are done using twenty data of chip memory values on each chip which retrieved on room temperature, 5V input and 10 seconds interval between each enrollment. From these chips, the average HD_{intra} is 4.94% and the average HD_{inter} is 39.18%.

Last, the effect of voltage variation on the HD_{intra} and HD_{inter} are also evaluated. The calculation are done using chip memory values on each chip which retrieved on room temperature and 10 seconds interval between each enrollment. The voltage range is between 4.5V and 5V with 0.1V increase on each step. On each step, there are ten data enrolled. The average HD_{inter} on voltage variation is 38.75%, while HD_{intra} is 3.55%. Figure 5.4 shows the HD_{intra} between the constant and the variated voltage.

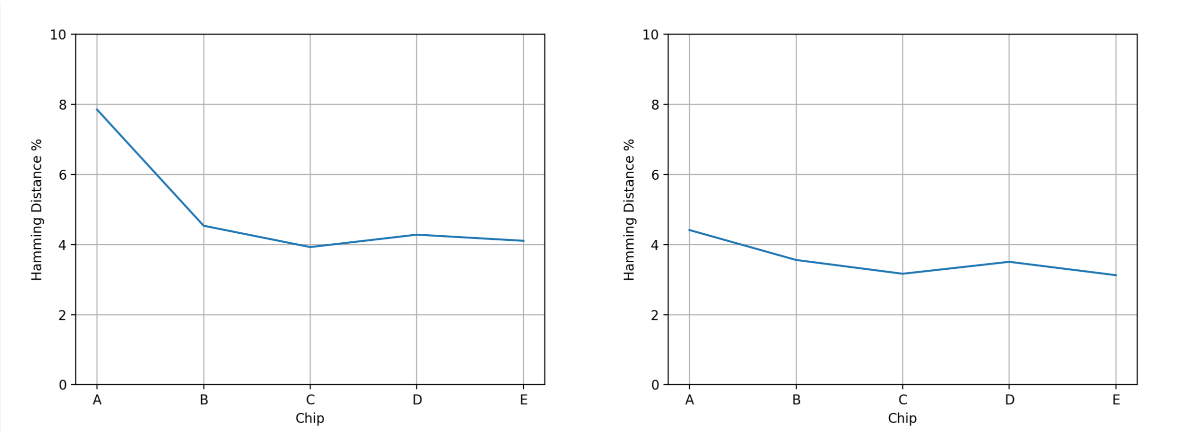

Figure 5.4: HD_{intra} of five SRAM Cypress CY62256NLL. The left is HD_{intra} with constant voltage, the right one is tested based on the voltage variation

From these data, it can be seen that the voltage variation has little effect on the HD_{intra} and HD_{inter}. This fact shows that SRAM Microchip 23LC1024 and Cypress CY62256NLL can be good candidates for SRAM PUF. Even though such fact exists, one should also pay attention that there's no testing on temperature and aging variation. To ensure whether this SRAM is indeed a good candidate, further experiment on the effect of temperature and aging should be conducted.

5.2 Arduino Mega 2560 as the Embedded Platform

After deciding the SRAMs, the next step is choosing the platform on where the system will be build. There are two major candidates, Arduino and Raspberry Pi. Both are chosen due to its popularity, availability (easy to get), and various types available. High popularity means the debugging process can be done fast and many references are available online to help the system development. Availability is important because this thesis goal should be easily used by anyone. Low availability will reduce significantly reusability of this project and user's interest.

Various types available is a good option for system flexibility. For example, if a user want to develop a more complex system on top of this thesis' system or desire to use a more complex error correcting codes, he/she can choose a platform with higher computing capability.

Beside those three factors, another feature to choose Raspberry Pi and Arduino is their GPIO. GPIO availability will enable easy communication between the SRAM and the platform.

Compared to Arduino, Raspberry Pi offers a higher computing capability and relatively easier development. This is because Raspberry Pi is basically a mini linux computer. One can develop using C, C++, Python, etc. Using high-level language will fasten the project development. Unfortunately, Raspberry Pi requires a longer start up time compared to Arduino. It also requires higher electricity power. If one want to use the developed project in embedded area, this two factor is a major trade off.

Due the above consideration, Arduino is chosen. Even though one has to construct the system in C++, this can be a good thing since one can maximize the computing capability easily.

