Unified Pre-fabrication and Post-fabrication Hardware Security

RANJAN MONDAL

ROLL NUMBER: MTC-1301

THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENT FOR THE DEGREE OF MASTER OF TECHNOLOGY
IN COMPUTER SCIENCE IN INDIAN STATISTICAL INSTITUTE 2015

INDIAN STATISTICAL INSTITUTE

TO WHOM IT MAY CONCERN

I hereby recommended that the thesis entitled **"Unified Pre-fabrication and Post-fabrication Hardware Security"** prepared under my supervision by Ranjan Mondal (Roll No. MTC-1301), may be accepted in partial fulfillment for degree of Master of Technology in Computer Science in Indian Statistical Institute.

Dr. Susmita Sur-Kolay
Professor
Advanced Computing and Microelectronics Unit (ACMU)
Indian Statistical Institute, Kolkata

Abstract

Hardware security has emerged as a premier design and manufacturing objective due to the confluence of economic, social, and technology forces. All algorithmically secured cryptographic primitives and protocols rely on a hardware root of trust to deliver the expected protections when implemented in software.

About 25% of the chip design companies send their design to another company for either further integration or fabrication. While this facilitates design reuse and shorter time-to-market, it also provides the opportunity for malicious misappropriation of design. Currently available key based methods resolve this problem by introducing an initialization phase. In this phase, if the initialization is done by a correct initialization code then only the chip works properly. However, the dsiadvantage of this method is that if someone gets hold of the initialization code, then they can initialize all other illegal copies of the chip with the same code.

In this thesis, we have proposed a model by which the chip initialization can be done with different codes. So even if someone knows one initialization key they cannot hack a new chip. Moreover, the method is more robust compared to the available key based method, in terms of the probability of success of a brute-force attack. It also provides a scheme by which each chip can be authenticated uniquely. The model consists of two special modules; one generates the initialization key and the other module enables the main module to work properly if the initialization code is correct.

Acknowledgement

I wish to express my deep sense of gratitude to Prof. Susmita Sur-Kolay, for her guidance, encouragement and facilities extended without which this project would not have taken this present shape. I would also like to thank Dr. Debasri Saha, for her support and valuable advice regarding the topic. I would like to thank my friends, especially Bapi Kar and Manjari Pradhan, and everyone else who supported me with this project. I am very much grateful to all those who were there to help me in understanding the topics, with useful discussions.Your help will always be remembered and appreciated.

Ranjan Mondal

Contents

1 Introduction **2**
1.1 Intellectual property 2
1.2 IC Supply Chain 3
1.3 Hardware authentication 4
1.4 Motivation 4
1.5 Organization of this dissertation 5

2 Pre-fabrication Security **6**
2.1 Introduction 6
2.2 Fundamental framework and Key based active protection . . . 7
2.3 Attacks 8

3 Post-fabrication Authentication **9**
3.1 Introduction 9
3.2 Hardware authentication with PUF-based secret key generation 10
3.3 Different types of PUF 10
3.3.1 Arbiter PUF 11
3.3.2 Ring oscillator PUF 11
3.4 Advantages of using PUF 12
3.5 Limitations 12

4 Model for Unified Pre-fab and Post-fab Hardware Security **14**
4.1 Proposed Model 14

4.2 Details of the Method 16
4.3 Robustness of the Method 17
4.4 Advantages of the Model 18
4.5 Limitations 19

5 Results **20**

6 Conclusion **23**

Chapter 1

Introduction

Hardware is the keystone of a system for computation and communication, on which we run software, or communication protocols to attain the desired functionalities. Hardware comprising integrated circuit chips have become ubiquitous and inexpensive. Computation may include highly confidential transactions from bank, secret data from national defense etc. Security of all these systems involve high-level algorithm as well as low level hardware[3]. For example, company 'X' may be providing a highly efficient missile chip to the national defense via some untrusted supply chain, while in the supply chain, this mission critical chip can be replaced by identical hackable ones with inferior specifications thereby leading to either failure or malfunction. Such scenarios mandate techniques for unique authentication of each chip.

1.1 Intellectual property

Chip is used maximum in industries, business world, national security, banking etc. Even in daily life necessity, entertainment, mobile communication are controlled by chip. Now a days multi-functional chip are integrated on a single chip according to the customer demand. Semiconductor technology has entered into nanometer range so designing a chip is becoming too

challenging, ensuring chip's reliability, stability, functionality and quality. To meet all the user's specification, design goals, low-cost, shorter time-to-market a strategy is used based on the reuse of designed components instead of designing a chip from scratch. This predefined components are called intellectual property(IP) in VLSI industry. Many sequential transformation[5] is done before sending a design for fabrication, so that IP's are not reused by another company.

1.2 IC Supply Chain

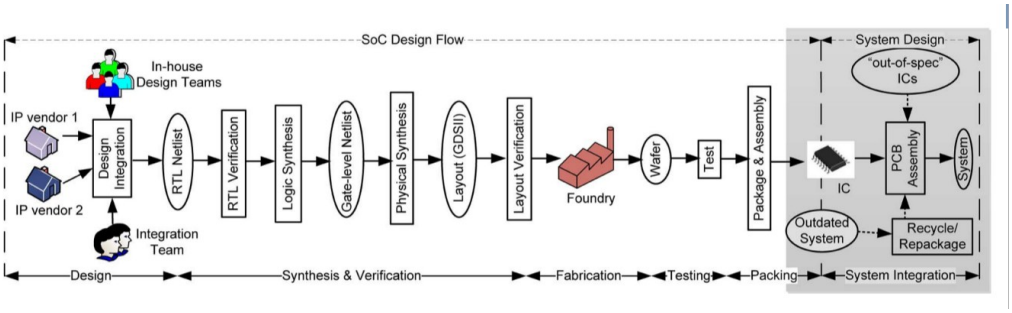

Figure 1.1: semiconductor supply chain

IC supply chain is distributed allover the world. Standard IC supply chain is given above. First system specification is taken and then procuring intellectual property (IP) designs from third-party design houses. Some components are designed in-house. Combining both of then IC layout is generated. A blueprint of the design is then sent to the foundry(fabrication) that manufactures the ICs. The ICs are then tested at the manufacturing site and often at third-party test facilities. Then the fault-free ICs are packaged and sold. There are multiple possible attacks that can be possible in this supply chain.

1. If the attacker is in design house, he/she may add malicious circuit or modify the existing circuit.

2. A malicious foundry may overbuild the number of ICs and sell the excess ICs in the gray market.

3. The attacker can do reverse engineering and can know about the IP so that he/she can reuse or improve it.

1.3 Hardware authentication

Hardware authentication is needed to prevent the Hardware piracy over the market. Some one can sell identical chip in the name of some other reputed company. It affects the business as well as the reputation of that company. So there should be some technique by which we can identify/authenticate each chip uniquely. We solve this problem using some hardware authentication techniques. In the chapter of Post-fabrication Authentication we will see this in more details.

1.4 Motivation

After design, blueprint of the design is then sent to foundry for fabrication. So a malicious foundry may overbuild the number of IC's and sell the excess IC's in the gray market. Currently available key based methods resolve this problem by introducing a initialization phase. In that phase if the initialization is done by correct initialization code then only chip will work properly. But if someone get hold of the initialization code then they the initialize all other gray market chips with the same code. As fabricated chips are available, attacker can try to hack parallely with brute-force method. If a single chips is cracked then all other chips can be hacked. Motivation is to solve this problem.

1.5 Organization of this dissertation

In this chapter we have discussed about the lacuna in basic hardware security while describing the semiconductor supply chain. In the second chapter we present the existing methods for key based protection against hardware piracy. The third chapter is about hardware authentication with secret key generation based on Physical Unclonable Functions (PUF). Next we propose our unified robust model, which solves the problem of hardware piracy more robustly as well as it does hardware authentication. The fifth chapter contains the experimental results on ISCAS89 benchmark circuits and the last chapter has the concluding remarks.

Chapter 2

Pre-fabrication Security

2.1 Introduction

From the last decade the complexity and cost of modern ICs are increasing. So design house has to seek various external agencies, such as EDA companies, IP vendors, library providers, and fabrication foundries. For the participation of external entities in the design and manufacturing flow, it produces numerous hardware security issues. Among all other security issues, hardware piracy is most expensive. Most leading edge design houses or companies outsource their fabrication to the foundries for manufacturing cost. Foundries are hard to be trusted. There are variety of techniques has been proposed for fighting against hardware piracy[2]. Here I have discussed one key Idea which will be useful later. The IC can work in two different modes: normal mode and slow mode depending on whether the circuit(flip-flop) is initialized in the given key state. If the key is given wrong then the circuit will work slowly.