There are various Arduino type available in the market. The chosen Arduino type is Arduino Mega 2560. It is selected because it offers larger memory capability compared to other types, such as 256k bytes of Flash memory, 8k bytes SRAM, and 4k byte EEPROM. Besides, it also has 54 digital I/O pins and 16 analog I/O pins which ease the communication to SRAM CY62256NLL (has 28 pins).

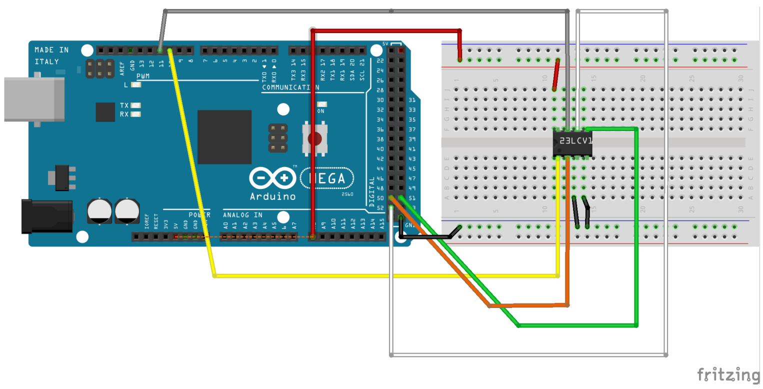

Figure 5.5: Schematic to connect Arduino Mega 2560 with SRAM Microchip 23LC1024

.

.

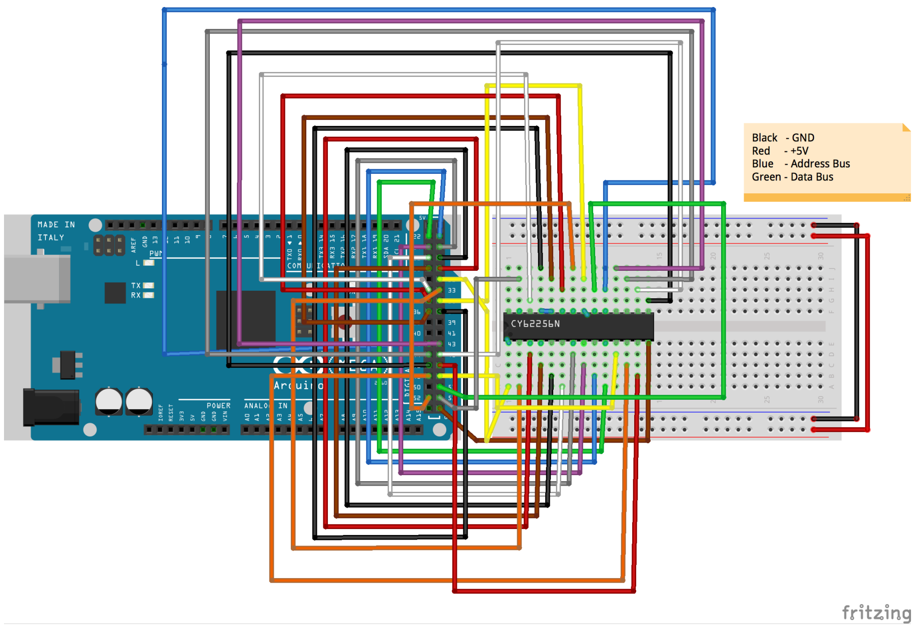

Figure 5.6: Schematic to connect Arduino Mega 2560 with SRAM Cypress CY62256NLL

5.3 Algorithm to Look for Stable Bits

In this section, the test on stable bits produced by two algorithm are shown. The test was done only on a single chip of each SRAM type, one 23L1024 and one CY62256NLL.

5.3.1 Neighbour Stability Analysis

To use this algorithm, first, data of SRAM bits value from various condition (voltages and time difference between enrollment). Afterwards, the bits which remained stable on those enrollments are located. Then, the rank of remained stable bits are calculated. Last, n bits with highest rank can be used according to the necessity. The higher the rank, the more stable that bit should be. The window size used to calculate the rank is 16 (eight neighbors in each side).

23LC1024

There are 500 data of SRAM bits value used for this chip. The voltage variation is from 2.5V - 5.0V. The time difference between enrollment is ranging from 5 seconds until 1 hour. SRAM

23LC1024 itself has capacity 1048576 bits. After doing the calculation from those five hundred data, there are 413374 remaining stable bits.