2.2 Fundamental framework and Key based active protection

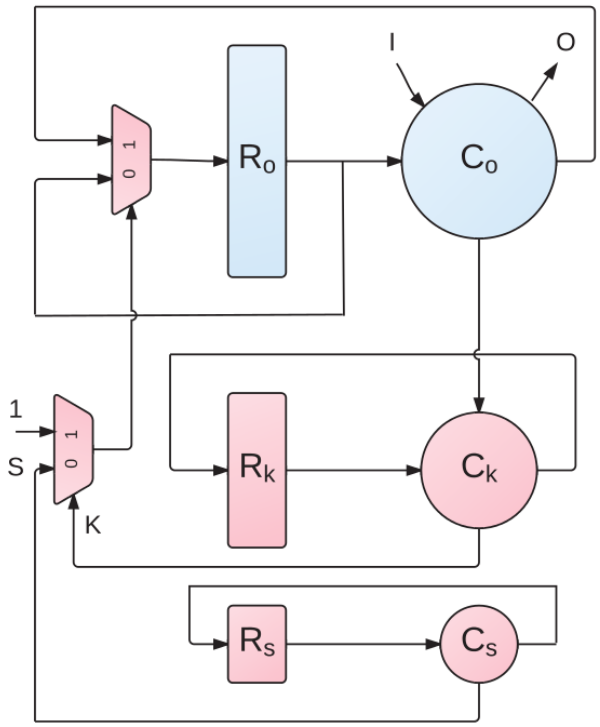

Figure 2.1: Fundamental framework

The Fundamental framework is the figure 2.1. There are three components. Blue components R_o and C_o are the registers and combinational logic of the original design, respectively. The red components are the stuttering control logic, which is composed of the key indicator (R_k, C_k) and the stuttering indicator (R_s, C_s). The key indicator will output an one-bit signal K, which will be 1 when the key is correct, i.e the flipflop values of R_k is equal to the given key(fixed), otherwise it will be 0. The stuttering indicator outputs a one-bit signal S. The stuttering indicator is a finite-state machine and it produces output S equal to 1 when it reaches to the particular state.

Except that particular state all other state gives S equal to 0. It rotates through all the states, so it produces some delay to the original circuit. The final stuttering control signal is $K + \bar{K}S$. Thus the circuit will stutter if and only if $K = S = 0$, otherwise the circuit will work as normal. Note that the security strength of obfuscation is directly determined by the number of FFs in R_k . Assume there are k FFs in R_k and there only exists one set of FF values that will render $K = 1$, then a random power-up state will have a possibility of $\frac{1}{2^k}$ to fall in the normal mode.

2.3 Attacks

1. Brute force attack can be done by randomly generating the power-up state i.e finding the right control input, until the throughput of tested IC is better. The key indicator contains k FFs, so the possibility of successful random guessing will only be $\frac{1}{2^k}$.

2. If the attacker is aware of the existence of Stuttering control logic he/she can systematically analyze to regain a normal design.

Chapter 3

Post-fabrication Authentication

3.1 Introduction

Embedded and mobile systems are increasingly getting involved in information security and safety-critical applications. For this sensitive applications of embedded device, there is a increasing need for enabling technologies to validate and verify the integrity of a system's software/hardware state against malicious attestation. For this purpose, various approaches to attestation have been proposed. Here the the discussion is limited to the hardware attestation. The basic structure is that prover evaluate the platform and sends a status report to verifier to demonstrate that it is in a known and trustworthy state. A practical lightweight attestation scheme for embedded devices should have low hardware overhead and reasonable attestation times. If the end users dosn't authenticate the chip correctly, it will not work properly.

3.2 Hardware authentication with PUF-based secret key generation

Physical unclonable function (PUF) a physical entity that is embodied in a physical structure and is easy to evaluate but hard to predict. An individual PUF device is easy to make but practically impossible to duplicate even if exact maufacturing process is given. PUFs depend on the uniqueness of their physical microstructure. This microstructure depends on random physical factors introduced during manufacturing. These factors are unpredictable and uncontrollable which makes it virtually impossible to duplicate or clone the structure. In hardware security, challenge-response authentication is a family of protocols in which one party presents a challenge and another party must provide a valid response to be authenticated. PUF works in the same way, each PUF device has a unique and unpredictable way of mapping challenges to responses, even if it was manufactured with the same process as a similar device. It is infeasible to construct a PUF with the same challenge–response behavior as another given PUF because exact control over the manufacturing process is infeasible. Today, PUFs are usually implemented in Integrated Circuits and are typically used in applications with high security requirements. Before releasing the chip into market, for each chip the challenge-response pair are kept in the company database. When the chip goes to the end users through the untrusted supply chain, the end users can authenticate that chip by observing the challenge-response pair.

3.3 Different types of PUF

There are different types of PUFs Like- Arbiter PUF,Ring oscillator PUF,ALU PUF[1]. Here I will Disscuss only two of them.

3.3.1 Arbiter PUF

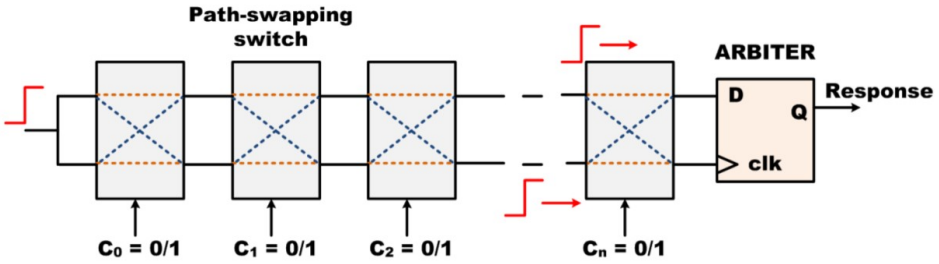

Figure 3.1: Arbiter PUF

Arbiter PUF[4] composed of n two-port switching stages, for an n bit challenge size as shown in the figure 3.1. If the switch input C_i is equal to 1 then it swaps the two line else leaves as it is. As there are n bit challenge, there can be 2^n possible paths, out of those 2^n paths a single path is selected by the challenge. The delay is accumulated at the end of the path. This delay is compared by an arbiter circuit. Usually arbiter circuit is taken as edge triggered D flipflop. For each clock, arbiter gives 1-bit decision or response. To get n-bit responses we need to use this same structure n times. The Advantages of using this arbiter PUF is that it has Simple structure and low hardware overhead as each stage takes only two 2:1 MUXes.

3.3.2 Ring oscillator PUF

Ring oscillator[4] based PUF consists of 2^n ring oscillators,for n bit challenge size as shown in the figure 3.2. An n bit challenge applied to the two $2^n : 1$ MUXes. It selects two different ring oscillator from the bank of 2^n oscillators. Due to Process variation during fabrication, all the ring oscillators have different oscillation frequencies. Multiplexers selects two oscillators and counters the frequencies of those using counters. By comparing those frequencies with comparator, it gives decision or response. The main disadvantage of using this type of PUF is it requires exponential hardware.

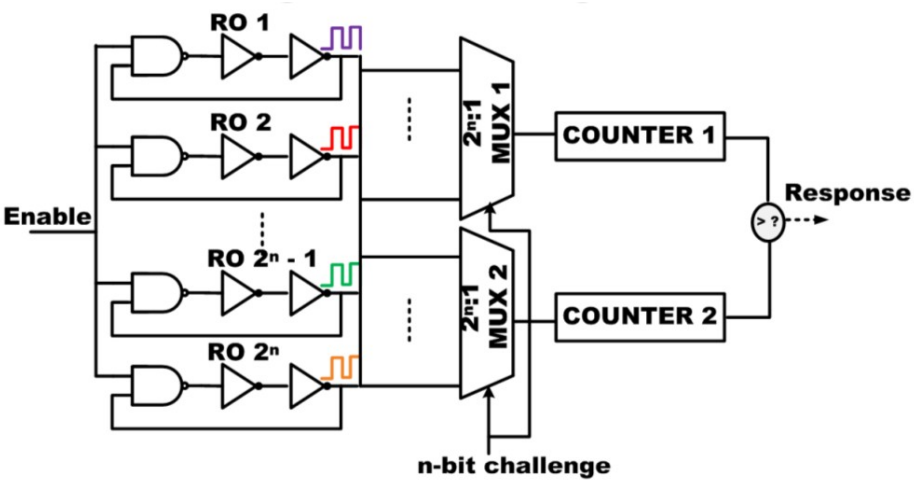

Figure 3.2: Ring Oscillator PUF

3.4 Advantages of using PUF

1. Secret key generation depends upon Intrinsic properties of devices.
2. Key never leaves the IC's cryptographic boundary, nor be stored in a non-volatile memory.
3. Key is deleted after usage in decryption or encryption process.
4. PUF reduces cost and rise the security.
5. PUF can be used as random number generator. Random in the sense, for each chip it is constant but random for different chip.