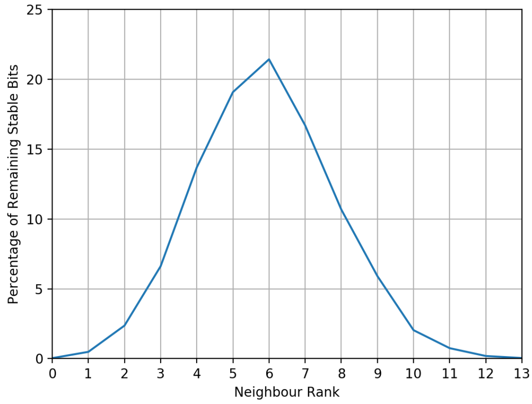

Figure 5.7: Remaining stable bits count according to their rank in SRAM Microchip 23LC1024

From those remaining stable bits, the rank of each bits are calculated. The frequency of bits rank is shown in Figure 5.7. As shown in this figure, there is no bit that has rank 14, 15 and 16. The highest one is only rank 13 with total 172 bits.

Using the bit location as the challenge, the HD_{inter} is 49.76%.

CY62256NLL

Unlike 23LC1024, there are only 109 enrollments done in CY62256NLL. The reason of this decision will be explained in the next section. SRAM CY62256NLL is able to store 262144 bits in its memory. The remained stable bits after 109 enrollments are 84870 bits (32,37%).

The result of the calculation is shown on Figure 5.8. Unfortunately, after the calculation there is no bit that show score 16 (has eight stable neighbor bits on each side). There are two bits that has score 15, 9 bits with score 14, 18 bits with score 13. The highest score count is achieved by score 5 with total count 16502.

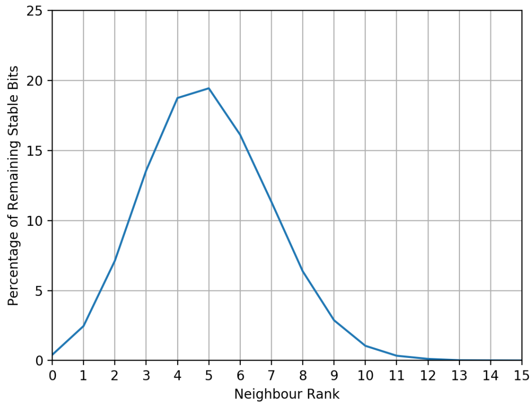

Figure 5.8: Remaining stable bits count according to their rank in SRAM Cypress CY62256NLL

5.3.2 Data Remanence Approach

The result of data remanence analysis on both SRAMs are shown below.

23LC1024

On SRAM 23LC1024, the data remanence analysis is done on time variance between 0-1.0 second. The result can be seen on Figure 5.9. In this figure, it is shown that SRAM 23LC1024 will reach the randomized point if it's turn off for 0.7 second.

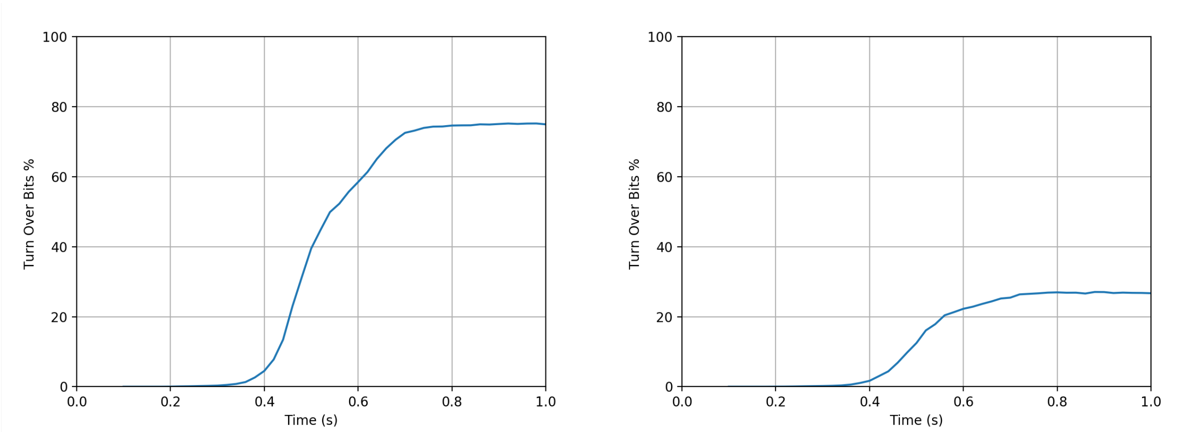

Figure 5.9: Remanence Graph of 23LC1024. Left is remanence 0 and right is remanence 1