3.5 Limitations

Although PUFs are physical structures that exploit side-effects in chip manufacturing, it has a limited resilience against the environmental effects, like

- temperature, power supply variations or silicon aging effect. So it is necessary to have effective error correction mechanisms keeping the hardware over head low.

Chapter 4

Model for Unified Pre-fab and Post-fab Hardware Security

In the pre-fab hardware Security all the chips are initialized with the same code. So, if an unauthenticated person somehow get hold of the key then he/she can validate all the extra chips which were fabricated by untrusted fabrication foundries. Apart from that, if someone gets hold of the control line of that flipflop initialization block then it can be attacked by the brute-force method. As the fabricated chips are available, they can try all 2^n combination parallely. If one of the chips is cracked then all other chips can be validated. Here I have Proposed a model by which we can initialize the chips with different initialization code. It also provides the facility to authenticate every chip when it goes to the end user. Even probability of breaking the code is less than the method mentioned previously in the chapter two.

4.1 Proposed Model

The figure 4.1 shows the block diagram of the proposed model of unified Pre-fab and Post-fab Hardware Security.

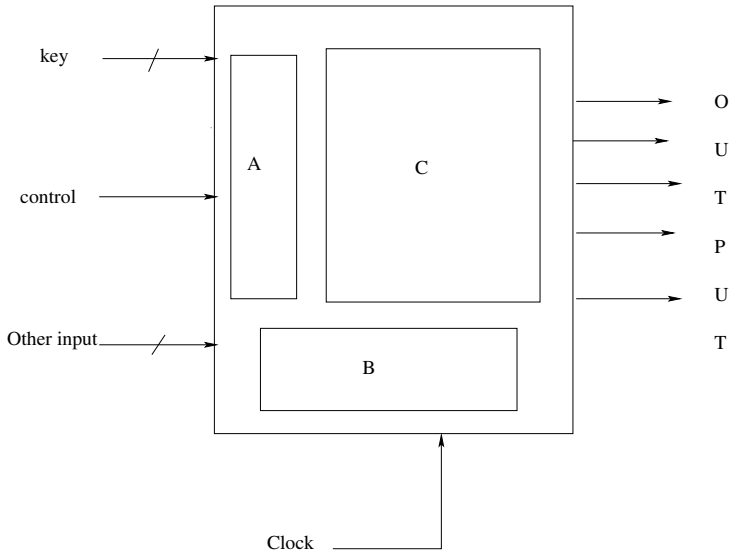

Figure 4.1: Fundamental Block Diagram

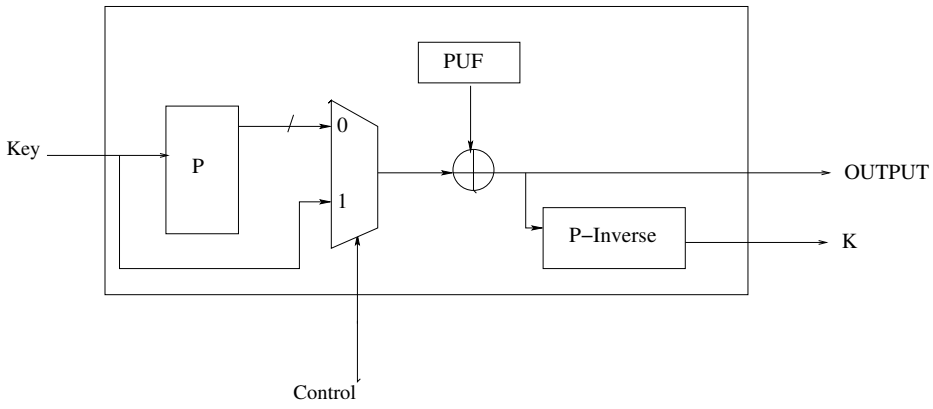

Figure 4.2: Block A

In the figure 4.2 the key and control is an input to the block 'A'. Block 'A' consist of a Permutation block 'P' which permutes the key. The key is of length n-bit. Within block 'A', there is also a another Permutation (P-inverse) block which is exactly inverse Permutation of block 'P'. 'A' Block works in two different modes according to the control input. If the control input is one then the Block 'A' works to supply the input 'K' to the Block

'B' and if the control input is zero then 'A' block works to generate key for chip authentication.

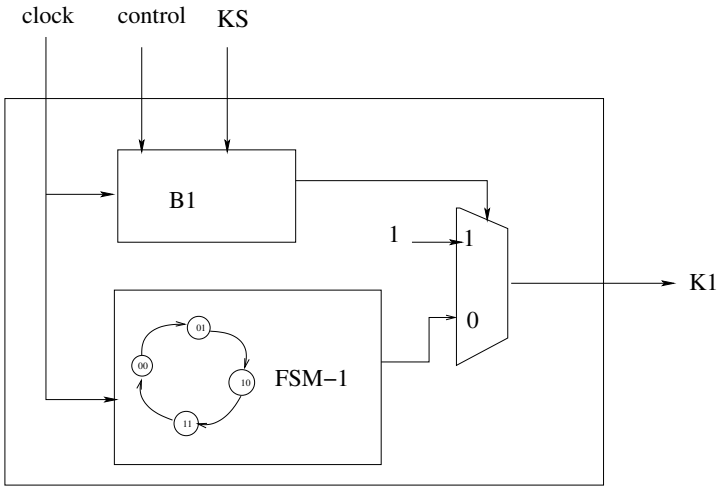

Figure 4.3: Block B

As shown in the figure 4.3 Block 'B' has 2 inputs, control and 'KS'. 'KS' is of k bit which is selected from the n bit line 'K'. From the block 'B' a single bit output 'K1' goes to the control of main module Block 'C'. Block 'B' consists of block B1 which has k flip-flops and the output from the Block 'B1' is one if it is initialized with the correct state. Block B has a stuttering control logic block 'FSM-1', which provides delay. Fsm-1 has some number of rotating state. Out of them, only one state gives output one and all other states give zero. 'K1' is one if the Block 'B1' is initialized with correct code through 'KS', else it is zero. If the Block 'B1' is not initialized with correct code, then 'K1' will be one after some clock pulse, not instantly.

4.2 Details of the Method

EDA companies outsource their fabrication to the offshore foundries for the sake of lower labor and manufacturing cost. Ater that fabricated chips are

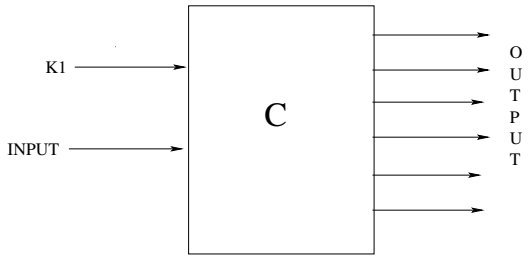

Figure 4.4: Block C

returned to the company (Design owner). Let us say we have used k number of flipflop in block B1 and input(key) is of n bit, where $n > k$. Typically k is taken to be $\frac{n}{2}$ to get maximum probability of robustness. The company takes each fabricated chip design owner and set the control line to 0 as well as get note down of the two keys.

1. Authentication key

2. Initialization key.

Initialization key and authentication may be the same. We apply the n bit input(key) by setting the control line to zero and note down the output as authentication key for that chip. So by this way we get the challenge response pair for that each chip.

The same process is applied to get the Initialization key. The most important thing is when we apply the n bit input, we have to carefully select those k places with correct activation key. When the Chip goes to the end users, initialization key is given to them so that they can initialize the chips and get good performance.

4.3 Robustness of the Method

The Robustness of this method can be defined for different type of attacks. In this model the control line is known to all the end users of chip.

CASE 1: If the attacker knows about the all the details about the block 'A' except the Permutation block, then the probability of attack is greater than $\frac{1}{(2^k)}$.

Let the exact k bits is Z_k, which initializes flip-flop correctly. Let the attacker guesses that k positions of the n bit input key and tries to initialize the block 'B1' with say M. Let $M_{i_1,i_2,i_3,...i_k} \in \{0,1\}^k$ denotes the bit string obtained by selecting $i_1, i_2, i_3, ...i_k$th bits from M.