CY62256NLL

On SRAM CY62256NLL, the data remanence analysis is done on time variance between 0-1.95 seconds. The result can be seen on Figure 5.9. In this figure, it is shown that SRAM CY62256NLL will reach the randomized point if it's turn off for 0.7 second.

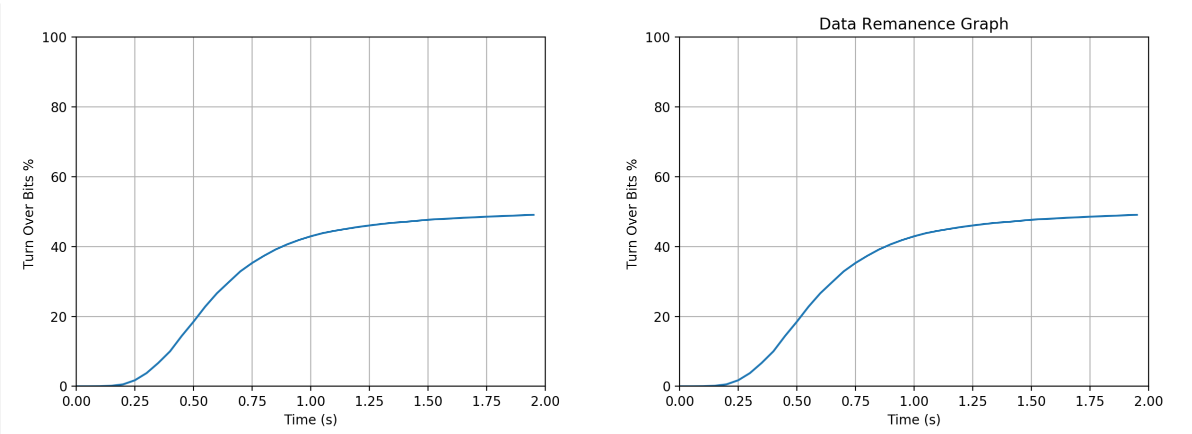

Figure 5.10: Remanence Graph of CY62256NLL. Left is remanence 0 and right is remanence 1

5.3.3 Stability Test on "Stable Bits"

In this section, test results on the effect of time interval and voltage on "stable bits" using both algorithm on each SRAM are shown. The effect of aging and temperature is not tested due to limitation on time and equipment. The test are done on 4662 bits which is twice the length of the bits required to store 256 bits key when using scheme shown on Figure 5.11.

Neighbor Stability Analysis

- Microchip 23LC1024

  To get 4662 bits, there are four ranks included; rank 13 with 172 bits, rank 12 with 778, rank 11 with 3092 bits, and 620 bits of rank 10.

  During testing on variated voltage and time interval, the stable bits generated using neighbor stability analysis show a poor performance by having maximum 2611 bits changing (56.0%). The maximum difference is produced when the difference between enrollment is a week.

- Cypress CY62256NLL

  To get 4662 bits, there are eight ranks included; rank 15 with 2 bits, rank 14 with 9, rank 13 with 18 bits, 99 bits of rank 12, 289 bits of rank 11, 890 bits of rank 10, rank 9 - 2438 bits, and rank 8 - 917 bits.

  Under the voltage and time interval variation, the stable bits generated using neighbor stability analysis show reliability by having maximum 3 changing bits (0,06%).

Data Remanence Approach

- Microchip 23LC1024

  To get 4662 bits, strong 1's are generated using time interval only 0.185 second, while strong 0's are calculated when 0.27 second. The difference between time interval during generation of strong 1's and strong 0's is because the number of 1's that flipped fast are more compared 0's. This also related to the 0's and 1's distribution during normal initialization (0's count for 30% and 1's filled 70%).

  Similar like previous algorithm, the stability of bits produced by using this algorithm is not good. The worst change is happen when using a week as time interval between testing which showing as many as 2183 bits (48.9%).