Let Pr be the probability of success of a brute-force attack.

$$\begin{aligned} &Pr[M_{i_1,i_2,i_3,...i_k} = Z_k] \\ &= P[Z_k | i_1, i_2, i_3, ...i_k \text{ is the right choice}] \times P[i_1, i_2, i_3, ...i_k \text{ is the right choice}] \\ &= \frac{1}{(2^k)({}^nC_k)}. \end{aligned} \tag{4.1}$$

CASE 2: If the attacker knows all about the block 'A' including the Permutation details and the attacker doesn't know about which particular k bit of input key are used in block 'B1' flipflop initializatio, then the probability of attack is $\frac{1}{{}^nC_k}$.

CASE 3: When the attacker doesn't know about anything expect control signal then attacker can try all combination of n bit number. So the probability of attack is $\frac{1}{2^n}$.

In the cases 1 and 2 the probability of attack is minimum when $k = \frac{n}{2}$.

4.4 Advantages of the Model

1. Using this model we get unique initialization key for each chip. After getting the fabricated chip parallel brute-force attack is not possible, and also if one chip is cracked, other chip is as hard as cracking a new chip.

2. The model is more robust than the method mentioned in chapter two

in terms of probability of attack.

3. This model can be extended as active chip protection by removing the stuttering control logic.

4. With this model we can validate or authenticate each of the chips uniquely so that chip owner can claim and prove his ownership of the chip.

4.5 Limitations

Implementing this model will increase hardware overhead. Still we can decrease it by converting the Permutation block into PUF random number generator compromising some clock cycle. Due to the environmental effects like temperature, power supply variations or silicon aging effect, PUF can be affected. So while implementing this model we have to use effective error correction mechanisms to get accurate PUF random number generator so that for each chip the number doesn't change with time.

Chapter 5

Results

The proposed module has been implemented in Verilog for the ISCAS 89 benchmark circuits. The values of n and k have been chosen by empirically based on the pair which provides maximum robustness against brute force attacks.

Table 5.1 shows the robustness of the method in terms of the probability of success of a brute-force attack and the values of n and k reported are the ones which gives the best robustness, i.e., minimum probability of success of brute-force attacks. As mentioned in chapter 4 the probability of success of a brute-force attack varies depending upon the amount of information available to the attacker. The first column denotes the benchmark circuit names of ISCAS89. The next five columns are the number of primary inputs, the number of primary outputs, and the number of FFs, length of initialization key, number of flipflop at block B1. The next five columns shows the the probability of success of brute-force attack Attack-1, Attack-2 and Attack-3 for CASE-1, CASE-2, CASE-3 respectively as mentioned in Section 4.3. Column 10 shows the probability of the success of a brute force attack in the currently available key based method.

Table 5.2 shows the performance overhead evaluations on the ISCAS89 benchmark suite. In the experiment we have taken the initialization key

Table 5.1: Robustness of the method

Benchmark	# PIs	# POs	# FFs	n	k	Attack-1	Attack-2	Attack-3	Cur-prob
s382	3	6	21	6	3	0.00625	0.05	0.015625	0.125
s400	3	6	21	6	3	0.00625	0.05	0.015625	0.125
s526	3	6	21	6	3	0.00625	0.05	0.015625	0.125
s838	34	1	32	16	8	3.0351E-07	7.77E-05	1.5258E-05	0.039
s953	16	23	29	16	8	3.0351E-07	7.77E-05	1.5258E-05	0.039
s5378	35	49	179	16	8	3.0351E-07	7.77E-05	1.5258E-05	0.039
s9234	36	39	211	16	8	3.0351E-07	7.77E-05	1.5258E-05	0.039
s13207	62	152	638	16	8	3.0351E-07	7.77E-05	1.5258E-05	0.039
s15850	77	150	534	16	8	3.0351E-07	7.77E-05	1.5258E-05	0.039
s35932	35	320	1728	16	8	3.0351E-07	7.77E-05	1.5258E-05	0.039
s38417	28	106	1636	16	8	3.0351E-07	7.77E-05	1.5258E-05	0.039
s38584	38	304	1426	16	8	3.0351E-07	7.77E-05	1.5258E-05	0.039

as 16 bit and the number of flipflop at block B1 is 8 for the circuits s838, s953, s5378, s9234, s13207, s15850, s35932, s38417, s38584. For the circuits s382,s400,s526. We have taken initialization key as 6 bit and the number of flipflop at block B1 is 3 as these circuits itself has number of inputs equal to 6. For all the circuits there is 2 bit for the stuttering indicator. In the experiment we have taken PUF, random number generator as constant. We have integrated the module with the ISCAS89 benchmark and synthesized the circuits in terms of area and power with the help of Synopsys Design Compiler. The first column denotes the benchmark circuit names of ISCAS89. The next five columns are the number of primary inputs, the number of primary outputs, and the number of FFs, length of initialization key, number of flipflop at block B1. Columns 7-9 shows the post-synthesis power in the flowing order: original synthesized power of benchmark circuits, the added power after integration with the module, the percentage of increase of power. Columns 10-12 shows the post-synthesis area in the flowing order: original

synthesized area of benchmark circuits, the added area after integration with the module, the percentage of increase of area.

Table 5.2: Power and Area overhead of proposed method for ISCAS 89 benchmark circuits

Benchmark	Circuit details					Power(mW)			Area		
	PI	PO	FF	n	k	ORI	INT	%	ORI	INT	%
s382	3	6	21	6	3	10.63	107.73	913.45	138	1664.1	1105.87
s400	3	6	21	6	3	11.89	108.9	815.9	147.18	1673.3	1036.91
s526	3	6	21	6	3	4.6	101	2095.65	101.2	1627.3	1508
s838	34	1	32	16	8	47.58	250.1	425.64	393.3	3349	751.51
s953	16	23	29	16	8	8.53	204	2291.56	174.7	3079	1662.45
s38417	28	106	1636	16	8	22.33	277.52	1142.81	295.7	3636	1129.62
s5378	35	49	179	16	8	1305.8	1511.9	15.78	6157.8	9215	49.65
s9234	36	39	211	16	8	1233	1511	22.55	2984	4760	59.52
s13207	62	152	638	16	8	17931	18053	0.68	88880	91785	3.27
s15850	77	150	534	16	8	519.74	790.12	52.02	11323	14891	31.51
s35932	35	320	1728	16	8	2810.1	2994	6.54	32460	35365.8	8.95
s38584	38	304	1426	16	8	597.73	798.32	33.56	5317	8250	55.16

From the experiments we observe that the proposed module can be used in large circuits such as s13207, s5378, s35932 very effectively. Further experiments on even larger benchmark circuits such as the ITC99 ones need to be carried out.

Chapter 6

Conclusion

In this thesis I have proposed a model by which hardware piracy can be prevented more robustly as well as IC can be authenticated. IC will work normally only when IC is powered up with a secret key initial state, otherwise it will become much slower. Advantage of using this model is the IC initial secret key is different for each chip. We can use initilization key as authentication key.

The efficiency of the method was demonstrated by evaluations on ISCAS89 benchmarks. We observe that the method is very effective for large circuits, which is very desirable. In future, further improvement in implementation to reduce the overheads need to be studied.

Bibliography

[1] Joonho Kong, Farinaz Koushanfar, Praveen K Pendyala, Ahmad-Reza Sadeghi, and Christian Wachsmann. Pufatt: Embedded platform attestation based on novel processor-based pufs. In *Design Automation Conference (DAC), 2014 51st ACM/EDAC/IEEE*, pages 1–6. IEEE, 2014.

[2] Li Li and Hai Zhou. Structural transformation for best-possible obfuscation of sequential circuits. In *Hardware-Oriented Security and Trust (HOST), 2013 IEEE International Symposium on*, pages 55–60. IEEE, 2013.

[3] Mohamad Rostami, Farinaz Koushanfar, and Ramesh Karri. A primer on hardware security: Models, methods, and metrics. *Proceedings of the IEEE*, 102(8):1283–1295, 2014.

[4] G Edward Suh and Srinivas Devadas. Physical unclonable functions for device authentication and secret key generation. In *Proceedings of the 44th annual Design Automation Conference*, pages 9–14. ACM, 2007.

[5] Hai Zhou. Retiming and resynthesis with sweep are complete for sequential transformation. In *Formal Methods in Computer-Aided Design, 2009. FMCAD 2009*, pages 192–197. IEEE, 2009.