- Cypress CY62256NLL

  Unlike SRAM 23LC1024, time interval on enrolling strong 1's and 0's on CY62256NLL is not different. Both are enrolled using time interval 0.28 seconds to get 4662 stable bits.

  During the voltage and time interval variation, the stable bits produced by using algorithm also shows a promising result. It only account for maximum 11 bits difference (0.23%).

5.4 Key Storage Scheme

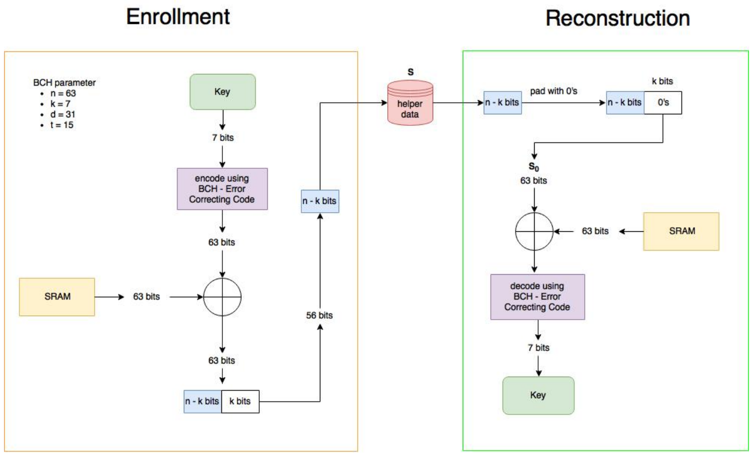

Figure 5.11: Scheme for Storing Key

Chapter 6

Conclusion

6.1 Summary of Thesis Achievements

Summary.

6.2 Applications

Applications.

6.3 Future Work

Future Work.

Bibliography

[1] Chasepaymentech.com. (2018). EMV Chip Card Technology FAQs — Chase Merchant Services. [online] Available at:
`https://www.chasepaymentech.com/faq_emv_chip_card_technology.html` [Accessed 20 Feb. 2018].

[2] Goucher, W. (2016). *Information security auditor*. BCS Learning & Development Limited.

[3] Tilborg, H. and Jajodia, S. (2011). Encyclopedia of cryptography and security. New York: Springer.

[4] S. Skorobogatov, "Physical attacks on tamper resistance: progress and lessons", Proc. of 2nd ARO Special Workshop on Hardware Assurance, 2011.

[5] B. Gassend, D. Clarke, M. Van Dijk, and S. Devadas, Silicon physical random functions, in Proc. 9th ACM Conf. Computer and Commun. Secu- rity, Washington, DC, 2002, pp. 148160.

[6] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld, Physical one-way functions, Science, vol. 297, no. 5589, pp. 20262030, Sept. 2002.

[7] Chang, C., Zheng, Y. and Zhang, L. (2017). A Retrospective and a Look Forward: Fifteen Years of Physical Unclonable Function Advancement. IEEE Circuits and Systems Magazine, 17(3), pp.32-62.

[8] Pim Tuyls, Boris Skoric, and Tom Kevenaar. 2007. Security with Noisy Data: Private Biometrics, Secure Key Storage and Anti-Counterfeiting. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[9] J. Guajardo, S. Kumar, Geert-Jan, and P. Tuyls. FPGA intrinsic PUFs and their use for IP protection (submitted). In Workshop on Cryptographic Hard- ware and Embedded Systems (CHES 2007), 2007.

[10] Ahmad-Reza Sadeghi and David Naccache. 2010. Towards Hardware-Intrinsic Security: Foundations and Practice (1st ed.). Springer-Verlag New York, Inc., New York, NY, USA.

[11] Xiaoxiao Wang and Mohammad Tehranipoor. 2010. Novel physical unclonable function with process and environmental variations. In Proceedings of the Conference on Design, Automation and Test in Europe (DATE '10). European Design and Automation Association, 3001 Leuven, Belgium, Belgium, 1065-1070.

[12] Xiao, Kan, et al. Bit Selection Algorithm Suitable for High-Volume Production of SRAM-PUF. 2014 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), 2014, doi:10.1109/hst.2014.6855578.

[13] Liu, Muqing, et al. A Data Remanence Based Approach to Generate 100% Stable Keys from an SRAM Physical Unclonable Function. 2017 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED), 2017, doi:10.1109/islped.2017.8009192